

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**SIRA BAĞIMLI HAZIRLIK ZAMANLI TEK MAKİNELİ ÇİZELGELEME
PROBLEMLERİ: GIDA SEKTÖRÜNDE BİR UYGULAMA**

**YÜKSEK LİSANS TEZİ
Sena KIR
(507091152)**

Anabilim Dalı : Endüstri Mühendisliği

Programı : Endüstri Mühendisliği

Tez Danışmanı: Doç. Dr. Mehmet Mutlu YENİSEY (İTÜ)

HAZİRAN 2011

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**SIRA BAĞIMLI HAZIRLIK ZAMANLI TEK MAKİNELİ ÇİZELGELEME
PROBLEMLERİ: GIDA SEKTÖRÜNDE BİR UYGULAMA**

**YÜKSEK LİSANS TEZİ
Sena KIR
(507091152)**

Tezin Enstitüye Verildiği Tarih : 04 Mayıs 2011

Tezin Savunulduğu Tarih : 10 Haziran 2011

**Tez Danışmanı : Doç. Dr. Mehmet Mutlu YENİSEY (İTÜ)
Diğer Jüri Üyeleri : Doç. Dr. Ekrem DUMAN (Doğuş Üniv.)
Yrd. Doç. Dr. Murat BASKAK (İTÜ)**

HAZİRAN 2011

Babama ve eşime,

ÖNSÖZ

Bu çalışma, günümüz rekabet koşullarında tam zamanında üretim felsefesini uygulamak zorunda olan işletmelerin, çizelgeleme problemlerine çözüm getirmek amacıyla hazırlanmıştır. Çalışma kapsamında gıda sektöründe faaliyet gösteren bir işletmede uygulama çalışması yapılmıştır.

Yüksek lisans tez çalışmalarım boyunca göstermiş olduğu destek ve teşvikinden dolayı değerli hocam, danışmanım, Doç. Dr. Mehmet Mutlu YENİSEY'e, yardım ve anlayışından dolayı değerli mesai arkadaşım Arş. Gör. Elif Elçin GÜNAY'a, desteklerinden dolayı TÜBİTAK Bilim İnsanı Destekleme Daire Başkanlığı'na ve çalışmam boyunca her zaman desteklerini hissettiğim, annem, babam ve eşime teşekkürü bir borç bilirim.

Haziran 2011

Sena KIR

İÇİNDEKİLER

ÖNSÖZ.....	v
İÇİNDEKİLER	vii
KISALTMALAR	ix
ÇİZELGE LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xiii
ÖZET.....	xv
SUMMARY	xvii
1. GİRİŞ	1
2. LİTERATÜR ÇALIŞMASI	5
2.1 Sıra Bağımsız Çizelgeleme Problemleri.....	5
2.1.1 Paralel Makineli Çizelgeleme Ortamı	5
2.1.2 Tek Makineli Çizelgeleme Ortamı	5
2.1.3 Akış Tipi Çizelgeleme Ortamı	6
2.1.4 Atölye Tipi Çizelgeleme Ortamı	8
2.2 Sıra Bağımlı Çizelgeleme Problemleri.....	8
2.2.1 Paralel Makineli Çizelgeleme Ortamı	8
2.2.2 Tek Makineli Çizelgeleme Ortamı	9
2.2.3 Akış Tipi Çizelgeleme Ortamı	12
2.2.4 Atölye Tipi Çizelgeleme Ortamı	14
3. ÇİZELGELEME PROBLEMLERİNDE KULLANILAN META-SEZGİSEL YÖNTEMLER	17
3.1 Tavlama Benzetimi.....	18
3.2 Yasaklı Arama Algoritması.....	21
3.2.1 Yasaklı Arama Algoritmasının Temel Bileşenleri	22
3.2.1.1 Kısa Dönemli Hafıza	22
3.2.1.2 Aspirasyon Kriteri	22
3.2.1.3 Uzun Dönemli Hafıza	22
3.2.1.4 Hareket Mekanizması	22
3.2.1.5 Durdurma Koşulu	22
3.2.2 Yasaklı Arama Algoritmasının İşlem Adımları	23
3.3 Genetik Algoritmalar.....	23
3.3.1 Genetik Algoritmalarla İlgili Temel Kavramlar	25
3.3.1.1 Gen ve Alel Gen.....	25
3.3.1.2 Kromozom.....	25
3.3.1.3 Popülasyon	25
3.3.1.4 Uygunluk Fonksiyonu	25
3.3.2 Genetik Algoritmalarda Kodlama	26
3.3.2.1 İkili Kodlama.....	26
3.3.2.2 Permütasyon Kodlama	26
3.3.2.3 Değer Kodlama	26
3.3.2.4 Ağaç Kodlama.....	26
3.3.3 Genetik Algoritmaların Parametreleri	26
3.3.3.1 Popülasyon Büyüklüğü	27

3.3.3.2 Çaprazlama Olasılığı	27
3.3.3.3 Mutasyon Olasılığı	27
3.3.3.4 Seçim Metodu.....	27
3.3.4 Genetik Algoritmada İşlem Adımları	29
3.4 Karınca Kolonisi Eniyileme Algoritması	29
3.5 Yapay Bağışıklık Sistemi Algoritması	31
3.6 Parçacık Sürü Eniyileme Algoritması	32
3.7 Yapay Sinir Ağları.....	34
3.7.1 Yapay Sinir Ağlarının Özellikleri	36
3.7.1.1 Doğrusal Olmama.....	36
3.7.1.2 Öğrenme	37
3.7.1.3 Genelleme.....	37
3.7.1.4 Uyarlanabilirlik.....	37
3.7.1.5 Hata Toleransı	37
3.7.2 Genel Bir Yapay Sinir Ağı Modeli	37
4. TEK MAKİNELİ SIRA BAĞIMLI HAZIRLIK ZAMANLI ÇİZELGELEME PROBLEMİ	39
4.1 Problemin Matematiksel Çözümü	40
4.1.1 Literatürde Mevcut Olan Çözüm	40
4.1.2 Önerilen Çözüm.....	42
4.2 Problemin Meta-sezgisel Çözümü	44
4.2.1 Problemin Çözümünde Kullanılan Yasaklı Arama Algoritması	45
4.2.1.1 Başlangıç Aşaması.....	45
4.2.1.2 İteratif Aşamalar	46
4.2.2 Problemin Çözümünde Kullanılan Genetik Algoritma	47
4.2.2.1 Başlangıç Aşaması.....	47
4.2.2.2 İteratif Aşamalar	49
4.3 Önerilen Meta-sezgisel Algoritmanın Test Edilmesi	49
4.3.1 İstatistiksel Analiz.....	52
4.3.1.1 Problem Parametrelerinin Değişim Durumu	52
4.3.1.2 Yasaklı Arama Algoritması Parametrelerinin Değişim Durumu	53
4.3.1.3 Genetik Algoritma Parametrelerinin Değişim Durumu	57
4.3.2 İstatistiksel Analiz Sonuçları	61
5. GIDA SEKTÖRÜNDE BİR UYGULAMA ÇALIŞMASI.....	63
5.1 İşletmenin Tanıtımı.....	63
5.2 Problemin Tanımı	63
5.2.1 Taze Peynir Üretimi.....	64
5.2.2 Taze Peynir Üretimi İçin Ana Üretim Planı	64
5.3 Problemin Çözümü	68
6. SONUÇ VE ÖNERİLER.....	71
6.1 İstatistiksel Sonuçlar	71
6.2 Öneriler.....	73
KAYNAKLAR.....	75
EKLER	83

KISALTMALAR

TSP	: Traveling Salesman Problem (Gezgin Satıcı Problemi)
YAA	: Yasaklı Arama Algoritması
GA	: Genetik Algoritma
ACO	: Ant Colony Optimization (Karıncı Kolonisi Eniyilemesi)
EDD	: Early Due Date (Erken Teslim Zamanlı)
SPT	: Shortest Processing Time (En Kısa İşlem Zamanlı)
LPT	: Longest Processing Time (En Uzun İşlem Zamanlı)
OSH	: Ortalama Sezgisel Hata
YAOH	: Yasaklı Arama Algoritması Ortalama Sezgisel Hatası
API	: Adjacent Pairwise Interchange (Bitişik Komşu Değişimi)

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3.1: Çizelgeleme problemlerinde kullanılan meta-sezgisel yöntemler	19
Çizelge 3.2: Yapay bağışıklık sistemi algoritması terimleri	33
Çizelge 4.1: Hazırlık zamanları matrisi	42
Çizelge 4.2: Hazırlık zamanları matrisi	44
Çizelge 4.3: Önerilen meta-sezgisel algoritmanın deney sonuçları	51
Çizelge 4.4: Önerilen algoritmanın deneyleri için varyans analizi	53
Çizelge 4.5: YAA parametre değişiminin deney sonuçları	55
Çizelge 4.6: YAA iterasyon sayısı ve tabu listesi uzunluğunun YAA'nın kendi çözümü üzerindeki etkisini gösteren deney sonuçları	56
Çizelge 4.7: YAA iterasyon sayısı ve tabu listesi uzunluğunun tüm algoritma çözümü üzerindeki etkisini gösteren deney sonuçları	56
Çizelge 4.8: GA parametre değişiminin deney sonuçları	58
Çizelge 4.9: GA iterasyon sayısı, çaprazlama ve mutasyon olasılığının OSH üzerindeki etkisini gösteren deney sonuçları	59
Çizelge 5.1: Aralık 2010 ana üretim planı	65
Çizelge 5.2: Ürünler arası geçiş hazırlık zamanları matrisi, ürün operasyon ve termin süreleri	67
Çizelge 5.3: Ürünlerin erkenlik ve geçlik cezaları	68
Çizelge 5.4: Deney verileri çizelgesi	69
Çizelge A.1: Test problemlerinin verileri	83

ŞEKİL LİSTESİ

	<u>Sayfa</u>
Şekil 3.1 : Tavlama benzetimi algoritmasının akış şeması.....	21
Şekil 3.2 : YAA akış şeması	24
Şekil 3.3 : Permütasyon kodlama örneği	26
Şekil 3.4 : GA'nın akış şeması	30
Şekil 3.5 : Parçacık sürü eniyileme algoritması akış şeması	35
Şekil 3.6 : Doğal ve yapay sinir hücresi	36
Şekil 3.7 : Yapay sinir ağları akış şeması	38
Şekil 4.1 : Problemin başlangıç çözümünde kullanılan YAA'nın akış şeması	47
Şekil 4.2 : Problemin çözümünde kullanılan GA'nın akış şeması	48
Şekil 4.3 : İş sayısı ve GA'nın işlevselliği arasındaki ilişki	52
Şekil 4.4 : İş sayısı ve ceza değerlerine bağlı olarak OSH'ın değişimi	54
Şekil 4.5 : YAA parametre değişiminin deney sonuçlarının grafiği	55
Şekil 4.6 : GA parametrelerinin değişim durumlarında OSH değişim grafiği	59
Şekil 4.7 : GA iterasyon sayısının değişim durumlarında OSH ve CPU değişim grafikleri	60

SIRA BAĞIMLI HAZIRLIK ZAMANLI TEK MAKİNELİ ÇİZELGELEME PROBLEMLERİ: GIDA SEKTÖRÜNDE BİR UYGULAMA

ÖZET

Tez kapsamında, sıra bağımlı hazırlık zamanlı tek makineli çizelgeleme problemlerinin, farklı teslim tarihli ve farklı erken ve geç bitirme cezalarının olduğu durumlar üzerinde çalışılmıştır. Amaç, tam zamanında üretim felsefesine göre istenilen işi istenilen zamanda tamamlayabilmek, böylece erken bitirme ve geç bitirmenin yaratmış olduğu cezalardan mümkün olduğunca kaçınmaktır.

Problemde tek makine üzerinde çizelgelenmesi istenen işlerin her birinin hazırlık zamanları, kendisinden bir önceki işe bağlı olarak değişmektedir. İşlem süreleri, teslim süreleri ve erken bitirme ile geç bitirmenin yaratmış olduğu maliyetler yani cezalar işlere göre değişkendir. Tüm bu kısıtların göz önüne alınması, problemin karmaşıklığını artırmış ve NP-zor karmaşıklık sınıfında bir problem oluşturmuştur. Çalışma kapsamında öncelikle literatürde bu tip problemlerin çözümü için var olan doğrusal modeller, sezgiseller ve meta-sezgiseller araştırılmıştır. Literatürde var olan ve küçük boyutlu problemler için kullanılan bir karışık tamsayılı doğrusal model incelenmiş, bu modelin zayıf yanlarına alternatif olarak yine küçük boyutlu problemlerin çözümü için kullanılabilecek bir başka karışık tamsayılı doğrusal model önerilmiştir.

Büyük boyutlu problemlerin çözümü, doğrusal modellerle sonlu zamanlarda olamayacağı ya da çok uzun süreceği için çalışmanın devamında meta-sezgisel yöntemler araştırılmıştır. Öncelikle, literatürdeki çizelgeleme problemleri için kullanılmış meta-sezgisel yöntemler incelenmiştir. Bu incelemeler sonucunda, problemin çözümü için yasaklı arama algoritmasının ve genetik algoritmanın kullanılmasına karar verilmiştir. Burada amaç; yasaklı arama algoritmasının genetik algoritma için bir başlangıç çözümü oluşturması, genetik algoritmanın da bu çözümü iyileştirerek en iyiye en yakın nihai çözümü elde edebilmesidir. Bu aşamadan sonra, algoritmalar kodlanarak, en iyi çözümü bilinen küçük boyutlu problem verileriyle çeşitli testler yapılmıştır. Bu testlerin asıl amacı, algoritmanın çözüm performansını en iyiye çekebilecek parametreleri elde ederek, çözümü bilinmeyen büyük boyutlu problemlerin çözümünde kullanmaktır.

Yapılan testlerin sonuçlarına göre yasaklı arama algoritması ve genetik algoritma için parametreler belirlenmiş ve uygulama çalışmasında gıda sektöründe faaliyet gösteren bir işletmedeki problemin çözümünde kullanılmıştır. En iyi çözümü bilinmeyen bu problem üzerinde de deneyler yapılarak meta-sezgisel algoritmayla bulunan en iyi çözüm, problemin en iyi çözümü olarak kabul edilmiş ve algoritmanın performansı istatistiksel olarak değerlendirilmiştir.

SINGLE MACHINE SCHEDULING PROBLEMS WITH SEQUENCE DEPENDENT SETUP TIMES: AN APPLICATION IN FOOD INDUSTRY

SUMMARY

In this thesis, single machine scheduling problem with sequence dependent setup times, different due dates, different earliness and tardiness costs was studied. The purpose is completing the intended works in the intended times so avoiding costs as far as possible, which is caused by earliness and tardiness.

The works which are intended to schedule in a machine have setup times that are dependent on its previous work. Processing times, due dates, earliness and tardiness costs are different in respect of the works. Considering all these restrictions increases complexity of the problem and the problem becomes into NP-hard complexity class. In this study, linear models, heuristics and meta-heuristics, which are used for solving this kind of scheduling problems, were researched in literature previously. A mixed integer linear model, which is used for solving small scaled problems in literature, was analyzed and another mixed integer linear model was proposed again for solving small scaled problems alternatively weakness of preexisting model.

Solution of large scaled problem is not possible in complete time or takes too long time, so later on meta-heuristic methods were researched. At first, meta-heuristic methods, which are used for scheduling problems in literature, were analyzed. Following this research, tabu search and genetic algorithm were decided to solve the problem. Tabu search algorithm finds a good initial solution and genetic algorithm enhances it. In this way genetic algorithm finds optimum solution or an approximate solution. After that, algorithms were coded and tested with data sets whose optimum solution is known. The main purpose of these tests is obtaining of best parameters, which optimize performance of algorithms, and using these parameters for large scaled problems.

According to results of tests, parameters were determined for tabu search and genetic algorithm and used to solve a problem of an enterprise, which operates in food industry. The problem, whose optimum solution is unknown, was tried to solve proposed meta-heuristic algorithm. Results of the experiments made, the best solution was considered of optimum solution and accordingly this solution performance of the algorithm was tested statistically.

1. GİRİŞ

Günümüz rekabet koşullarında işletmeler, pazar ve müşteri portföylerini korumaya çalışmakta, yeni pazarlar aramakta, müşteri istek ve beklentilerini tam olarak karşılayabilmek için yeni tasarımlar üzerinde yoğunlaşmaktadırlar. Fakat rekabet üstünlüğü için mükemmel bir ürün tasarlamaya ve pazarlamaya yoğunlaşmanın yanı sıra, bu ürünleri müşteriye tam zamanında sunmak da bir zorunluluk halini almıştır.

İşletmelerin, istek ve beklentilerinde standartları her geçen gün artan müşteriye etkili ve verimli olarak cevap verebilmeleri gitgide karmaşıklaşan üretim problemlerini de beraberinde getirmektedir. Çünkü işletmeler, kısıtlı kaynaklar altında, maliyetlerini yükseltmeden tam zamanlı üretim yapmak zorundadırlar. İşe değer katmayan her türlü atığın ve değişkenliğin yok edilmesi ve malzemelerin gerektiği yerde ve gerektiği zamanda çekilmesi tam zamanında üretimin esasını oluşturur [1].

Tam zamanında üretim stratejisini uygulayan bir işletmenin etkili olması, operasyonların geç kalan iş sayısını ve hazırlık sayısını enküçüklemek gibi belli bir hedef kritere göre kısıtlar göz önünde bulundurularak, kaynaklar üzerinde sıralanması işleminin, yani çizelgelemesinin iyi ve doğru yapılması demektir. Kısacası tam zamanında üretim felsefesinde sıralama ve çizelgeleme, önemli bir yer tutmaktadır.

Literatürde yarım asrı aşkın zamandır sıralama ve çizelgeleme problemi ele alınmış ve zaman zaman, hazırlık sürelerinin işlem sürelerine dahil edilip edilmemesi veya göz ardı edilip edilmemesi noktasında da çalışmalar yapılmıştır.

Allahverdi ve diğerleri [2] 1999 yılında yaptıkları bir çalışmada, o güne kadarki literatürü incelemiş ve büyük bir çoğunluğunda, analizi kolaylaştırmak veya belirli uygulamaları yansıtmak için hazırlık zamanlarının proses zamanları içerisine dahil edildiğini veya tamamen göz ardı edildiğini belirtmişlerdir. Bu durum hazırlıkların açıkça ele alınması gereken çözüm kalitesini olumsuz yönde etkilemektedir. Zamana dayalı rekabet ve grup teknolojisi gibi üretim konseptlerinin ortaya çıkması ile ilgili

uygulamalar, çizelgeleme problemlerinde hazırlık zamanlarının göz önüne alınmasına sebep olmuştur.

Hazırlık zamanlarını göz önüne alan çizelgeleme problemleri son zamanlarda araştırmacıların daha çok ilgisini çekmektedir. İlerleyen teknolojiyle beraber işletmelerde, tek bir kaynak üzerinde birçok farklı ürün üretilmekte ve bu da hazırlık aktivitelerinin yapılmasını gerektirmektedir. Hazırlık aktiviteleri de genellikle üretim prosesleri içerisinde maliyetleri artırıcı bir rol almaktadır [3]. Liu ve Chang [4] üretimi çizelgelerken makine hazırlık zamanları ele alınmadığı takdirde, makine kapasitelerinde %20'den fazla israfın oluşacağına değinmişlerdir.

Hazırlık zamanlarının göz önüne alındığı problemler literatürde ikiye ayrılmıştır:

1. Sıra bağımsız hazırlık zamanları (Hazırlıklar sadece işlem görecekt işe bağlıdır.)
2. Sıra bağımlı hazırlık zamanları (Hazırlıklar hem o anda işlem gören işe, hem de bir önceki işe bağlıdır) [2].

Kimyasal madde, matbaa, tıbbi malzeme, otomobil üretimi gibi gerçek hayattaki durumlarda, temizlik veya takım değiştirme gibi hazırlık operasyonları aynı makinedeki önceki işlemlere bağlıdır [5]. Dolayısıyla bu sektörlerde hazırlık zamanının sıra bağımlı olduğunu söyleyebiliriz.

Çalışmanın 2. bölümünde, literatürdeki bu ayırmadan yola çıkılarak hazırlık zamanlarının sıra bağımlı olmadığı veya göz önüne alınmadığı durumlar ile hazırlık zamanlarının sıra bağımlı olduğu durumlar ayrı ayrı incelenmiştir.

Gıda üretim sektörü, üretilen gıda maddelerinin kısa sürede bozulması ve hurdaya ayrılması nedeniyle yığınlar halinde erken üretilip depolamanın mümkün olmadığı ve rekabet koşulları gereği tam zamanında üretime zorlanan sektörlerin başında gelmektedir. Çalışmada, tek makinenin çok fonksiyonlu kullanılması durumunda meydana gelen sıra bağımlı hazırlık zamanlı çizelgeleme probleminde, farklı teslim tarihli ürünlerin üretim planını da göz önünde bulundurarak, toplam ağırlıklı erken bitirme ve ağırlı geç bitirme enküçüklenmeye çalışılmıştır. Çalışmada incelenen problemi, $n/1/S_{ij}, d_i/F_{\min}(\sum \alpha_i.E_i + \sum \beta_i.T_i)$ şeklinde gösterebiliriz.

Literatürde bu tip hazırlık zamanları sıra bağımlı olan problemlerin gezgin satıcı problemine (TSP) benzetildiği örnekler söz konusudur [6-9]. Bu çalışmadaki problem de, sıra bağımlı hazırlık zamanlı olmasından dolayı gezgin satıcı problemine

benzetilebilir. Her bir iş, gezgin satıcı probleminde bir şehre karşılık gelirse, işler arası hazırlık süreleri de şehirlerarası mesafeler olarak düşünülebilir. Fakat amaç, gezgin satıcı problemindeki gibi bu sürelerinin toplamını (kat edilen mesafelerin toplamını) enküçüklemek değil, tam zamanlı üretim yapabilmek için erken ve geç üretim cezalarını enküçüklemektir. Bunun yanı sıra ele alınan problemde gezgin satıcı probleminden farklı olarak başlanılan noktaya geri dönme zorunluluğu da yoktur. Çünkü çizelge siparişlere göre belirlenmiş bir üretim planından da etkilenecektir.

Tek makineli sıra bağımlı hazırlık zamanlı çizelgeleme probleminde toplam tamamlanma zamanını enküçükleme problemi $n/1/S_{ij} / \Sigma C$, NP-zor yapıda olduğundan dolayı [10] ele aldığımız iki ölçütlü sıra bağımlı hazırlık zamanlı tek makineli çizelgeleme probleminin $n/1/S_{ij}, d_i / F_{\min}(\Sigma \alpha_i.E_i + \Sigma \beta_i.T_i)$ de NP-zor yapıda olduğunu söyleyebiliriz.

NP, bir karmaşıklık sınıfıdır ve NP-zor yapıda olan problemler ancak; küçük boyutlu olmaları halinde analitik modeller kullanılarak sonlu zamanda çözülebilirler, çünkü bu problemlerde en iyi çözümü bulmak için gerekli olan zaman, problem boyutuna bağlı olarak üstel artış gösterir. Büyük boyutlu NP-zor yapıdaki problemlerin çözümleri için meta-sezgisel yöntemler geliştirilmiştir.

Çalışmanın 3. bölümünde sıra bağımlı hazırlık zamanlı çizelgeleme problemlerinin çözümünde kullanılan meta-sezgisel yöntemler kısaca anlatılmış ve çalışmanın uygulama bölümünde de kullanılacak olan yasaklı arama algoritması (YAA) ve genetik algoritma (GA) detaylıca anlatılmıştır.

Çalışmanın 4. bölümünde yapılan araştırmalar sonucu, gıda sektöründe üretim yapmakta olan bir firmada hazırlık zamanlarının sıra bağımlı olduğu durumda bir çizelgeleme probleminin olduğu tespit edilmiş ve bu problem ele alınarak, çözüm için iki meta-sezgisel yöntemden (YAA ve GA) yararlanılmıştır. Elde edilen çözümler deneyler yapılarak istatistiksel olarak değerlendirilmiştir.

Çalışmanın 5. bölümünde bir örnek olay incelemesi yapılmıştır. Gıda sektöründe faaliyet gösteren bir firmanın taze peynir üretim tesisindeki çizelgeleme problemi incelenmiş, önerilen meta-sezgisel yöntemle çözümler üretilmiş ve üretilen çözümler istatistiksel açıdan değerlendirilmiştir.

Çalışmanın 6. bölümünde, tüm çalışmayı kapsayan sonuçlar, uygulama bulguları ve gelecek çalışmalar için öneriler sunulmuştur.

2. LİTERATÜR ARAŞTIRMASI

Bu bölümde üretimde çizelgeleme problemlerini konu alan literatürdeki bazı çalışmalar, öncelikle sıra bağımlı ve sıra bağımsız olarak iki kısımda, bunlar kendi içlerinde de üretim tipine göre paralel makineli, tek makineli, akış tipi ve atölye tipi olmak üzere dört kısma ayrılarak incelenmiştir.

2.1. Sıra Bağımsız Çizelgeleme Problemleri

2.1.1. Paralel Makineli Çizelgeleme Ortamı

Bruno ve Sethi [11] özdeş paralel makineli çizelgeleme problemlerinde toplam elde bulundurma maliyetini enküçüklemek için bir dinamik programlama algoritması önermişlerdir.

Aynı problemleri göz önünde bulunduran Tang [12] toplam tamamlanma zamanını enküçüklemek için bir sezgisel yaklaşım ve iki alt sınır önermiştir.

1993 yılında Cheng ve Chen [13] tarafından, toplam tamamlanma zamanı enküçüklendiği 2 özdeş paralel makineli çizelgeleme problemlerinin NP-zor olduğu gösterilmiştir.

Schutten ve Leussink [14] ise; en büyük gecikmeyi enküçüklemek için bir dal sınır algoritması ve biri işlerin tamamlanmadan kesilmesine izin veren iki alt sınır önermişlerdir.

2.1.2. Tek Makineli Çizelgeleme Ortamı

Glassey [15] 1967'de tek makineli ortamda hazırlık sayısını enküçükleyen bir dinamik programlama metodolojisi geliştirmiştir.

Shwimer [16] pozitif gecikmelerden kaynaklanan toplam ceza maliyetini en küçüklemek için bir dal-sınır algoritması önermiştir. Ceza fonksiyonunun doğrusal olduğu ve doğrusal olmadığı durumlara da değinmiştir.

Rinooy ve diğerleri [17] çalışmalarında n adet işin aynı anda bir makineye gelerek teker teker belirli işlem zamanlarında işlem gördüğünü ve her bir işin bir maliyet fonksiyonu olduğunu düşünerek toplam işlem maliyetini enküçükleyecek çizelgeyi geliştirmeyi amaçlamışlardır. Genel bir maliyet fonksiyonu için bir dal-sınır algoritmasında bir alt sınır önermişlerdir.

Baker [18] en büyük gecikmeyi enküçükleyen bir sezgisel algoritma önermiştir. Pan ve diğerleri [19] işleri sınıflara ayırarak, bir iş kümesindeki en büyük gecikmeyi enküçükleme amaçlı bir matematiksel model kurmuşlar ve bir sezgisel yöntem önermişlerdir.

Sourd ve Kedad-Sidhoum [20] teslim zamanı, erken ve geç bitirme maliyetleri belirgin olan tek makineli çizelgeleme problemlerinde maliyeti enküçüklemeyi amaçlayan bir alt sınır önermişlerdir. Önerilen alt sınır tabanlı dal-sınır algoritması ve bazı yeni baskınlık kurallarının deneysel testleri yapılmış, yeni metodun etkin olduğu gözlenmiştir.

Son zamanlarda Nazif ve Lee'nin [21] yaptıkları bir çalışmada, sıra bağımsız hazırlık zamanlarını göz önüne alarak toplam ağırlıklı tamamlanma zamanını enküçükleyecek tek makineli çizelgeleme problemlerini incelemişlerdir. Bu problemlerin çözümü için optimize edilmiş çaprazlama ile genetik algoritma önerilmiştir. Önerilen metot standart genetik algoritma ve çoklu çaprazlama genetik algoritmalarıyla (Multicrossover GA) karşılaştırılmış, daha iyi sonuç verdiği gözlemlenmiştir.

2.1.3. Akış Tipi Çizelgeleme Ortamı

İki makineli sıra bağımsız çizelgeleme problemleriyle ilgili ilk çalışma 1953 yılında Johnson [22] tarafından yapılmıştır. Üç makineli akış problemleri için de genişletilen ve literatürde Johnson algoritması olarak da oldukça bilinen bu algoritma işlerin tamamlanma zamanını enküçüklemeyi amaçlar.

Yine iki makineli akış tipi imalat ortamında, hazırlık zamanlarının işlem zamanına bağlı olmadığı ve önceki çalışmalardan farklı olarak stokastik makine arızalarının da göz önüne alındığı çizelgeleme problemlerinin de Allahverdi [23], Sule ve Huang'ın [24] iki ve üç makineli akış tipi problemlerin çözümünde önerdikleri algoritmayla çözülebildiğini göstermiştir.

Ben-Daya ve Al-Fawzan [25] akış tipi sıralama problemlerinin çözümü için bir tabu arama yaklaşımı geliştirmişlerdir. Çalışma daha önce yapılan çalışmalardan farklı olarak birleştirilmiş kuvvetlendirme ve çeşitlendirme şemalarını (intensification and diversification schemes) içermektedir. Tabu arama yönteminde başlangıç çözümünü elde etmek için NEH algoritması ve ekleme algoritması (insertion algorithm) kullanılmıştır. Yazarlar kendi geliştirdikleri algoritmayla Taillard'ın [26] yasaklı arama algoritmasını ve Obgu & Smith'in [27] tavlama benzetimi algoritmasını karşılaştırmışlardır. En iyi sonucu tavlama benzetimi algoritması vermiştir.

Yine Allahverdi [28] 2000 yılında sıra bağımsız hazırlık zamanlarının olduğu iki makineli akış tipi çizelgeleme problemlerinde ortalama akış zamanını enküçüklemeye çalışmıştır. Bu çalışmada, çözümler iki özel akış için elde edilmiştir. İki baskın ilişki geliştirilmiş ve geliştirilen ilişkilerin etkinliği dal-sınır algoritması üzerinde değerlendirilmiştir. Ayrıca üç tane de sezgisel algoritma tasarlanmış ve test edilmiştir.

Gupta ve Schaller [29] birbirinden bağımsız aile hazırlık zamanlarının imalat akış hücrelerinde olduğu, parça ailesi ve işlerin çizelgelendiği, pozitif gecikmeyi enküçüklemeyi amaçlayan problemlerin çözümleri üzerinde çalışma yapmışlar ve orta büyüklükteki problemleri çözen bir dal-sınır algoritması geliştirmişlerdir. Problemin genel çözümü için birkaç sezgisel algoritma geliştirilmiş ve bu algoritmaların etkinlik ve verimlilikleri deneysel olarak değerlendirilmiştir.

Yenisey ve Yağmağan [30] NP-zor permütasyon akış tipi çizelgeleme problemlerinde kullanılan karınca kolonisi eniyileme (ACO) meta-sezgisel yaklaşımıyla daha kaliteli sonuçlar elde edebilmek için parametre eniyileme üzerinde çalışmışlardır. Bu çalışma kapsamında 2 seviyeli bir deneysel tasarım uygulanmış ve problemin karınca kolonisi eniyileme algoritmasıyla çözümünde en iyi parametre değerlerinin kullanılmasının çözüm kalitesini önemli ölçüde etkilediği gözlenmiştir.

Yenisey ve Yağmağan [31] 2010 yılında yayınladıkları bir başka çalışmada yine literatürde NP-zor olarak bilinen çok amaçlı akış tipi çizelgeleme problemlerini ele almışlar ve bu problemin çözümü için yerel arama stratejisi ve karınca kolonisi eniyileme yaklaşımının birleşimi olan, toplam tamamlanma zamanı ve toplam akış zamanını eniyilemeyi amaçlayan, çok amaçlı bir karınca koloni sistemi algoritması geliştirmişlerdir. Geliştirilen algoritma Rajendran'ın CR, Ravindran ve diğerlerinin

geliştirdiği HAMC1, HAMC2, HAMC3 algoritmalarıyla ve çok amaçlı akış tipi çizelgeleme problemlerinde kullanılan GA ile karşılaştırıldığında daha iyi sonuç verdiği gözlenmiştir.

2.1.4. Atölye Tipi Çizelgeleme Ortamı

Laarhoven ve diğerleri [32] atölye tipi çizelgeleme problemlerinde tamamlanma zamanını enküçülemek için tavlama benzetimi tabanlı, rastsal yinelemelerle ilerleyen yeni bir yaklaşım önermişlerdir.

Colorni ve diğerleri [33] atölye tipi çizelgeleme problemleri için arama işlevi ve yeni bir sezgisel metot olan karınca kolonisi sisteminin kullanılabilirliğini ve başarılı çözümler üretebildiğini göstermişlerdir.

Sotskov ve diğerleri [34] sıra bağımsız hazırlık zamanlarının, spesifik teknolojik rotanın ve serbest bırakma tarihinin göz önüne alındığı, işlerin gruplara ayrıldığı, problemlerinin çözümü için ekleme tekniklerini, ışın arama (beam search) algoritması ile birleştirerek bir çözüm metodu önermişlerdir.

Gonçalves ve diğerleri [35] atölye tipi çizelgeleme problemlerinin çözümü için hibrit bir algoritma geliştirmişlerdir. Genetik algoritmada tanımlanan önceliklere göre çizelgeler oluşturulmuş ve elde edilen çizelgeleri geliştirmek için yerel arama sezgiselinden yararlanılmıştır. Önerilen metot literatürde var olan 43 farklı durumla ve 12 farklı metotla karşılaştırılmış, en iyi veya en iyiye yakın çözüm verdiği gözlenmiştir.

Hemamalini ve diğerleri [36] teslim zamanını göz önünde bulunduran karmaşık atölye tipi çizelgeleme problemlerinin çözümü için derin hafızalı aç gözlü arama (deep memory greedy search) metodu adını verdikleri sezgiseli geliştirmişlerdir. Önerilen metot farklı durumlar için analiz edildiğinde kombinatoryal (katışimsal) eniyileme problemlerinin çözümünde neredeyse en iyi sonuçları verdiği gözlemlenmiştir.

2.2. Sıra Bağımlı Çizelgeleme Problemleri

2.2.1. Paralel Makineli Çizelgeleme Ortamı

Deane ve White [37] sıra bağımlı hazırlık zamanlı, n adet paralel makine üzerinde işlem görebilen m adet bağımsız iş yığınının olduğu ortamda, iş yükü dengesini de

göz önünde bulunduran optimal çizelgeyi araştırmışlardır. Çalışmada, sayısal bir problem dal-sınır algoritmasıyla çözülmüş ve hazırlık maliyetleri enküçüklenerek dengeli ve dengesiz çizelgeleme arasındaki maliyet ödünleşiminden bahsedilmiştir.

Guinet [38] bir araç rotalama probleminde en büyük tamamlanma zamanını veya ortalama tamamlanma zamanını enküçükleyecek bir sezgisel algoritma üzerinde çalışmıştır. Önerilen algoritma Macar algoritmasının çok kaynaklı kullanımı için genişletilmiş halidir.

Lee ve Pinedo [39] sıra bağımlı hazırlık zamanlı, paralel makineli çizelgeleme problemlerinin çözümü için 3 fazlı bir sezgisel yöntem önermişlerdir. Bu yöntemde 2. fazda bir dağıtma kuralı ve buradan elde edilen sonuçla da 3. fazda tavlama benzetimi metodu kullanılmaktadır. Makalede bu metodun daha karmaşık problemlere çözüm üretebilmesi için bazı modifikasyonlara ihtiyaç duyduğundan bahsedilmektedir.

Yine Eom ve diğerleri [40] işlem zamanlarının, teslim tarihlerinin, ağırlıkların ve iş ailelerinin bilindiği paralel makineli çizelgeleme problemlerinde toplam ağırlıklı pozitif gecikmeyi en küçükmek için 3 fazlı bir sezgisel önermişlerdir. Kapsamlı simülasyon sonuçlarına göre önerilen algoritmanın hazırlık zamanlı belirli gecikme maliyeti (apparent tardiness cost with setup) ve döngüsel zaman prosedüründen (rolling horizon procedure) iyi sonuç vermiştir.

Paulo ve diğerleri [41] 2007 yılında sundukları bir çalışmada, tamamlanma zamanı ile ağırlıklı pozitif gecikmenin toplamını enküçükmek için değer komşuluk arama (value neighbourhood search) algoritmasını önermişlerdir. Önerilen algoritma açgözlü rastsal uyarlanabilir arama (greedy randomized adaptive search) algoritmasıyla, teslim zaman aralığının kısa, uzun, hem uzun hem kısa olduğu üç durum göz önüne alınarak karşılaştırılmış ve 60 işten fazla olan durumlarda önerilen algoritmanın daha iyi ortalama çözüm değerleri elde ettiği gözlenmiştir.

2.2.2. Tek Makineli Çizelgeleme Ortamı

Gavett [42] 1965 yılında yayınladığı gezgin satıcı probleminin biraz daha farklı bir halinin ele alındığı makalede, hazırlık zamanları sıra bağımlı olan tek makineli problemlerde, sıralama yapabilmek için 3 farklı sezgisel kural önerilmiştir. Önerilen sıralamaların rastsal sıralamaya göre daha etkili sonuç verdiği gözlenmiştir.

Lockett ve Muhlemann'ın [43] 1971 yılında yaptıkları çalışmada başlığı üzerinde m adet farklı takım olan bir makinede yapılacak işlerin ihtiyaç duyduğu takımların değişim sayısının makine üzerindeki bir önceki işe bağlı durumları incelenmiştir. Sayısal bir örnekle problem gezgin satıcı problemine benzetilmiştir. (Bu örnekte her bir işin yapılması için her bir takım istasyonunda farklı takımlara ihtiyaç duyulduğu düşünülmüştür.) Problemin çözümü için bir dal-sınır algoritması önerilmiştir. (Bu algoritma işler arası geçişlerdeki değişim sayılarının verildiği bir p matrisi üzerinden çalışmaktadır.)

White ve Wilson [44] çalışmalarında hazırlık operasyonlarını sınıflandıran ve hazırlık sürelerini tahmin eden bir yöntemden bahsetmişlerdir. Önerilen bu sezgisel çizelgeleme prosedürü, toplam hazırlık zamanını enküçüklemeyi amaçlayan, çok karmaşık hesaplara gerek duymayan, elle çözmek için oldukça pratik ve anlaşılması, kullanılması kolay bir metottur. (Önerilen metot sıralamaya yeni bir işin de dâhil edilmesine izin verir.)

Ragatz [45] pozitif gecikmeyi enküçükleyecek çizelgeleme üzerinde çalışmıştır. Problem için bir alt sınır geliştirilmiş ve dal sınır algoritmasının çözümünün bir parçası olarak kullanılmıştır. Önerilen metodun performansı 8-16 işe kadar test edilmiştir. Yapılan testler sonucu algoritmanın sınırlamasının zayıf olduğu, baskınlık testlerinin kullanımının dal sınır algoritmasının performans artışı, önemli derecede geliştirdiğinden söz edilmiştir.

França ve diğerleri [46] toplam pozitif gecikmeyi enküçükleyecek teslim tarihli ve hazırlık zamanlı problemlerin çözümü için yeni bir memetik algoritma önermişlerdir. Memetik algoritma çalışmada, ad hoc yerel arama prosedürünün kuvvetlendirme gücü ile hiyerarşik popülasyon yaklaşımının gücü birleştirilerek uygulanmıştır. (memetik algoritmalar kuadratik atama, çizelgeleme ve zaman programlama gibi birçok NP optimizasyon problemleri için kullanılmaktadırlar.) Memetik algoritma ve saf genetik algoritma, yapıcı iki sezgiseli çalıştıran all-pairs komşuluk arama algoritmasıyla karşılaştırılmıştır. Diğer algoritmalarla kıyaslama yapıldığında RRX (crossover) çaprazlamasının kullanıldığı memetik algoritma daha başarılı olmuştur.

Cheng ve diğerleri [47] 2003 yılında yaptıkları çalışmada Bruno ve Downey'in 1978 yılında hazırlık zamanlarını göz önünde bulundurarak en büyük gecikmeyi enküçüklemeyi amaçlayan tek makineli çizelgeleme problemlerinin NP-Hard

olduğunu göz önüne alarak, Ghosh ve Gupta'nın da 1997 yılında bu tip problemler için geliştirdiği dinamik algoritmaya atıfta bulunmuştur. Bunlara bağlı kalarak montaj işlemlerinin olduğu bir yığın üretiminde, tek makineli bir çizelgeleme yaparken hazırlık zamanlarını da göz önüne bulundurarak, toplam gecikmenin sıfırlanması probleminin de NP-zor olduğunu ispatlamıştır. Yine aynı problemin polinomiyal zaman sınırı olmayan bir NP-zor olduğunu da ispatlamıştır.

Sourd [48]'un çalışmasında tek makineli n adet iş yükü olan, kesintiye izin vermeyen fakat boş zamana (idle time) izin veren, hazırlık zamanlarının dikkate alındığı, gecikme ve erken bitirmenin cezalandırıldığı optimum çizelgeyi bulmak için önce bir karışık tamsayılı (mixed integer) model önermiştir. Bu modelin daha uzun işlem zamanlı problemlere uygulanabilmesi için doğrusal genişletme önerilmiştir. Çalışmada hazırlık zamanlarının göz ardı edildiği bir dal-sınır algoritmasından da söz edilmiştir. (Alt sınır ve baskınlık kuralları önerilmiştir.) Ancak; hazırlık zamanlarının göz önüne alınmasının önemine değinen yazar sezgisel bir metot da önermiştir.

Choobineh ve diğerleri [49] n adet işin bir makinede m amaca göre sıralanması problemini ele almışlardır. Çalışmada problem gezgin satıcı problemine benzetilerek yorumlanmış ve üç amaçlı karışık tamsayılı bir doğrusal algoritma önerilmiştir. Fakat amaçlar üçten fazla olduğunda bu algoritmanın çözümü aşırı derecede zorlaşacağından kapsamlı problemlerin çözümü için üç özel niteliği olan bir yasaklı arama algoritması önerilmiştir. Bu nitelikler; m paralel tabu listesi kullanması, her bir amacın ağırlığını göstermesi, sınırlandırılmış bir çözüm uzayı yaratmasıdır. Üç amaçlı bir problem önerilen doğrusal modelle çözüldükten sonra, önerilen sezgiselle çözülmüş ve sezgiselin en iyi veya en iyiye yakın sonuç verdiği gözlenmiştir. Başka bir gözlem de, hazırlık aralığı azaldıkça hesaplama zamanının arttığıdır. Bu durum da, hazırlık aralığının daraltılması durumunda çözüm uzayında daha fazla çözümün değerlendirilmesiyle açıklanmıştır.

Gupta ve Smith [50] toplam pozitif gecikmeyi enküçüklemek için uzay tabanlı yerel arama sezgiselini (space-based local search heuristic) ve açgözlü rastsal uyarlanabilen arama prosedürünü (greedy randomized adaptive search) önermişlerdir. Önerilen algoritmalar, literatürdeki test problemleri üzerinden genetik algoritma, tavlama benzetimi, karınca kolonisi eniyilemesi, dal-sınır algoritması ve ikili değiştirme yöntemiyle karşılaştırılmıştır. Sonuçlar değerlendirildiğinde, uzay tabanlı yerel arama sezgiseliyle karınca kolonisinden elde edilen değerler birbirine

oldukça yakındır. Açgözlü rastsal uyarlanabilen arama prosedürü ise, karınca kolonisinden daha uzun süre almasına rağmen en iyiye daha yakın çözümler vermiştir. Başka bir sonuca göre de, önerilen algoritmalar diğer algoritmalarından daha dayanıklıdır. (20 bağımsız koşum sonrası, önerilen algoritmaların en kötü değerleri gözle görünür biçimde diğerlerinin en kötü değerlerinden daha iyidir. Bu da önerilen algoritmaların daha tutarlı olduğunu gösterir.)

Lin ve Ying [51] toplam ağırlıklı gecikmeyi enküçüklemek için tavlama benzetimi, genetik algoritmalar ve yasaklı arama metodu üzerinde ayrı ayrı çalışmışlardır. Tavlama benzetimine rassal değiş tokuş (random swap) ve ekleme (insertion) aramaları eklenmiştir. Genetik algoritmanın mutasyon operatörü açgözlü yerel arama (greedy local search) ile çalıştırılmıştır. Yasaklı arama algoritmasına da rastsal değiş tokuş ve tabu arama listeleri adapte edilmiştir. Geliştirilen yaklaşımlar 120 problem üzerinde karşılaştırılmış, hesaplanan sonuçlara göre önerilen yaklaşımların toplam cezaları düşürdüğü gözlenmiştir ve buradan hareketle önerilen yaklaşımlarla çözülen birçok karşılaştırma problemi için yeni bir üst sınır tanımlanmıştır.

Eren ve Güner [10] hem toplam tamamlanma zamanını hem de enbüyük erken bitirmeyi enküçüklemeyi amaçlayan bir 0-1 karışık tamsayılı programlama modeli önermişlerdir. Çalışmada bu model 12 işe kadar olan problem setlerini çözebildiği için daha büyük boyutlu problemlerde, tek makineli problemlere uyarlanmış NEH algoritması, yasaklı arama ve rastsal arama algoritmaları kullanılmıştır. 1000 işe kadar olan problem setleri bu sezgisellerle çözülmüş ve elde edilen sonuçlar önerilen yasaklı arama algoritmasının daha iyi sonuç verdiğini göstermiştir.

Liao ve Juan [52] ağırlıklı pozitif gecikmeyi enküçüklemek için bir karınca kolonisi eniyileme algoritması önermişlerdir. Önerilen algoritmanın iki ayırıcı özelliği vardır. Bunlar; başlangıç feromon yolu (initial pheromone trail) için yeni parametreler içermesi ve yerel arama uygulamaları için zamanlamayı değiştirmedir. Bu özellikler sayesinde algoritma genetik algoritmalar ve tabu-vns ile karşılaştırıldığında iyi performans gösterdiği söylenebilir.

2.2.3. Akış Tipi Çizelgeleme Ortamı

Corwin ve Esogbue [53] iki makineli akış tipi imalatta, makinelerin birinin sıra bağımlı hazırlık zamanıyla işlem yaptığı iki farklı çizelgeleme problemini ele almışlardır. Amacı toplam tamamlanma zamanını enküçüklemek olan birer dinamik

programlama formülasyonu oluşturmuşlardır. İki problem de göz önüne alınarak önerilen algoritma, dal-sınır yaklaşımla karşılaştırılmıştır.

Sule ve Huang [24] her bir işin önce 1. makinede, sonra 2. makinede işlem gördüğü bir ortamda, hazırlık, işlem ve sökme takma zamanlarını ayrı düşünerek bir çizelgeleme yapmışlardır. Çalışmada aktiviteler sıra bağımlı ve sıra bağımsız olarak düşünülmüştür. Stokastik olarak gerçekleşen makine duruşlarını da göz önüne alan, toplam tamamlanma zamanını enküçükleyen iki sezgisel algoritma önerilmiştir.

Gupta [54] 1986 yılında toplam tamamlanma zamanının enküçüklendiği durumu asimetrik bir gezgin satıcı problemi gibi modellemiş ve bu problemlerin NP-zor olduğunu göstermiştir. 1995 yılında Gupta ve diğerleri [55] yine aynı tip problemlerde, aynı amaca ulaşmak için bir dal-sınır algoritması geliştirmişlerdir.

Aldowaisan ve Allahverdi [56] beklemenin olmadığı ve hazırlık zamanlarının işlem süresinden ayrılmış olduğunu birlikte düşünerek, performans ölçütünün de toplam akış süresi olduğu bir çizelgeleme problemi üzerinde çalışmışlardır. (İki makineli akış tipi imalat düşünülmüştür.) Bu problemin çözümü için bir sezgisel algoritma önerilmiş ve test edilmiştir. Testler sonucu, önerilen çözüm algoritmasının iyi sonuç verdiği söylenmektedir.

Ruiz ve diğerleri [57] toplam tamamlanma zamanını enküçüklemeyi amaçlayan iki gelişmiş genetik algoritma üzerinde çalışmışlardır. Tecrübe tasarımı (design of experiment) yaklaşımıyla genetik algoritmanın parametrelerinde ve operatörlerinde düzenlemeler yapmışlardır. Önerilen meta-sezgiselleri daha önceden literatürde var olan birçok algoritmayla karşılaştırmışlar ve açık bir şekilde üstün olduklarını göstermişlerdir.

Zandieh ve diğerleri [58] literatürde hazırlık zamanı sıra bağımlı hibrit akış tipi çizelgeleme problemi olarak bilinen, kimya, tekstil, otomobil,...vs gibi sektörlerde sıkça karşılaşılan çizelgeleme problemi üzerinde çalışmışlardır. İmalattaki karmaşıklık arttıkça, çizelgeleme de bir o kadar zorlaşmaktadır. Bunun üzerine yazarlar, karmaşık problemlerin ele alınması ve kabul edilebilir bir zamanda uygun çizelgenin oluşturulması için bir bağımsızlık sistemi algoritması önermişlerdir. Önerilen algoritma rastsal anahtar genetik algoritmasıyla (random key genetic algorithm) ile karşılaştırılmış ve daha iyi çalıştığı gözlenmiştir.

2.2.4. Atölye Tipi Çizelgeleme Ortamı

Wilbrecht ve Prescott [59] çalışmalarında farklı iş sıralamalarının atölye tipi üretim performansına olan etkisini tanımlamak için bir atölye simüle etmişlerdir. Şu 7 farklı kurala göre sıralama yapılmıştır; rassal, EDD, hazırlık zamanı kısa olan öncelikli, koşum zamanı kısa olan öncelikli, SPT, koşum zamanı uzun olan öncelikli, LPT. (koşum zamanı: run time) İşlem zamanını koşum zamanı ve hazırlık zamanının toplamı olarak düşünmüşlerdir. Bu simülasyonların sonuçlarına göre LPT ve SPT'nin koşum zamanı uzun olan ve kısa olan öncelikli kurallarına göre daha iyi sonuçlar verdiğini, hazırlık zamanının da performans için önemli rol teşkil ettiği sonucuna varmışlardır.

Srikar ve Ghosh [60] sıra bağımlı hazırlık zamanlı, n adet iş ve m adet makinenin olduğu bir akış tipi atölyede çizelgeleme problemi ele almışlardır. Bu problem için, amacı toplam tamamlanma zamanını enküçüklemek olan karışık tamsayılı bir model önermişlerdir. Önerilen model, 6 iş ve 6 makineli problemi başarıyla çözebilmiştir.

Kim ve Bobrowski [61] çalışmalarında dinamik atölye tipi imalat, sıra bağımlı hazırlık sürelerinin imalat performansı üzerindeki etkisini araştırmışlardır. 9 makineli bir atölyenin verileri üzerinde bir simülasyon yapılmış ve kuramsal, asimetrik bir hazırlık zamanları matrisi kullanılmıştır. En küçük kritik oranın işi (job of smallest critical ratio), benzer hazırlık, kritik oran, SPT gibi sıralama kuralları, iş başına düşen hazırlık zamanı, iş başına düşen ortalama akış zamanı, yüzde gecikmesi,...vs gibi performans ölçütlerine göre simülasyonlar yapılarak değerlendirilmiştir. Simülasyonda teslim sürelerinin sıklığı (esnek olup olmadığı) da belirtilmektedir ve sonuca göre, eğer işler sıra bağımlı ise teslim sürelerinin esnekliğinin belirtilmesi, işlerin zamanında yetiştirilme performansını artırmaktadır. Ayrıca hazırlık zamanına bağlı sıralama kuralları (benzer hazırlık, en küçük kritik oranın işi) diğer kurallara göre daha verimli sonuç vermektedir.

Low [62], hazırlık zamanlarının sıra bağımlı olduğu atölye tipi çizelgeleme problemlerinde, daha gerçekçi düşünerek çoklu makineler, çoklu kaynaklar ve alternatif rotaların yer aldığı karmaşık kısıtları da modellediği çalışmada, 3 farklı sezgisel yöntemle çizelgeleme yapmıştır.

Cheung ve Zhou [63] 2002 yılında yayınladıkları makalede sıra bağımlı hazırlık zamanlı genel atölye tipi çizelgeleme problemleri için bir karışık tamsayılı doğrusal programlama algoritması geliştirmiş ve genetik algoritma ve sezgisel kuralların entegrasyonu ile oluşturulan bir hibrit algoritma önermişlerdir. (genetik algoritmanın optimizasyon prosedürü, sezgisel kuralları daha etkili kılarken, sezgisel kurallar, genetik algoritmayı daha verimli kılmaktadır.) Önerilen hibrit algoritma daha önceden yayınlanan algoritmalarından ve sezgisel kurallardan daha iyi sonuç vermektedir.

Choi ve Choi [64] çizelgelenen bir iş için alternatif operasyonların olduğu ve hazırlık sürelerinin sıra bağımlı olarak değiştiği, toplam tamamlanma zamanının enküçüklenmesinin istendiği atölye ortamını göz önüne almışlardır. Çalışmada problemin çözümü için yerel arama şemasıyla birlikte bir karışık tamsayılı arama modeli önerilmiştir. Yerel arama şeması kullanılarak hesaplama süresinin azaltılması düşünülmüştür. Buna ek olarak çeşitli aç gözlü arama (greedy) tabanlı dağıtma kurallarının (greedy-based dispatching rules) performansının yerel arama şemasıyla geliştirilebileceği gözlenmiştir.

Zhou ve diğerleri [65] çalışmalarında toplam tamamlanma zamanını enküçükleyecek, biyolojik bağışıklık mekanizmasını baz alan bir metot geliştirmişlerdir. Amaç fonksiyonunu; toplam tamamlanma zamanı ve kısıtları; sıra bağımlı hazırlık zamanları, proses rotası,..vs gibi antiijenler olarak düşünmüşlerdir. Önerdikleri algoritmayı bir örnekle açıklamışlardır.

Naderi ve diğerleri [66] yakın tarihteki çalışmalarında bugüne kadar imalat çizelgeleme ve koruyucu bakıma katkı olarak yapılan çalışmaların çoğunun uygulamaya elverişsiz olduğunu savunmaktadırlar. Buradan yola çıkarak hazırlık zamanlarını ve koruyucu bakım faaliyetlerini göz önüne alan, tavlama benzetimi ve genetik algoritmalara dayanan 4 meta-sezgisel ve literatürde yer alan 2 farklı meta-sezgiselin uyarlanmasıyla problem çözümü yapılmıştır. Koruyucu bakımdan ve belli bir zaman aralığında aylak zamanı (idle time) da içerebilen bir teknik ile optimum zaman aralığında aylak zamana izin vermeyen 2 teknikten bahsedilmiştir. (GA, tavlama benzetimi, SPT-GA, SPT-benzetilmiş tavlama, GA-Ch ve Bağışıklık sistemi-YZ) Sonuçta performansların karşılıklı kıyaslanmasıyla genetik algoritmanın SPT ile hibritleşmiş metodunun diğerlerinden mutlak üstünlüğü ispatlanmıştır.

Moghaddas ve Houshmand [67] çalışmalarında, belirli ve ifade edilen varsayımlar altında karışık tamsayılı doğrusal bir matematik model kurmuşlardır. Bu model, 5 makine ve 20 işlik bir çizelgeleme problemine çok uzun zamanda yanıt vermiştir. Bu yüzden böyle problemlere daha makul zamanlarda uygun çözümler veren bir sezgisel metot geliştirilmiştir. Literatürde sonucu kesin olan bir probleme uygulanan sezgisel metodun verdiği olurlu sonuç, problemin en iyi çözümüne eşit çıkmıştır. Yine de, en iyi sonuca varmanın mümkün olmadığı büyük problemlerde, herhangi bir sezgisel veya meta-sezgisel metodu değerlendirmek için üç farklı alt sınır (lower bound) oluşturulmuştur. Bu alt sınırlardan biri iş tabanlı (en küçük tamamlanma zamanını hesaplayan), diğer ikisi makine tabanlıdır (bir makinedeki tüm işlerin toplam tamamlanma zamanını hesaplayan).

Naderi ve diğerleri [68] hazırlık zamanının sıra bağımlı olduğu atölyelerdeki çizelgeleme problemi için en büyük tamamlanma zamanını enküçükleyecek, çeşitlendirme mekanizmasıyla hibritleşmiş bir genetik algoritma önermişlerdir. Bu genetik algoritmanın çeşitli operatörleri ve parametreleri, tekrarlamayı azaltmak ve çeşitliliği artırmak için Taguchi metodu ile düzenlenmiştir. Geliştirilen algoritma, GA-Cheung, Bağışıklık sistemi-Zhou (immune algorithm of Zhou), SPT ve saf GA ile karşılaştırılmıştır. Bu karşılaştırmada geliştirilen algoritmanın üstünlüğü gözlenmiştir.

Naderi ve diğerleri [69] son zamanlarda yaptıkları bir çalışmada, toplam tamamlanma zamanını en küçükleyen, etkin çalışan bir meta-sezgisel algoritma geliştirmişlerdir. Bu algoritma, yeni operatörlerle tavlama benzetimi yöntemini kullanarak çözüm yapmaktadır. (tavlama benzetimi yönteminin performansı, komşuluk arama yapısındaki operatörlere ve parametrelere bağlıdır.) Çalışmada birçok operatörün ve parametrenin tavlama benzetimi üzerindeki etkisi araştırılmıştır. (Başlangıç çözümü, soğutma çizelgesi, başlangıç sıcaklığı ve komşuluk arama yapısı,...vs.) Önerilen algoritma, sayısal bir örnekle, daha önceden var olan algoritmalarla karşılaştırılmış ve sonuçlar, önerilen algoritmanın diğer algoritmalarından daha iyi çalıştığını göstermektedir.

3. ÇİZELGELEME PROBLEMLERİNDE KULLANILAN META-SEZGİSEL YÖNTEMLER

Üretim girdilerinin en etkin şekilde kullanılması, tam zamanında üretimin uygulanması, ara stok ve ürün stoklarının mümkün olan en düşük seviyeye indirilmesi ya da tamamen ortadan kaldırılması gibi üretim ve hizmet sistemlerinin her aşamasında karşılaşılan problemler, aslında birer sırlama ve çizelgeleme problemidir. Sırlama ve çizelgeleme problemlerine ilişkin geliştirilen çözüm yöntemleri, analitik yaklaşımlar ve sezgisel yaklaşımlar olmak üzere iki farklı sınıf altında toplanmaktadır [70].

Sıralama ve çizelgeleme problemlerinde ölçütler ve boyut arttıkça karmaşıklık artmakta ve buna bağlı olarak çözüm zamanı da üstel olarak artmaktadır. Problemin boyutu ve etkili çözüm şemalarının olmaması sezgisel yaklaşımlara doğan ihtiyacı ortaya çıkarmaktadır. Bir sezgisel yaklaşım rasyonel bir metodu kullanmakla beraber en iyiye yakın iyi bir çözüm bulmayı dener [71], yani en iyi çözümü bulmayı garanti edemez. Sezgisel yöntemler şöyle sınıflandırılır:

- i. Kurucu Algoritmalar
- ii. Geliştirici Algoritmalar
- iii. Melez Algoritmalar
- iv. Graf Teorisi Algoritmalar [72].

Genel eniyileme sezgiselleri, amaç fonksiyon değerinin kötüleşmesine neden olan geçici hareketleri kabul ederek yerel en iyiye yakalanmaktan kurtulabilmektedirler. Bu gibi sezgiseller genellikle meta-sezgiseller olarak adlandırılmaktadır [73].

Ancak, sezgisel yöntemler ile meta-sezgisel yöntemler ya da yaklaşımlarını ayırt etmek gereklidir. Sezgisel ve meta-sezgisel yöntemlerin pek çok ortak yanları olmalarına karşın, çözüm uzayının araştırılması için kullanılan stratejileri farklıdır. Meta-sezgiseller başlangıç çözümüne ve değişim metoduna çok fazlasıyla bağımlıdır [72].

Bu çalışmada incelenen problem NP-zor sınıfı bir problemdir ve çözümü için meta-sezgisellerden yararlanılmıştır. Yapılan literatür araştırmasına göre, çizelgeleme problemlerinin çözümlerinde kullanılan meta-sezgisel yöntemler:

- i. Tavlama Benzetimi
- ii. Yasaklı Arama Algoritması
- iii. Genetik Algoritmalar
- iv. Karınca Kolonisi Eniyileme Algoritması
- v. Yapay Bağışıklık Sistemi
- vi. Parçacık Sürü Eniyileme Algoritması
- vii. Yapay Sinir Ağları

Çizelge 3.1’ de çizelgeleme konusunda literatürde yer alan bazı çalışmalar ve ele alınan problemlerin meta-sezgisel çözüm yöntemleri tarihsel sıralamaya uygun olarak örneklendirilmiştir.

3.1. Tavlama Benzetimi

Tavlama benzetimi algoritması, statik fizikte bir katı maddenin enerji durumunu en aza indiren tavlama süreci ve kombinatoryel eniyileme problemlerinin çözüm sürecine benzerliğine dayanarak ilk kez Kirkpatrick ve diğerleri [81] tarafından 1983 yılında geliştirilmiş bir stokastik arama algoritmasıdır. Tavlama süreci, katıların belirli bir başlangıç sıcaklığından başlayarak uygun kristal yapı elde edilinceye kadar yavaş yavaş soğutulması prensibine göre çalışır. Sistemde azaltılmak istenen enerji, eniyileme problemlerindeki en aza indirilmek istenen amaç (maliyet) fonksiyonuna karşılık gelir.

Algoritma, aramaya rastsal olarak belirlenmiş bir başlangıç çözümünden başlar. Hesaplama adımlarında mevcut çözümün komşu çözümleri uygun komşuluk mekanizmalarından yararlanılarak elde edilir. Üretilen komşu çözümün amaç değeri hesaplanarak mevcut çözümden daha iyi ise yeni çözüm olarak kabul edilir. Daha kötü olması halinde belirlenen kritere göre kabul veya reddedilir.

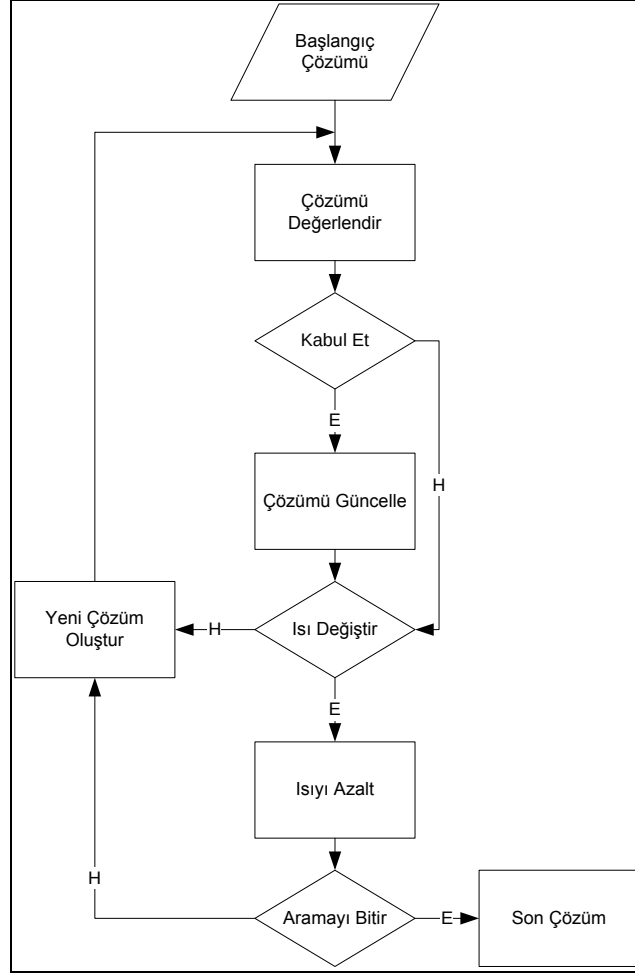
Hesaplama adımları sonunda sıcaklık belirli bir fonksiyona göre azaltılır ve bu işlem önceden belirlenen iterasyon sayısı veya önceden belirlenen sıcaklık değerine ulaşana kadar devam eder [82]. Şekil 3.1’de tavlama benzetiminin akış şeması görülmektedir [83].

Çizelge 3.1: Çizelgeleme problemlerinde kullanılan meta-sezgisel yöntemler.

Ref. No	Yayın Yeri	Yayın Yılı	Yayın Türü	Yazar	Yayın Adı	Çözüm Yöntemi
[74]	Control Engineering Practice	1994	Makale	Williems, M., Rooda, J. E.	Neural Networks for Job-Shop Scheduling	Yapay Sinir Ağları
[75]	European Journal of Operational Research	1996	Makale	Sabuncuoglu, İ., Gürgün, B.	A Neural Network Model for Scheduling Problems	Yapay Sinir Ağları
[25]	European Journal of Operation Research	1998	Makale	Ben-Daya, M., Al-Fawzan, M.	A Tabu Search Approach for The Flow-Shop Scheduling Problem	Yasaklı Arama Alg.
[76]	Generation, Transmission and Distribution, IEE Proceedings	2002	Konferans Yayını	Liang, R.-H., Hsu, Y.-Y.	Scheduling of Hydroelectric Generation Using Artificial Neural Networks	Yapay Sinir Ağları
[63]	Annals of Operations Research	2002	Makale	Cheung, W., Zhou, H.	Using Genetic Algorithms and Heuristics for Job-Shop Scheduling with Sequence Dependent Setup Times	Genetik Alg.
[19]	AT&T Labs Research Technical Report	2002	Makale	Gonçalves, J., Mendes, J., Resende, M.	A Hybrid Genetic Algorithm for Job-Shop Scheduling	Genetik Alg.
[35]	Computers & Operations Research	2004	Makale	Ying, K.-C., Liao, C.-J.	An Ant Colony System for Permutation Flow-Shop Scheduling	Karınca Kolonisi Algoritması
[49]	European Journal of Operational Research	2006	Makale	Choobineh, F. Fred, Mohebbi, E., Khoo, H.	A Multi-Objective Tabu Search for A Single Machine Scheduling Problem with Sequence Dependent Setup Times	Yasaklı Arama Algoritması
[50]	European Journal of Operational Research	2006	Makale	Gupta, Skylab L., Smith, Jeffrey S.	Algorithms for Single Machine Total Tardiness Scheduling with Sequence Dependent Setups	Genetik Algoritma Benzetilmiş Tavlama Karınca Kolonisi Algoritması
[65]	Int. J. Adv. Manuf. Technol.	2006	Makale	Zhou, Y., Li, B., Yang, J.	Study on Job-Shop Scheduling with Sequence-Dependent Setup Times Using Biological Immune Algorithm	Yapay Bağışıklık Sistemi Algoritması
[77]	Applied Mathematics and Computation	2006	Makale	Lian, Z., Jiao, B., Gu, X.	A Similar Particle Swarm Optimization Algorithm for Job-Shop Scheduling to Minimize Makespan	Parçacık Sürü Algoritması

Çizelge 3.1: Çizelgeleme problemlerinde kullanılan meta-sezgisel yöntemler (devamı).

Ref. No	Yayın Yeri	Yayın Yılı	Yayın Türü	Yazar	Yayın Adı	Çözüm Yöntemi
[51]	International Journal of Advanced Manufacturing Technology	2007	Makale	Lin, S.-W., Ying, K.-C.	Solving Single-Machine Total Weighted Tardiness Problems with Sequence-Dependent Setup Times by Meta-Heuristics	Genetik Algoritma Tavlama Benzetimi Yasaklı Arama Algoritması
[10]	Erciyes Üniversitesi FBE Dergisi	2007	Makale	Eren, T., Güner, E.	Sıra- Bağımlı Hazırlık Zamanlı İki Ölçütlü Çizelgeleme Problemi:Toplam Tamamlanma Zamanı ve Maksimum Erken Bitirme	Yasaklı Arama Algoritması
[78]	Computers & Operations Research	2008	Makale	Liao, C.-J., Tseng, C.-T., Luarn, P.	A Discrete Version of Particle Swarm Optimization for Flow-Shop Scheduling Problems	Parçacık Sürü Algoritması
[79]	Computers & Operations Research	2008	Makale	Pan, Q.-K., Tasgetiren, M.F., Liang, Y.-C.	A Discrete Particle Swarm Optimization Algorithm for the No-Wait Flow-Shop Scheduling Problem	Parçacık Sürü Algoritması
[66]	Int. J. Adv. Manuf. Technol.	2008	Makale	Naderi, B., Zandieh, M., Fatemi Ghomi, S.M.T.	Scheduling Sequence-Dependent Setup Time Job-Shops with Preventive Maintenance	Genetik Algoritma Tavlama Benzetimi
[80]	Computers & Industrial Engineering	2008	Makale	Yağmağan, B., Yenisey, Mehmet M.	Ant Colony Optimization for Multi-Objective Flow- Shop Scheduling Problem	Karınca Kolonisi Algoritması
[68]	International Journal of Production Research	2009	Makale	Naderi, B., Zandieh, M., Fatemi Ghomi, S.M.T.	Scheduling Job-Shop Problems with Sequence-Dependent Setup Times	Genetik Algoritması
[21]	European Journal of Scientific Research	2009	Makale	Nazif, H., Lee, L. S.	A Genetic Algorithm on Single Machine Scheduling Problem to Minimize Total Weighted Completion Time	Genetik Algoritması
[69]	Applied Soft Computing	2010	Makale	Naderi, B., Fatemi Ghomi, S. M. T , Aminnayeri, M.	A High Performing Meta-Heuristic for Job-Shop Scheduling with Sequence-Dependent Setup Times	Tavlama Benzetimi



Şekil 3.1: Tavlama benzetimi algoritmasının akış şeması.

3.2. Yasaklı (Tabu) Arama Algoritması (YAA)

Yasaklı arama algoritması (tabu arama algoritması) ilk kez Glover [84,85] tarafından 1989 yılında ortaya konulmuş, kombinatoriyel problemlerde yerel en iyiye yakalanmadan, en iyi veya en iyiye en yakın çözüme ulaşmak için kullanılan meta-sezgisel bir yöntemdir. Bu yöntemde aramanın yerel bir en iyiye yakalanıp kısır bir döngüye girmemesi için, daha önce ziyaret edilmiş ya da herhangi bir nedenle ziyaret edilmesi istenmeyen aday çözümler, tabu listesi adı verilen bir listede tutulur ve yöntem bu listede tutulan hamlelerin yapılmasını bir süre için yasaklar.

3.2.1. Yasaklı Arama Algoritmasının Temel Bileşenleri

3.2.1.1. Kısa Dönemli Hafıza (Tabu Listesi)

Algoritmanın işlemlere devam etmesi sürecinde bazı seçimlere izin verilmez ve yasaklanır. Yasaklanan bu seçimler tabu olarak isimlendirilir ve diğerlerinden ayırt edilir. Yasaklı seçimlerin tutulduğu hafızaya, yani listeye tabu listesi denir. Tabu listesinin hafızası, problemin çözüm performansını etkileyen bir parametredir. DeneySEL sonuçlar tabu listesi hafızasının, problemin büyüklüğüyle orantılı olarak büyümesi gerektiğini göstermektedir. Algoritmanın işleyişi sırasında bu hafıza dolduğunda, hafızada en uzun kalan hareket silinerek seçilmesine izin verilir. (İlk giren ilk çıkar yöntemi uygulanır)

3.2.1.2. Aspirasyon (Tabu Yıkma) Kriteri

Algoritmanın temel özellikleri korunurken esnekliğini artırmak, yerel en iyi çözümden kaçınmak ve kısır döngüden kurtulmak maksadıyla bir seçimin yasak konumundan geçici olarak serbest bırakılması için belirlenen kriterdir.

3.2.1.3. Uzun Dönemli Hafıza

Araştırmanın genel çeşitlendirmesini ve yerel yoğunlaştırılmasını sağlamak için kullanılır. Bulunan en iyi ve elit çözümleri tutar.

3.2.1.4.Hareket Mekanizması

Mevcut bir çözümde yapılan bir değişiklik ile yeni çözümün elde edilmesi hareket mekanizmasıyla gerçekleştirilir. Hareket mekanizmasındaki olası hareketler, mevcut çözümün komşularını oluşturur. Hareket mekanizması algoritmanın etkinliği açısından önemli olduğu için problemin yapısına bağlı olarak uygun bir şekilde seçilmelidir [86].

3.2.1.5. Durdurma Koşulu

Tabu arama algoritmasının iterasyonları, durdurma koşulu sağlanana kadar devam eder. Bu koşullar [86];

- i. Seçilen bir komşu çözümün komşusunun olmaması
- ii. Belirli bir iterasyon sayısına ulaşılması
- iii. Belirli bir çözüm değerine ulaşılması

iv. Algoritmanın bir yerde tıkanması ve daha iyi sonuç üretememesi

3.2.2. Yasaklı Arama Algoritmasının İşlem Adımları

1. Yasaklı arama algoritması, seçilen herhangi bir başlangıç çözümü ile aramaya başlar, diğer meta-sezgisel yöntemlerde olduğu gibi iyi bir başlangıç çözümü genellikle daha iyi sonuçların elde edilmesine imkan verir.
2. Başlangıç çözümü ilk adımda en iyi çözüm olarak düşünülür. Daha sonra, tanımlanan hareket mekanizmasına göre mevcut çözümün komşu çözümleri oluşturulur.
3. Elde edilen çözümler kümesinde en iyi amaç değerine sahip olan çözüm, eğer tabu listesinde değilse yeni mevcut çözüm olarak seçilir ve tabu listesine eklenir (kısa dönemli hafıza). Eğer bu çözüm bir önceki adımdan gelen en iyi çözümden de iyiyse, en iyi çözüm olarak da seçilir (uzun dönemli hafıza).
4. Algoritma önceden belirlenen durdurma koşulu sağlanana kadar aşamalı işlemlerle sürdürülür.
5. Tabu listesinin hafızası dolduğunda ise, listede en uzun kalan hareket, listeden çıkarılır. Bazı durumlarda da, önceden belirlenen bir tabu yıkma kriterine göre tabu listesine giren hareketin gerçekleşmesine izin verilir. Şekil 3.2’de yasaklı arama algoritmasının akış şeması görülmektedir [87].

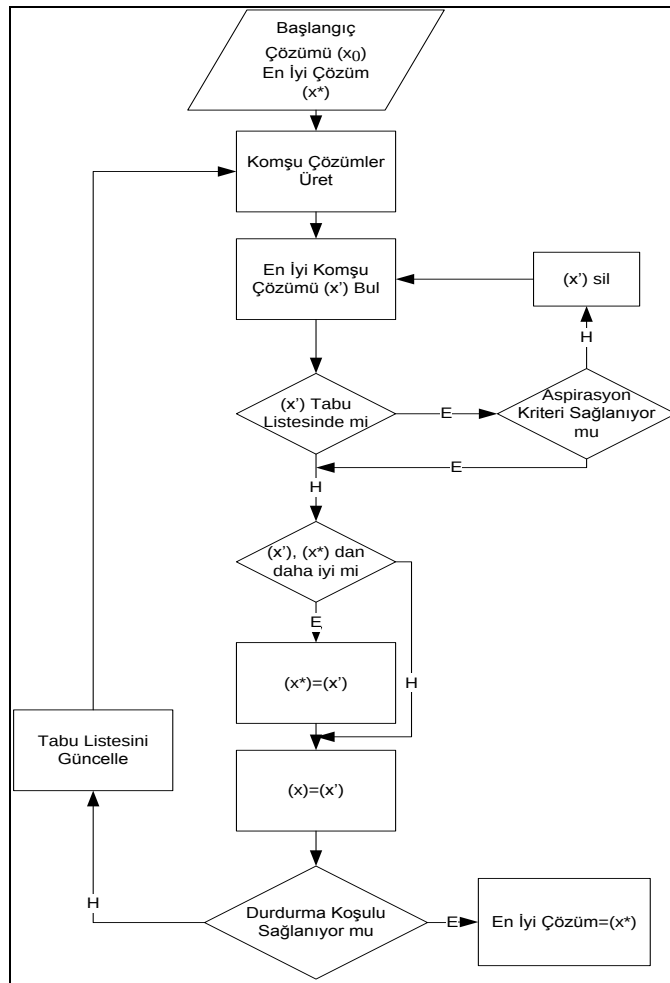
3.3. Genetik Algoritmalar

İlk olarak 1975 yılında Michigan Üniversitesi’nde John Holland tarafından ortaya konulan ve 1989 yılında Goldberg tarafından geliştirilen genetik algoritmalar, doğada aynı türe ait canlı topluluğunu oluşturan bireylerin, yaşam ortamı koşullarına uyumluluğuna göre seçimlerini temel almaktadır. Buna göre, ortam koşullarına uyumlu özellikleri taşıyan bireyler, hayatta kalarak özelliklerini bir sonraki nesillere aktarabilirken, taşımayan bireyler ölür ve bunların özellikleri nesiller geçtikçe gen havuzundan silinir.

Olasılık kurallarına göre çalışan genetik algoritmalar, yalnızca amaç fonksiyonuna gereksinim duyarlar. Çözüm uzayının tamamını değil, belirli bir kısmını tararlar. Böylece, etkin arama yaparak çok daha kısa sürede çözüme ulaşırlar. Diğer bir önemli üstünlükleri ise çözümlerden oluşan popülasyonu eş zamanlı incelemeleri ve

böylelikle yerel en iyi çözümlere takılmamalarıdır. Genetik algoritmalar daha çok, matematiksel modeli kurulamayan, çözüm alanı oldukça geniş, problemi etkileyen faktörlerin çok fazla olduğu problemlerin çözümünde etkin olarak kullanılmaktadır. Bu problemlerin başında endüstride karşılaşılan iş sıralama çizelgeleme problemleri gelmekte ve en çok kullanılan alanlardan birisini oluşturmaktadır [88].

Reeves; ulusal hükümetler ve organizasyonlar tarafından genetik algoritma tabanlı projelere, tavlama benzetimi ve yasaklı arama tabanlı projelere göre daha fazla kaynak ayrılmakta olduğunu belirtmektedir. Bu ifadeden anlaşılacağı üzere, genetik algoritmaların geleneksel optimizasyon yöntemlerine olduğu gibi sözü edilen yapay zeka yöntemlerine göre de çeşitli alanlarda üstünlükleri bulunmaktadır. Bu üstünlükler genetik algoritmanın arama yapısıyla ilgilidir [89].



Şekil 3.2: YAA akış şeması.

3.3.1. Genetik Algoritmalarla İlgili Temel Kavramlar

3.3.1.1.Gen ve Alel Gen

Gen, bir canlının karakteristik özelliklerini taşıyan en küçük temel kalıtım birimidir. Yavru bireyde karakteristik özelliği oluşturan genlerden biri anneden diğeri babadan gelir. Her biri farklı özelliği temsil eden bu gen çiftine “Alel Gen” adı verilir.

3.3.1.2. Kromozom

Kromozom, bir ya da daha fazla genin bir araya gelmesiyle oluşan ve üzerinde gerekli tüm bilgiyi taşıyan genetik yapıdır ve her biri problemin bir alternatif çözümünü göstermektedir.

Kromozomlar, genetik algoritma yaklaşımının üzerinde uygulandığı en temel birimler olduğundan bilgisayar ortamında iyi ifade edilmelidir. Örneğin, 100011101 genleri 4, 3, 5 birim uzunluklara sahip olan üç doğru parçasına ait uzunluk bilgisinin ikilik düzende kodlandığı bir kromozomu göstermektedir [90].

3.3.1.3. Popülasyon (Topluluk)

Kromozomlardan oluşan topluluğa popülasyon denir. Popülasyon, geçerli alternatif çözüm kümesidir. Popülasyonda farklı karakterlerin ortaya çıkması, genetik işlemler olarak adlandırılan, seçim, çaprazlama ve mutasyon süreçleriyle meydana gelir. Bu sayede topluluktaki gen havuzuna yeni karakterler eklenmiş olur.

3.3.1.4. Uygunluk Fonksiyonu

Yeni popülasyona taşınacak bireylerin belirlenmesinde kullanılan bir değerlendirme kriteridir. Kromozomun problemin çözümüne uygunluğunu gösteren başarı ölçüsü olarak da düşünülebilir. Verilen belirli bir kromozom için uygunluk fonksiyonu, o kromozomun temsil ettiği çözümün kullanımıyla veya yeteneğiyle orantılı olan sayısal bir uygunluk değeri verir. Bu bilgi, her kuşakta daha uygun çözümlerin seçiminde yol göstermektedir.

Uygunluk Fonksiyonu, problem için en uygun çözümü belirleme kriteri olduğundan üzerinde durulan konuyla ilgili kar veya verimliliği maksimum yapacak değişkenlerin ölçülmesini sağlayacak bir fonksiyon olmalıdır. Bu fonksiyonun belirlenmesi için problem iyi tahlil edilerek objektif bir değerlendirme kriteri seçilmelidir [88].

3.3.2. Genetik Algoritmelerde Kodlama

Bir problemin çözümüne başlarken kromozomların nasıl kodlanması gerektiğine karar verilmelidir. Kodlama işlemi, tamamen problemin yapısına bağlı olarak yapılır.

3.3.2.1. İkili Kodlama

Bu kodlama yönteminde her kromozom bit (0 veya 1) karakter dizilerinden oluşmaktadır. Örneğin; sırt çantası probleminde bu yöntem kullanılabilir. (Sırt çantası probleminde, her bir kromozomdaki bir bit değeri verilen nesnenin sırt çantasında olup olmadığını ifade eder.)

3.3.2.2. Permütasyon Kodlama

Çizelgeleme problemleri veya gezgin satıcı problemi gibi problemlerde kullanılabilir. Bu yöntemde, her kromozom her sırasında konum belirten karakter dizisinden oluşur. Örneğin; 5 farklı işi nasıl sıralayacağımızı bulmak istediğimizde Şekil 3.3.'deki gibi örnek kromozomlar elde ederiz.

Kromozom A	2 4 1 5 3
Kromozom B	1 3 4 5 2

Şekil 3.3: Permütasyon kodlama örneği.

3.3.2.3. Değer Kodlama

Değer kodlamada her kromozom bazı değerlere eşittir. Değerler problemle ilgili herhangi bir şey olabilir. Gerçek sayılar, karakterler veya herhangi nesneler.

3.3.2.4. Ağaç Kodlama

Genellikle evrimleşen program veya ifadeler için kullanılmaktadır. Her kromozom bazı nesnelerin ağacıdır.

3.3.3. Genetik Algoritmaların Parametreleri

Kontrol parametreleri genetik algoritmaların performansı üzerinde önemli etkiye sahiptir. Bunlar; popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı ve seçim metodu olarak sayılabilir.

3.3.3.1. Popölasyon Büyüklüğü

Literatürde, popölasyondaki birey sayısı (kromozom sayısı) ile ilgili genel bir kural yoktur ve kullanıcının kendisinin bir büyüklük ataması uygun görölmektedir. Bu deęer çok küçük olduęunda, genetik algoritma yerel bir optimuma takılabilmektedir. Popölasyonun çok büyük olması ise çözüme ulaşma zamanını artırmaktadır. Bu konuda Goldberg 1985'te, yalnızca kromozom uzunluęuna baęlı bir popölasyon büyüklüğü hesaplama yöntemi önermiştir [89].

3.3.3.2. Çaprazlama Olasılığı

Çaprazlama, mevcut iyi kromozomların özelliklerinin birleştirilerek daha iyi özellikte kromozomlar yaratmak için gerçekleştirilir. Ebeveyn olarak seçilen iki bireyin çaprazlanması bir olasılık içerisinde gerçekleşir. Algoritmaya önceden girilen çaprazlama olasılıęının, (0,1) aralıęından rastgele seçilen bir R sayısından küçük olması durumunda çaprazlama gerçekleşirken, bu koşulun sağlanmaması durumunda bir önceki nesildeki birey yeni nesle taşınır. Başka bir deyişle, çaprazlama olasılığı çaprazlamanın ne kadar sıklıkla yapılacaęını belirtir. Çaprazlama olasılığı %100 olduęunda tüm ebeveynler çaprazlanırken, %0 olduęunda tüm yavrular ebeveynlerinin kromozomlarının aynısına sahip olurlar.

3.3.3.3. Mutasyon Olasılığı

Nesiller ilerledikçe topluluğun gen havuzu duraęan hale gelmektedir. Bu durum toplulukta yeni karakteristik özelliklere sahip kromozomların ortaya çıkmasını engeller ve buna baęlı olarak algoritmanın yerel en iyi çözümlere takılması olasılığı doğmaktadır. Mutasyon operatörü modellerde küçük deęişimler meydana getirerek algoritmanın çözüm uzayında aramaya devam etmesini sağlamaktadır. Eęer, mutasyon olasılığı yüksek olursa genetik algoritma rastsal aramaya dönüşebilmektedir.

3.3.3.4. Seçim Metodu

Seçim metodu, genetik bilgi alışverişinin sağlandığı çaprazlama için ebeveynleri belirler. Seçilen bireyler karakterlerini sonraki nesillere aktarma şansını elde ederler. Genetik algoritmada deęişik seçim metotları vardır:

- i. Rulet Çemberi Seçimi: Her bir kromozom için uygunluk değeri hesaplanır ve her birinin uygunluk değeri toplam uygun değerine bölünür. Elde edilen 0-1 arasındaki değere göre bir rulet çemberine yerleştirilir. (Rulet çemberi üzerindeki kromozomun yerinin boyutu kromozomun uygunluğuyla orantılıdır.) Bir bilyenin bu rulet çemberi üzerine atıldığı düşünülürse daha uygun olan kromozomlar daha fazla seçilecektir. Bu süreçte, rulet çemberi üzerindeki kümülatif seçim ihtimalleri hesaplanır ve her seçimde bir rastsal sayı üretilir. Üretilen sayı hangi kromozomun kümülatif seçim ihtimalinden küçükse o kromozom seçilir. Rulet çemberi seçiminde uygunluk değerleri arasında büyük farklar olması durumunda seçimde bazı problemler çıkabilmektedir.
- ii. Sıralama Seçimi: Burada kromozomlar büyükten küçüğe sıralanır ve en kötü uygunluk değerini 1. sıradaki, en iyi uygunluk değerini de n. sıradaki kromozom alır. $P_n = \frac{N_{iyi-n+1}}{\sum_{n=1}^n N_{iyi}}$ formülüne göre her bir kromozomun sıra numarasını göz önüne alan bir normalize seçim olasılığı hesaplanır. Kromozom seçiminde kümülatif ihtimaller kullanılmaktadır. Üretilen rastsal sayı, hangi kromozomun kümülatif ihtimal değerinden küçükse o kromozom seçilir. Kromozomlar kendisi ile eşleştirilirse, bu kromozomun gelecek nesilde daha etkin olarak yer aldığı söylenebilir. Bu yaklaşım tarzı doğaya daha uygundur.
- iii. Sabit Durum Seçimi: Bu seçimin temel amacı, toplumun var olan kromozomlarının büyük bir kısmının yeni nesle aktarılmasıdır. Her yeni nesilde yüksek uygunluk değerine sahip kromozomlar yeni yavruları oluşturmak için seçilir ve düşük uygunluk değerine sahip kromozomlar kaldırılarak yerlerine yeni oluşturulan bu yavrular konur. Popülasyonun geri kalan kısmı aynen yeni nesle aktarılır.
- iv. Turnuva Metodu: Popülasyon içerisindeki iki kromozom rastsal olarak seçilerek uygunluk değeri yüksek olan çaprazlama havuzuna gönderilir. Çaprazlama için iki turnuva tutulur. Bunlardan biri her bir ebeveyni seçmek içindir. Bir ebeveyni birden fazla seçme olasılığı vardır. Turnuva seçiminin avantajı, popülasyonun daha kötü bireylerinin seçilmeyecek olması ve bundan dolayı sonraki neslin genetik yapısına katılmayacak olmasıdır.

- v. Seçkinlik: Çaprazlama ve mutasyon yöntemleriyle yeni bir nesil oluştururken, en iyi kromozomları kaybetme olasılığımız vardır. Seçkinlik, en iyi kromozomların ilk önce kopyalanıp yeni nesle aktarıldığı yöntemdir. Geri kalan kromozomlar diğer yöntemlerle üretilir. Seçkinlik genetik algoritmanın performansını hızlı bir şekilde artırabilir, çünkü bulunan en iyi çözümün kaybolmasını önlemektedir.

3.3.4. Genetik Algoritmada İşlem Adımları

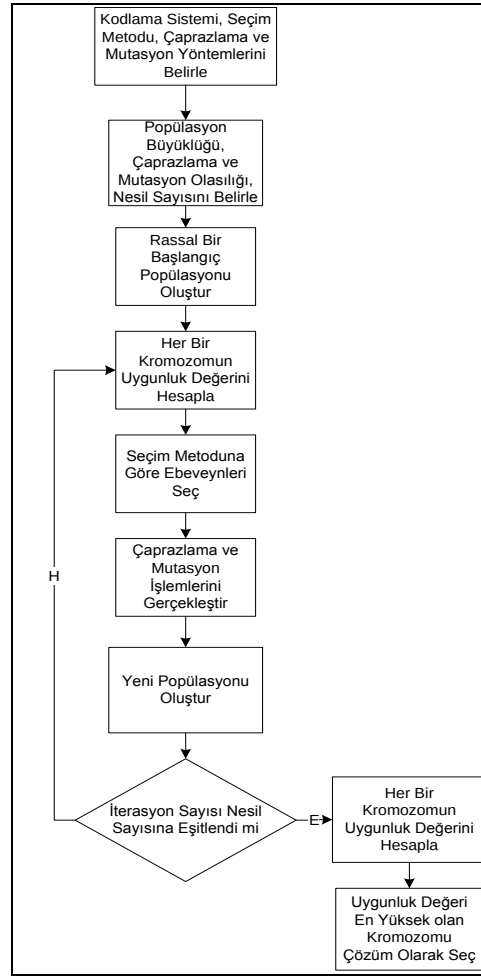
- i. Öncelikle probleme uygun olan kodlama sistemine, seçim metoduna, çaprazlama ve mutasyon yöntemlerine karar verilir. Popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, nesil sayısı gibi parametreler belirlenir.
- ii. Rastsal olarak kromozomlar belirlenerek bir başlangıç popülasyonu oluşturulur.
- iii. Her bir kromozomun uygunluk değeri hesaplanır.
- iv. Yeni popülasyonun oluşumu için öncelikle seçim metoduna göre ebeveynler seçilir ve çaprazlama ile mutasyon işlemleri yapılır. Sonuçta yeni bir popülasyon meydana gelir.
- v. Yeni popülasyon oluşturulduktan sonra üçüncü ve dördüncü adımları iterasyon sayısı nesil sayısına ulaşınca kadar devam ettirilir.
- vi. İterasyon sayısı nesil sayısına ulaştığında son popülasyondaki her bir kromozomun uygunluk değeri hesaplatılır ve en iyi olan kromozom çözüm olarak seçilir. Şekil 3.4'te genetik algoritmaların işleyişinin bir akış şeması görülmektedir.

3.4. Karınca Kolonisi Eniyileme (ACO) Algoritması

Kombinatoriyel eniyileme problemlerinin çözümü için Dorigo ve arkadaşları 90'lı yılların başında, karınca sistemi adı verdikleri bir algoritma geliştirmişlerdir. Karınca sistemi ilk olarak gezgin satıcı (TSP) problemine uygulanmış fakat, daha sonra geliştirilerek ve performansı artırılarak diğer eniyileme problemlerine de uygulanmıştır [92].

Karınca kolonisi eniyilemesi algoritması, gerçek karınca kolonilerinin yapmış olduğu doğal eniyileme işleminin yapay bir versiyonudur. Gerçek karıncalar, yuvaları ile

yiyecek kaynağı arasındaki en kısa yolu bulma kabiliyetine sahiptirler ve ayrıca çevredeki değişimlere de adapte olabilmektedirler. Diğer bir ilginç noktada karıncaların çok iyi görme kabiliyetlerinin olmamasıdır. Yani, en kısa yolu keşfetme uğraşında yönleri seçmek için etrafı tam olarak göremezler. Karıncalar üzerine yapılan çalışmalar, en kısa yolu bulma kabiliyetlerinin birbirleri arasındaki feromon kimyasalını salgılayarak haberleşmenin bir sonucu olduğunu göstermiştir [92].



Şekil 3.4: Genetik algoritmanın akış şeması.

Karıncı kolonisi eniyilemesi algoritması, karınca sisteminden dört ana noktada farklılık gösterir. İlki karınca sisteminden daha atılgan bir seçme kuralı uygulanmasıyla, karınca sistemindeki karıncaların birikimi olağan üstü bir başarıya dönüştürülmüştür. İkincisi feromon buharlaşımı ve feromon bırakımı, sadece o ana kadarki en iyi tura ait arklarda geçerlidir. Bir tur sonunda bütün karıncalar bir çözüm oluşturduklarında, feromon izi tekrarın başlangıcından o yana en iyi turu bulan karıncanın kullandığı arklara ilave edilir. Üçüncüsü, karıncaların i 'den j 'ye hareket ederken kullandıkları bir (i,j) arkındaki feromonun bir kısmı, her defasında, alternatif

yolların keşfini arttırmak için, azalır. Bu şekilde farklı bir lokal feromon güncelleme gerçekleştirilir. Dördüncüsü ziyaret edilecek bir sonraki yerin seçimini sınırlamak için aday listeleri kullanılır [93].

Karınca sayısı, q_0 (en iyi çözümün sonraki iterasyonlara aktarılması olasılığı), α (feromon miktarının önemi), β (yol uzunluklarının önemi) karınca kolonisi eniyilemesi algoritmalarının performansını etkileyen ayarlanabilir temel parametreleridir. Diğer karınca algoritmaları şöyledir:

- i. Maksimum-Minimum Karınca Sistemi
- ii. Rank Temelli Karınca Sistemi
- iii. Çoklu Karınca Koloni Algoritmaları
- iv. Melez Karınca Kolonisi Eniyileme Algoritması

Algoritmanın çalışma sürecinde temel işlemler, yapay karıncaların turları sonunda geçmiş oldukları yolların feromon miktarlarının arttırılması, belirli bir oranda feromon buharlaşmasının gerçekleştirilmesi, en iyi çözümün bulunması, buna bağlı olarak global feromon güncellemesinin yapılması ve karıncaların yenilenen bu feromon miktarlarına bağlı olarak yeni turlarını gerçekleştirmeleridir. Algoritmanın adımları şöyledir:

- i. Başlangıç feromon değerleri belirlenir.
- ii. Karıncalar her düğüme rastsal olarak yerleştirilir.
- iii. Her karınca, sonraki yeri denklemde verilen yerel arama olasılığına bağlı olarak seçmek suretiyle turunu tamamlar.
- iv. Her karınca tarafından kat edilen yolların uzunluğu hesaplanır ve yerel feromon güncellemesi yapılır.
- v. En iyi çözüm hesaplanır ve global feromon yenilemesinde kullanılır.
- vi. Maksimum iterasyon sayısı ya da yeterlilik kriteri sağlanana kadar ikinci adıma gidilir.

3.5. Yapay Bağışıklık Sistemi Algoritması

Canlılarda vücudun mikropları tanıyabilme ve ani olarak onları tahrip edebilmeleri doğal bağışıklık sistemi olarak bilinir. Yapay bağışıklık sistemleri de canlılardaki bu doğal bağışıklık sisteminin işleyişinin, fonksiyonlarının ve mekanizmalarının çeşitli optimizasyon problemleri, veri madenciliği, makine öğrenmesi, görüntü tanımlama,

bilgisayarlarda virüs tespiti, vb. alanlarda taklit edilerek uygulanması sonucu ortaya çıkmıştır.

Yapay bağışıklık sistemi, çözüm uzayının kodlanması, uygunluk fonksiyonunun belirlenmesi, mutasyon vb. ortak operatörler ve süreçleri kullanarak genetik algoritmalarla, hafıza ve öğrenme özellikleriyle de yapay sinir ağlarıyla benzerlik göstermektedir. Yapay bağışıklık sistemi, yerel optimumlara daha hızlı yönelmekte ancak bir genetik algoritma kadar global optimuma yakınsamayabilmektedir.

Yapay bağışıklık sisteminin aşamaları şöyledir:

- i. Hafıza hücrelerinin bir alt kümesi (M) ile bir önceki iterasyondan kalan kümeyi (Pr) birleştirerek aday çözümler kümesi olan P'yi oluştur ($P = Pr + M$).
- ii. Oluşturulan popülasyondan benzerlik ölçümüne bağlı olarak n tane en iyi bireyi seç (Pn).
- iii. Bu n tane en iyi bireyi kopyalayarak C geçici popülasyonunu oluştur. Bu geçici popülasyondaki klonların sahip oldukları alanlar, antijene benzerlikleri ile doğru orantılıdır.
- iv. C içerisindeki klonların her birini bir mutasyon oranına ve klonun benzerlik değerine bağlı olarak mutasyona tabi tut. Bu sayede olgunlaşmış antikör popülasyonunu (C*) oluştur.
- v. Hafıza kümesini, gelişmiş hücrelerden C* dan tekrar seçerek oluştur (P kümesinin bazı hücreleri C*'ın bazı gelişmiş hücreleri ile yer değiştirebilir).
- vi. Farklılaşmayı sağlamak için popülasyonun düşük benzerlikli antikörlerini (d tane), yenileri ile değiştir [94].

Yukarıda verilen adımlardaki terimleri bir çizelgeleme problemi ile Çizelge 3.2' de ilişkilendirilmiştir [95].

3.6. Parçacık Sürü Eniyileme Algoritması

NP-zor yapıdaki problemler için sezgisel ve meta-sezgisel bir çok çözüm yaklaşımı söz konusudur. Zaman içerisinde, farklı çözüm noktaları üzerinde dolaşan fakat karmaşık matematiksel hesaplara gerek duymayan bir çözüm yöntemi ihtiyacı ortaya çıkmıştır.

Çizelgeleme Probleminde Kavramlar	Yapay Bağışıklık Sisteminde Karşılıkları
Çizelgelenecek İşler	Antijenler
Problem İçin Yeni Üretilen Çizelgeler	Antikor
Yeni Üretilen Çizelgelerdeki Tüm İşlerin Tamamlanma Zamanı (Makespan)	Antikorum Uyum Değeri
Çözümlerden Oluşan Küme (Popülasyon)	Hücre Repertuarı
İyi Çizelge Veren Çözümler	Klonlar
En Kötü Çizelgeyi Veren Çözümler	Kötü Antikorlar
Komşuluk Mekanizması İle Diğer Bir Çizelgeye Giriş (Mutasyon)	Somatik Hipermutasyon
İyi Çizelgelerin Mutasyonla Geliştirilmesi	Uyum Değeri Olgunlaşması
Kötü Çizelgelerin Popülasyonla Silinmesi Ve Yerlerine Yenilerin Oluşturulması	Reseptör Denetimi
En İyi Çizelgelerin Hafızada Saklanması (Elitizm)	Hafıza Hücreleri
Popülasyondan Uyum Değerine Göre Herhangi Bir Bireyin Seçimi	Antijen Antikor Eşleşmesi

Çizelge 3.2: Yapay bağışıklık sistemi algoritması terimleri.

Bu ihtiyacın fark edilmesiyle 1995 yılında kuş ve balık sürülerinin iki boyutlu davranışlarından esinlenen Kennedy ve Eberhart [96] parçacık sürü eniyileme algoritmasını geliştirmişlerdir. Algoritma, diğer eniyileme metotlarıyla karşılaştırıldığında uygulanmasının daha kolay olduğu ve ayarlanması gereken çok az parametresinin olduğu görülür. Parçacık sayısı, parçacık boyutu, parçacık aralığı, her bir iterasyonda bir parçacıkta meydana gelebilecek en büyük değişiklik (V_{max}), öğrenme faktörleri ve durdurma koşulu parçacık sürü eniyileme algoritmasının parametreleridir.

Parçacık sürü eniyileme algoritması, bir grup rastgele çözüm kümesi (parçacık sürüsü) ile başlar ve güncellemelerle en iyi çözüm bulunmaya çalışılır. Algoritmanın adımları şöyledir [97]:

- i. Parçacıklar rastgele üretilen başlangıç pozisyonları ve hızları ile birlikte oluşturulur.
- ii. Popülasyon içerisindeki tüm bireylerin uygunluk değerleri hesaplanır.
- iii. Her jenerasyondaki bütün bireyler bir önceki jenerasyonda bulunan en iyi (p_{best}) ile karşılaştırılır. Eğer daha iyi birey varsa yer değiştirilir.
- iv. Jenerasyondaki en iyi değer global en iyi değerden daha iyiye yer değiştirilir.
- v. $v_i^{k+1} = w * v_i^k + c_1 * rand_1^k * (p_{best}^k - x_i^k) + c_2 * rand_2^k * (g_{best}^k - x_i^k)$
 $x_i^{k+1} = x_i^k + v_i^{k+1}$ Formüllerine göre pozisyon ve hızlar yenilenir. Burada x_i pozisyon ve v_i hız değerlerini verirken, k iterasyon sayısını verir. $rand_1$ ve $rand_2$ değerleri, 0 ile 1 arasında rastgele üretilmiş sayılardır. w atalet ağırlık değeridir. c_1 ve c_2 değerleri sabittir ve genelde 2'ye yakın bir değer alınır.
- vi. Durdurma kriteri sağlanıncaya kadar ii. ve iv. arasındaki adımlar tekrar edilir. Şekil 3.5'de algoritmanın akış şeması görülmektedir.

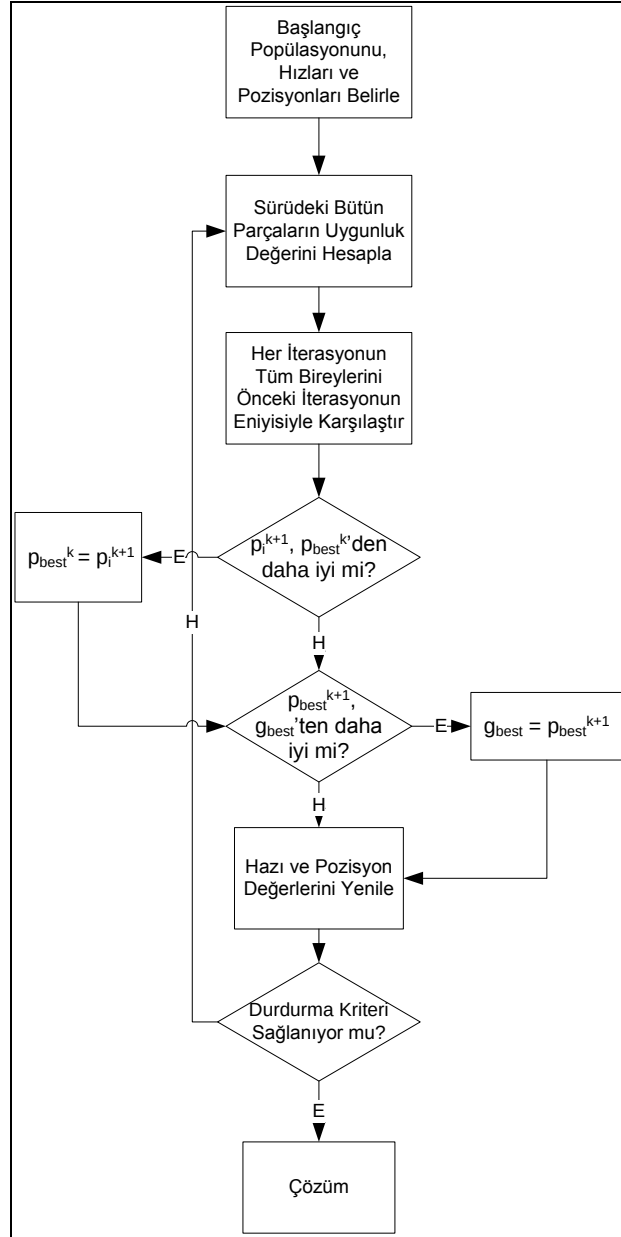
3.4. Yapay Sinir Ağları

Yapay sinir ağları, insan beyninin çalışma sisteminin yapay olarak benzetimi çabalarının bir sonucu olarak ortaya çıkmıştır. En genel anlamda, insan beynindeki birçok nöronun (sinir hücresinin), ya da yapay olarak basit işlemcilerin birbirlerine değişik etki seviyeleri ile bağlanması sonucu oluşan karmaşık bir ağ sistemi olarak düşünülebilir. Bu ağlar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler.

Biyolojik sistemlerde öğrenme, sinir hücreleri arası sinaptik bağlantıların ayarlanmasıyla olur. Yani, insanlar doğumdan itibaren yaşayıp tecrübe ettikçe sinaptik bağlantılar ayarlanır, hatta yeni bağlantılar oluşur.

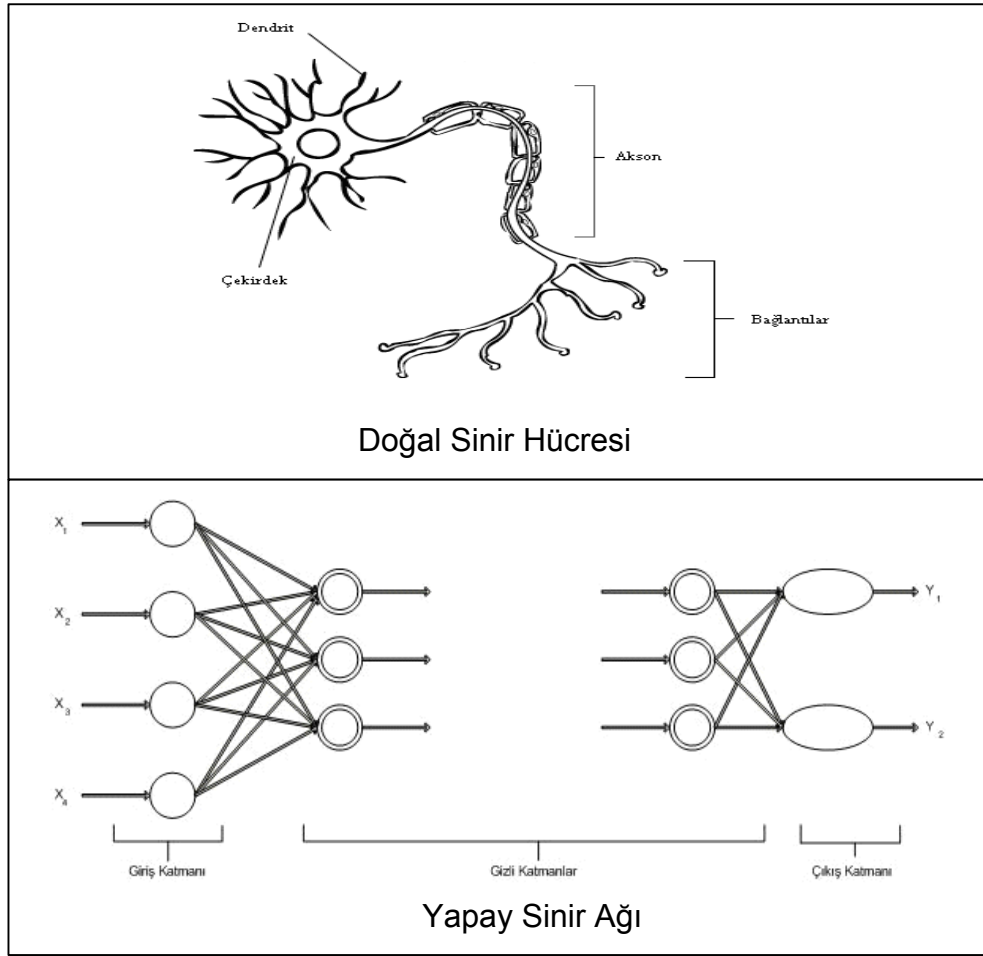
Aynı durum yapay sinir ağları için de geçerlidir. Öğrenme girdi/çıktı verilerinin işlenmesiyle, yani eğitme algoritmasının bu verileri kullanarak bağlantı ağırlıklarını bir yakınsama sağlanana kadar, tekrar tekrar ayarlamasıyla olur. Sinir hücresi, girdi ve çıktı yolunu sağlayan bağlantılar ve bu bağlantıların sağlamlığını gösteren

bağlantı ağırlığı bir yapay sinir ağının ana temel işlem elemanlarıdır. Şekil 3.6’da doğal sinir hücresi ve yapay sinir ağı gösterilmiştir.



Şekil 3.5: Parçacık Sürü Eniyileme algoritması akış şeması.

Yapay sinir ağlarının veriler arasındaki ilişkiyi ortaya çıkarabilme gücünü, paralel dağılmış yapısından, öğrenebilme ve genelleme yeteneğinden aldığı söylenebilir. Yapay sinir ağlarının böyle üstün özelliklere sahip olması, karmaşık problemleri çözebilmesini sağlamaktadır.



Şekil 3.6: Doğal ve yapay sinir hücresi.

3.7.1. Yapay Sinir Ağlarının Özellikleri

Yapay sinir ağları, günümüzde birçok bilim alanında aşağıdaki özellikleri nedeniyle etkin olmuş ve uygulama yeri bulmuştur [98].

3.7.1.1. Doğrusal Olmama

Yapay sinir ağlarının temel işlem elemanı olan hücre, doğrusal değildir. Dolayısıyla hücrelerin birleşmesinden meydana gelen yapay sinir ağları da doğrusal değildir ve bu özellik bütün ağa yayılmış durumdadır. Bu özelliği ile yapay sinir ağları, doğrusal olmayan karmaşık problemlerin çözümünde en önemli araç olmuştur.

3.7.1.2. Öğrenme

Yapay sinir ağlarının arzu edilen davranışı gösterebilmesi için amaca uygun olarak ayarlanması gerekir. Bu, hücreler arasında doğru bağlantıların yapılması ve bağlantıların uygun ağırlıklara sahip olması gerektiğini ifade eder. Yapay sinir ağlarının karmaşık yapısı nedeniyle bağlantılar ve ağırlıklar önceden ayarlı olarak verilemez ya da tasarlanamaz. Bu nedenle yapay sinir ağları, istenen davranışı gösterecek şekilde ilgilendiği problemde aldığı eğitim örneklerini kullanarak problemi öğrenmelidir.

3.7.1.3. Genelleme

Yapay sinir ağları, ilgilendiği problemi öğrendikten sonra eğitim sırasında karşılaşmadığı test örnekleri için de arzu edilen tepkiyi üretebilir. Örneğin, karakter tanıma amacıyla eğitilmiş bir yapay sinir ağı, bozuk karakter girişlerinde de doğru karakterleri verebilir.

3.7.1.4. Uyarlanabilirlik

Yapay sinir ağları, ilgilendiği problemdeki değişikliklere göre ağırlıklarını ayarlar. Yani, belirli bir problemi çözmek amacıyla eğitilen bir yapay sinir ağı, problemdeki değişimlere göre tekrar eğitilebilir ve değişimler devamlı ise gerçek zamanda da eğitime devam edilebilir.

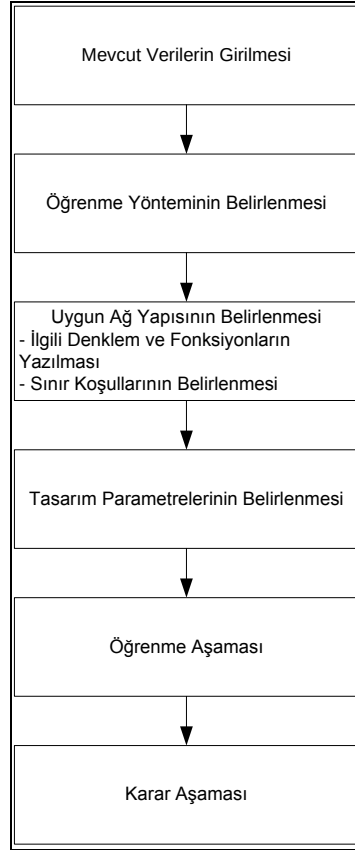
3.7.1.5. Hata Toleransı

Yapay sinir ağları, çok sayıda hücrenin çeşitli şekillerde bağlanmasından oluştuğu için paralel dağılmış bir yapıya sahiptir ve ağına sahip olduğu bilgi, ağıdaki bütün bağlantılar üzerine dağılmış durumdadır. Bu nedenle, eğitilmiş bir Yapay sinir ağının bazı bağlantılarının hatta bazı hücrelerinin etkisiz hale gelmesi, ağına doğru bilgi üretmesini önemli ölçüde etkilemez. Bu nedenle, geleneksel yöntemlere göre hatayı tolere etme yetenekleri son derece yüksektir.

3.7.2. Genel Bir Yapay Sinir Ağı Modeli

Yapay sinir hücresinde, öncelik ve önemine (ağırlık faktörü) bağlı olarak hücreye gelen veriler (uyarılar) hücre içi denge durumu veya sınır değeri dikkate alınarak doğrusal olmayan bir aktivasyon fonksiyonu yardımıyla çıktı şeklinde kararlara dönüştürülür.

Yapay sinir hücrelerinin genellikle katmanlı bir yapı gösterecek biçimde birbirine bağlanmasıyla yapay sinir ağları meydana getirilir ve yapay sinir hücreleri arasında kurulan bağlantılara ait ağırlık faktörlerinin, verilerle uyumlu olacak şekilde değiştirilmesi benzer şekilde biyolojik “öğrenme” sürecine karşılık gelir. Yapay sinir ağları kullanımındaki aşamalar bir akış şeması ile gösterilebilir. (bkz. Şekil 3.7)



Şekil 3.7: Yapay sinir ağları akış şeması.

4. TEK MAKİNELİ SIRA BAĞIMLI HAZIRLIK ZAMANLI ÇİZELGELEME PROBLEMİ

Tek makineli çizelgeleme problemi, çizelgeleme problemleri içinde en basit yapıya sahip olanıdır. Ancak, bu modelleri önemli kılan çeşitli sebepler vardır. Tek makineli ortam diğer ortamların en yalın ve en özel durumudur. Tek makineli modeller, genellikle paralel veya sıralı makinelerin özelliklerini sergiler. Sonuç olarak, tek makineli modeller sadece içinde bulundukları tek makineli ortam için değil, daha karmaşık makineli ortamlarda da sezgisellerin elde edilmesini sağlamaktadır. Uygulamada çizelgeleme problemlerinde, daha karmaşık makineli ortamlar daha alt problemlere ayrılarak tek makineler ele alınır. Örneğin; tek bir darboğazı olan karmaşık makineli bir ortam, tek makineli bir model kurulmasına olanak vermektedir [99].

Gerçek hayatta, farklı özellikler taşıyan işlerin hazırlık zamanlarının işlerin sıralamasına göre değişmesi, siparişlerin tam zamanında karşılanma zorunluluğu, üretim sistemine ait teknik kısıtların varlığı ve problemin çok amaçlı doğası gibi unsurlar dahil edildiğinde tek makineli çizelgeleme probleminin çözümü oldukça zorlaşmaktadır. Öte yandan, problemin klasik hali literatürde çok sayıda yer almasına rağmen, sıra bağımlı hazırlık sürelerini ve birden fazla amacı göz önünde bulunduran az sayıda çalışmaya erişilmiştir.

Son zamanlarda hazırlık zamanlarını göz önüne alan çizelgeleme problemleri araştırmacıların ilgisini çekmektedir. Çünkü işletmeler, tek bir kaynak üzerinde birçok farklı ürün üretmekte ve bu da hazırlık aktivitelerinin yapılmasını gerektirmektedir. Hazırlık aktiviteleri genellikle üretim prosesleri içerisinde maliyetleri artırıcı bir rol oynamaktadır. Bu yüzden planlamacılar çizelgelerinde hazırlık aktivitelerini de göz önünde bulundurmalıdırlar [3].

Bu çalışmada incelenen problemin özellikleri aşağıdaki gibidir:

- i. N tane birbirinden bağımsız iş vardır ve bu işler 0 anında hazırdır.

- ii. İşler arası hazırlık zamanları işlerin sıralamasına bağımlıdır. (Hazırlık zamanları işlem süreleri üzerine eklenemez.)
- iii. İş tanımları önceden bilinmektedir.
- iv. Makine sürekli çalışır durumdadır veya boş kalmamaktadır. (İşler arası gerçekleştirilen hazırlık faaliyetleri makinenin boş olduğunu ifade etmez. Bu sürelerde de işlem görüyor kabul edilir.)
- v. Bir işlem başladıktan sonra yarıda kesilemez.

Yukarıda sayılan özelliklerin göz önüne alındığı durumda, farklı teslim tarihli işlerin tam zamanında yetiştirilebilmesi için ağırlıklandırılmış (cezalandırılmış) erken bitirme ve geç bitirmenin eniyilenmesi problemi üzerinde durulmuştur. (Buradaki ağırlıklar, her bir iş için de farklılık göstermektedir.)

Çalışmanın 1. bölümünde de anlatıldığı üzere, problemi $n/1/S_{ij}, d_i/F_{\min}(\sum \alpha_i \cdot E_i + \sum \beta_i \cdot T_i)$ şeklinde gösterebiliriz. Bu problem 1992 yılında B. Jay Coleman [100] tarafından ele alınmış ve verilerin özel bir durumu için 1974 yılında Kenneth R. Baker tarafından önerilmiş olan bir matematiksel çözümden söz edilmiştir.

4.1. Problemin Matematiksel Çözümü

4.1.1. Literatürde Mevcut Olan Çözüm

$$\text{MIN } \sum_{i=1}^N (e_i \text{Early}_i + t_i \text{Tardy}_i) \quad (4.1)$$

St

$$X_i \geq p_i \quad i=1, \dots, N \quad (4.2)$$

$$X_i - \text{Tardy}_i + \text{Early}_i = d_i \quad i=1, \dots, N \quad (4.3)$$

$$X_j - X_i + M(1 - Y_{ij}) \geq p_j + s_{ij} \quad i=1, \dots, N, \quad j=i+1, \dots, N \quad (4.4)$$

$$X_i - X_j + M(Y_{ij}) \geq p_i + s_{ji} \quad i=1, \dots, N, \quad j=i+1, \dots, N \quad (4.5)$$

$$X_i, \text{Early}_i, \text{Tardy}_i \geq 0 \quad i=1, \dots, N \quad (4.6)$$

$$Y_{ij} \in \{0,1\} \quad i=1, \dots, N, \quad j=i+1, \dots, N \quad (4.7)$$

X_i : i işinin tamamlanma zamanı

Early_i : i işinin erken bitirilme zamanı

Tardy_i : i işinin geç bitirilme zamanı

Y_{ij} : i işi j işinden önceyse 1 değerini alır, değilse 0 değerini alır

N: Çizelgelenen iş sayısı

p_i : i işinin işlem süresi

d_i : i işinin teslim zamanı

e_i : i işinin erken bitirme cezası

t_i : i işini geç bitirme cezası

s_{ij} : i işiyle onu takip eden j işi arasındaki hazırlık zamanı

(4.2) numaralı kısıt, bir işin tamamlanma zamanının hazırlık zamanıyla birlikte işin işlem zamanında büyük olacağını göstermektedir. (4.3) numaralı kısıt, işin teslim tarihinin, işin gecikmesi durumunda tamamlanma zamanıyla gecikme zamanı farkına, işin erken bitmesi durumunda ise tamamlanma zamanıyla erken bitirme zamanının toplamına eşit olduğunu ifade etmektedir. (4.4) ve (4.5) numaralı kısıt aynı anda iki işin yapılmayacağını, (4.6) numaralı kısıt tamamlanma zamanı, erkenlik ve geçliğin sıfırdan büyük veya eşit olacağını, (4.7) numaralı kısıt da Y_{ij} karar değişkeninin 0 veya 1 değerini aldığını ifade etmektedir.

Yalnız, bu algoritmanın en iyi çözümü bulabilmesi için hazırlık zamanlarını içeren matris üçgensel olmalı yani, $s_{ij} + s_{jk} \geq s_{ik}$ koşulunu sağlamalıdır. Bu koşula göre, i işinden sonra j yapıldığı durumdaki hazırlık zamanı ile j işinden sonra k işinin yapıldığı durumdaki hazırlık zamanının toplamının, i işinden sonra k işi yapıldığı durumdaki hazırlık zamanından büyük veya eşit olması gerekmektedir. Aksi takdirde algoritma problem için olurlu bir çözüm bulamamaktadır.

Örnek 4.1: Ghaith Rabadi'nin [101] 1999 yılında yazmış olduğu doktora tezinde kullandığı aşağıdaki Çizelge 4.1'deki hazırlık zamanları matrisi ve işlem zamanlarıyla, bitiş zamanı tüm işler için eşit ve $d=600$, erkenlik ve geçlik cezaları da tüm işler için eşit ve $w=1$ olduğu durumda yukarıdaki metodu kullanarak erkenlik ve geçlik cezalarını en küçükleyecek sıralamayı bulalım.

Çizelge 4.1: Hazırlık zamanları matrisi.

S_{ij}	1	2	3	4	5	6
1	-	59	59	34	50	36
2	38	-	74	70	38	73
3	44	45	-	75	61	52
4	66	39	58	-	29	29
5	58	40	69	62	-	52
6	39	63	29	73	70	-
P_i	11	25	87	41	67	45

Problem LINDO 6.1’de modellenmiş, program 3 sn de en iyi (optimal) çözümü bulmuştur. Bu çözüme amaç fonksiyonunun değeri 629 ve sıralama değeri 3-5-2-1-4-6’dır.

Coleman, makalesinde [100] 8 işe kadar olan büyüklükteki problemi ele almış, daha büyük boyutlu problemlerin çözümlerinin etkin ve verimli bir şekilde elde edilebilmesi için sezgisel metodlar geliştirilmesinin uygun olacağından bahsetmiştir.

Bu matematiksel metodun küçük boyutlu problemler içerisindeki en zayıf tarafı, üçgensel olmayan hazırlık süreleri matrislerinin olduğu durumlarda olurlu çözüm üretememesidir. Bu yüzden model gezgin satıcı problemlerinin dışındaki her çizelgeleme problemine yanıt veremeyecektir. Metodun bu zayıf yönünden yola çıkılarak, küçük ölçekli problemler için tam kombinatoryel arama yapan şöyle bir metod önerilebilir:

4.1.2. Önerilen Çözüm

$$\text{MIN } \sum_{k=1}^N (\alpha_k E_k + \beta_k T_k) \quad (4.8)$$

St

$$\sum_{i=1}^{N!} A_i = 1 \quad (4.9)$$

$$C_k = \sum_{i=1}^{N!} A_i \cdot \left(\sum_{t=1}^{j_{[i,t]}=k} P_{j_{[i,t]}} + \sum_{t=2}^{j_{[i,(t-1)]}=k} S_{j_{[i,(t-1)]}, j_{[i,t]}} \right) \quad k=1,2,\dots,N \quad (4.10)$$

$$T_k \geq C_k - d_k \quad (4.11)$$

$$E_k \geq d_k - C_k \quad (4.12)$$

$$T_k, E_k \geq 0 \quad (4.13)$$

$$A_i \in \{0,1\} \quad (4.14)$$

E_k : k işinin erken bitirilme süresi

T_k : k işinin geç bitirilme süresi

α_k : k işinin erken bitirilme cezası

β_k : k işinin geç bitirilme cezası

A_i : i'nci sıralamanın karar değeri, 0 veya 1 değerini alır

C_k : k işinin tamamlanma zamanı

d_k : k işinin bitiş zamanı

$P_{j[i,t]}$: i'nci sıralamanın t'nci sıradaki işi olan $j[i,t]$ işinin işlem zamanı

$S_{j[i,(t-1)],j[i,t]}$: i'nci sıralamanın (t-1) sırasındaki işi olan $j[i,(t-1)]$ işinden, i'nci sıralamanın t sırasındaki işi olan $j[i,t]$ işine geçiş arasındaki hazırlık süresi

N : Çizelgelenen iş sayısı

(4.9) numaralı kısıt sadece bir sıralamanın atanmasını sağlamaktadır. (4.10) numaralı kısıt, gerçekleşen sıralamaya göre her bir işin tamamlanma zamanının, işin kendinin ve kendisinden önceki işlerin işlem süreleriyle, işin yapılacağı zamana kadarki gerçekleşen tüm hazırlık sürelerinin toplamına eşit olduğunu ifade etmektedir. (4.11) ve (4.12) numaralı kısıtlar işin erken bitirilme ve geç bitirilme durumlarını göstermekte, (4.13) numaralı kısıt ise iş erken bitirildiğinde, geç bitirilme süresinin ya da iş geç bitirildiğinde erken bitirilme süresinin negatif olmamasını sağlamaktadır. (4.14) numaralı kısıt ise A değişkeninin 0 veya 1 değerlerini alabilen bir karar değişkeni olduğunu ifade etmektedir. (4.8)'de ifade edilen amaç fonksiyonuna göre her bir işin erken bitirilme ve geç bitirilme cezaları farklıdır, bu cezaların enküçüklenmesi gerekmektedir.

Örnek 4.2: Yukarıdaki çözüm metodunu $N=4$ iş için, teslim tarihi ortak $d=100$, ağırlıkları eşit ve $w=1$ olan, hazırlık zamanları ve işlem süreleri Çizelge 4.2'de verilmiş olan problemin erkenlik ve geçlik cezalarını enküçükleyecek çözümü bulalım.

Çizelge 4.2: Hazırlık zamanları matrisi.

S_{ij}	1	2	3	4
1	-	10	10	10
2	15	-	15	10
3	5	15	-	10
4	15	10	5	-
P_i	20	30	10	20

Bu tablonun değerlerine bakıldığında üçgensellik özelliği taşımadığı görülür. Şöyle ki; $S_{43} + S_{31}$ değeri 10 olup S_{41} değeri olan 15'ten küçüktür. Dolayısıyla, literatürde mevcut olan çözüm algoritmasıyla olurlu çözümü elde edilemez.

Problem yukarıda önerilen metod ile LINDO 6.1'de modellenmiş, program 1 sn içerisinde en iyi (optimal) çözümü bulmuştur. Bu çözüme göre amaç fonksiyonunun değeri 135 ve sıralama değeri 2-4-3-1'dir.

Görüldüğü gibi önerilen metod, veri setleriyle ilgili hiçbir koşul olmaksızın küçük boyutlu problemlerin optimum çözümünü bulabilmektedir. Metodun zayıf yönü ise, karmaşık olması ve bu yüzden model kodlarını yazmanın zor olmasıdır. Örneğin; LINDO 6.1 çözücüsünde tüm kısıtları yazmak oldukça zaman almıştır. N adet işin sıralandığı bir problem için N! adet sıra olacak ve her bir sıranın gerçekleşme ihtimalini göz önüne alarak N adet işin ayrı ayrı tamamlanma zamanlarını yazmak gerekecektir. N adet işin sıralandığı bir problemde $N! + 3N$ adet değişken olacaktır. Yani, bu çözüm metodu sadece küçük boyutlu problemler için kullanışlıdır. Büyük boyutlu problemlerin çözümü için meta-sezgisel algoritmalar kullanılabilir.

4.2. Problemin Meta-Sezgisel Çözümü

Deterministik yöntemler bazı problemlerin çözümü için mümkün olmayabilmekte veya çok zaman ve maliyet gerektirebilmektedir. Bu tür problemlerde deterministik olmayan yöntemlerin kullanımı başarılı olabilir. Deterministik olmayan yaklaşımlar, aynı başlangıç durumu için farklı sonuçlar verebilmekte, ancak en iyi olmasa da

kabul edilebilir bir sonuca belirli bir süre sonunda ulaşabilmektedirler. Meta-sezgisel yöntemler deterministik olmayan yöntemlerdir ve en iyi çözümü garanti etmemekle birlikte, denenmesi gereken olasılıkların çok fazla olduğu durumlarda daha az deneme ile kabul edilebilir seviyede iyi bir çözüm bulmaktadırlar [102].

Giriş bölümünde de bahsedildiği gibi çalışmada ele alacağımız tek makineli hazırlık zamanları sıra bağımlı uygulama problemi, $n/1/S_{ij}, d_i/F_{\min}(\sum \alpha_i.E_i + \sum \beta_i.T_i)$, NP- zor karmaşıklık sınıfında bir problemdir. Bu problemin deterministik yöntemlerle çözümü mümkün değildir. Şöyle ki; $n=15$ büyüklüğünde bir problemin hazırlık matrisinin üçgensel olduğunu düşünürsek, Coleman'a [101] göre 240 tane kısıt, 150 tane de değişken olacaktır. Problemin hazırlık matrisinin üçgensel olmadığı durumda ise; $15!$ yani $1,3.10^{12}$ adet büyüklüğünde bir çözüm uzayında tam kombinatoryel olarak çözüm aranacaktır. Bu yüzden çalışmada ele alacağımız uygulama probleminin çözümü için iki farklı meta-sezgisel yöntemden faydalanacağız.

Uygulama probleminin çözümü, 3. bölümde kısaca anlatılan çizelgeleme problemlerinde kullanılan meta-sezgisel yöntemlerden, yasaklı arama algoritması ve genetik algoritmayla yapılacaktır. Problemin çözümüne ilk olarak yasaklı arama algoritmasıyla başlanacak, bu algoritmanın bulduğu çözüm, genetik algoritmanın başlangıç çözüm kümesi içerisine alınarak problem bir de genetik algoritmayla çözülecektir. Başlangıç çözümü meta-sezgiseller için önemli bir başarı kriteri olduğundan, genetik algoritmayla çözüm uzayını aramadan önce iyi bir başlangıç çözümü elde etmek için yasaklı arama algoritmasından faydalanacağız.

4.2.1. Problemin Çözümünde Kullanılan Yasaklı Arama Algoritması

Problemin başlangıç çözümünü elde etmek için kullanılan YAA'nın akış şeması şekil 4.1'de verilmiştir.

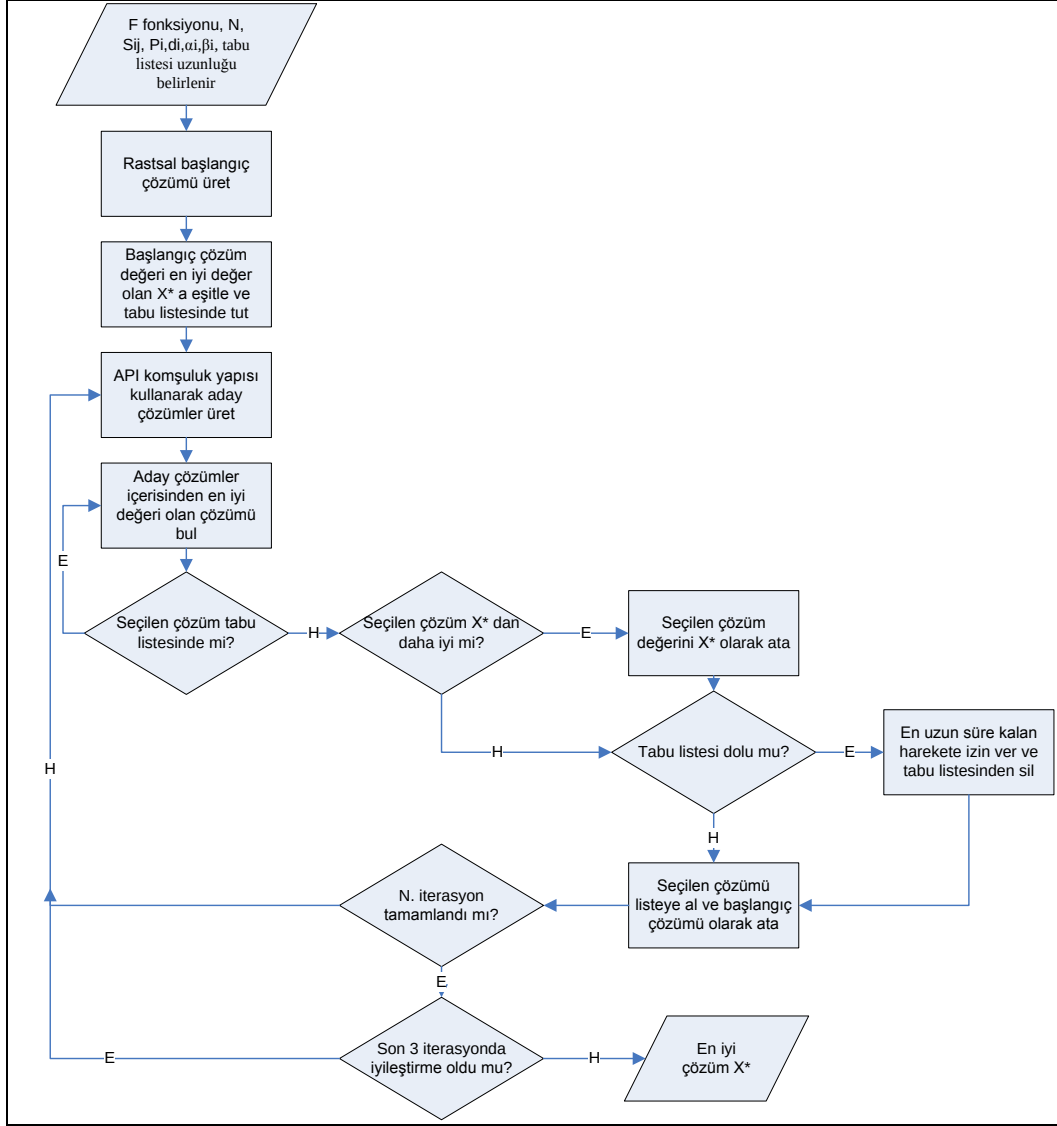
4.2.1.1. Başlangıç Aşaması

- i. Başlangıçta F fonksiyonu, $F_{\min}(\sum_{i=1}^N \alpha_i E_i + \beta_i T_i)$ ve N, S_{ij} matrisi, $P_i, d_i, \alpha_i, \beta_i$, tabu listesi uzunluğu gibi parametreler belirlenir.
- ii. Rastsal bir başlangıç sıralaması oluşturulur.
- iii. Başlangıç sıralamasını F değeri (çözüm değeri) en iyi çözüm olan X^* değerine eşitlenir ve sıralama en iyi çözüm olarak hafızada tutulur.

- iv. Tabu listesi başlangıçta boştur. Tabu listesi uzunluğu kullanıcıya bağlı bir parametredir. Çalışmanın örnek olay kısmında kullanılan algoritmada $2\sqrt{N}$ olarak belirlenmiştir. Bunun sebebi, yasaklı arama algoritmasının iyi bir başlangıç çözümü bulmak için kullanılmasıdır. Bu yüzden, mümkün olduğu kadar kısa süreli ve az iterasyonlu çalışması istenmektedir.

4.2.1.2. İteratif Aşamalar

- i. API komşuluk yapısı kullanılarak başlangıç çözümünden (N-1) adet komşu çözüm üretilir. (API komşuluk yapısı başlangıç sırasında ikili komşu sıraların, sırayla yer değiştirilerek elde edildiği bir yöntemdir.)
- ii. Bu komşu sıraların F değerleri hesaplatılır ve çözüm değeri en iyi, yani $F_{\min}$ olan sıralama için tabu listesine bakılır. Eğer bu çözüm tabu listesindeyse, komşu çözümler içerisinde ikinci en küçük çözüm değerine bakılır. Bu işlem tabu listesinde olmayan en iyi sıra bulunana kadar devam ettirilir ve bulunduktan sonra F değeri X (geçici çözüm değeri) olarak atanır.
- iii. X^* değeri ile X' değeri karşılaştırılır, $X' \leq X^*$ ise $X^* = X'$ değeri atanır ve sıralama en iyi çözüm olarak hafızaya alınır. $X' \leq X^*$ kuralı gerçekleşmiyorsa, herhangi bir değişiklik yapılmaz.
- iv. X' değerinin ait olduğu sıralama tabu listesine eklenir. Tabu listesinin hafızası doluyorsa listede en uzun süre kalan hareket listeden çıkarılır, yerine yeni çözüm eklenir. Çalışmada kullanılan algoritmada aspirasyon kriteri kullanılmamıştır.
- v. İterasyon durdurma koşulu, M adet iterasyonun koşulsuz olarak yapılması ve M'inci iterasyondan sonra X^* değerinde üst üste 3 iterasyonda bir iyileşme olmaması olarak tanımlanmıştır. Durdurma koşulu sağlanıyorsa, çözüm en iyi çözüm olarak hafızada tutulan sıralama ve çözüm değeri olarak X^* ekranda gösterilir. Durdurma koşulu sağlanmıyorsa, çözüm değeri X' olan sıralama başlangıç çözümü olarak atanır ve iterasyonlara durdurma koşulu sağlanana kadar devam edilir.



Şekil 4.1: Problemin başlangıç çözümünde kullanılan YAA'nın akış şeması.

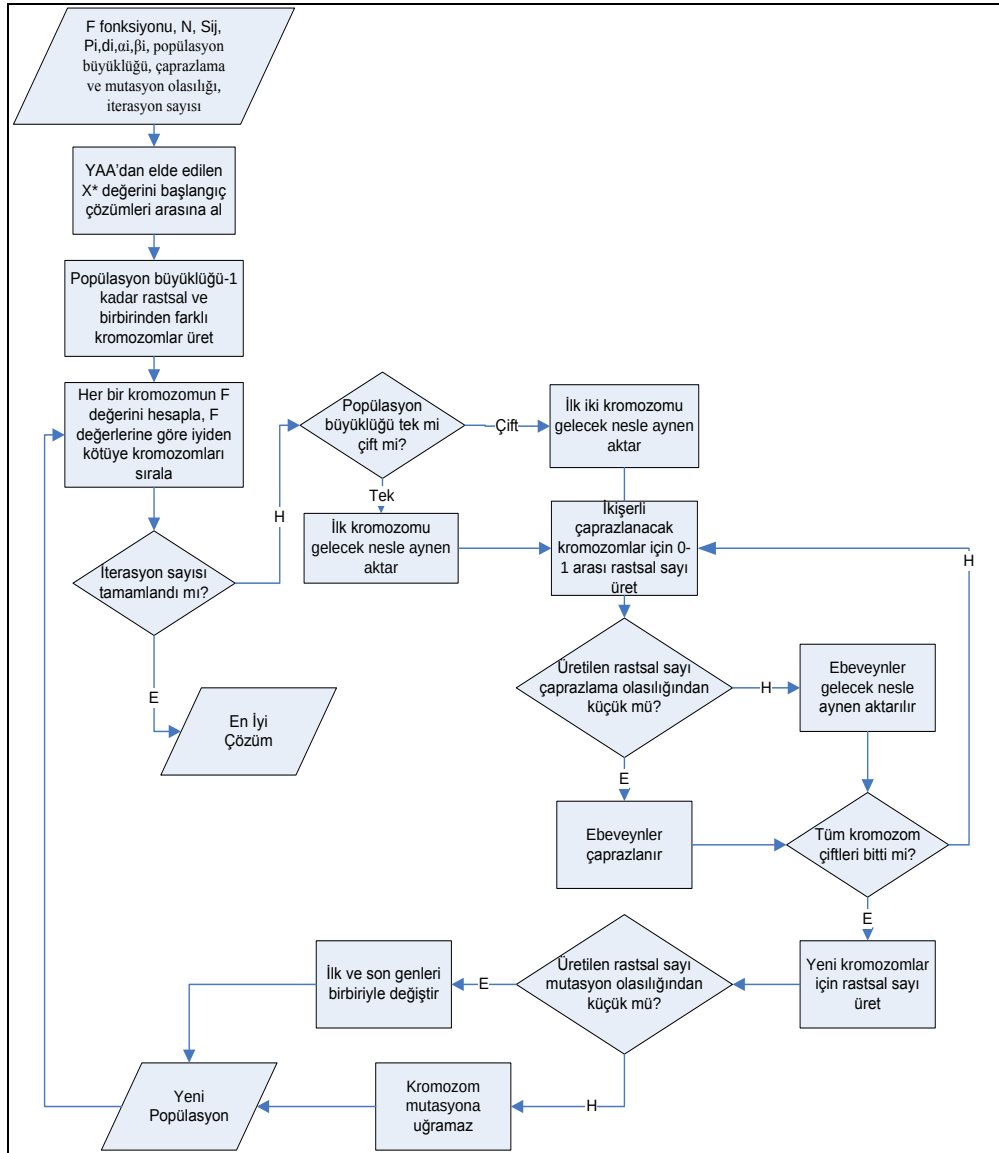
4.2.2. Problemin Çözümünde Kullanılan Genetik Algoritma

Problemin çözümünde kullanılan GA'nın akış şeması şekil 4.2'de verilmiştir.

4.2.2.1. Başlangıç Aşaması

- Başlangıçta F fonksiyonu, $F_{min}(\sum_{i=1}^N \alpha_i E_i + \beta_i T_i)$ ve N , S_{ij} matrisi, P_i , d_i , α_i , β_i , popülasyon büyüklüğü, çaprazlama olasılığı, mutasyon olasılığı, iterasyon sayısı gibi parametreler belirlenir. Kodlama yapısının nasıl olacağı, seçim, çaprazlama ve mutasyon metotlarının nasıl yapılacağı belirlenir. Bu çalışmada, popülasyon büyüklüğü N , çaprazlama olasılığı %50'den büyük değerler, mutasyon olasılığı %1 olarak alınmıştır. Kodlama yapısı

- ii. permutasyon kodlama, seçim metodu olarak seçkinlik, çaprazlama metodu olarak iki noktalı çaprazlama ve mutasyon metodu olarak ilk ve son sıraların yer değiştirmesi uygulanmıştır.
- iii. Tabu arama algoritmasından dönen X^* değeri başlangıç çözümü içerisine alınır.
- iv. (Popülasyon büyüklüğü-1) kadar rastsal ve birbirinden farklı sıralar (kromozomlar) üretilir.
- v. Her bir kromozomun F (uygunluk) değeri hesaplatılır.



Şekil 4.2: Problemin çözümünde kullanılan GA'nın akış şeması.

4.2.2.2. İteratif Aşamalar:

- i. Kromozomlar F değerlerine bakılarak en uygundan geriye doğru sıralanır. Eğer, popülasyon büyüklüğü tek ise ilk en uygun kromozom, çift ise en uygun ilk iki kromozom seçilerek bir sonraki nesle aynen aktarılır. Kalan kromozomlar da sırayla çiftlerli olarak çaprazlama olasılığına göre çaprazlanır. (Burada en iyi kromozomların kendi aralarında, en kötü kromozomların da kendi aralarında çaprazlanacağı düşünülür. Seçilen en iyi kromozom ise, daima korunur.)
 - a. Çaprazlama olasılığı kullanıcı tarafından girilir. Her bir kromozom için 0-1 arası rastsal sayı üretilir ve bu sayıların kullanıcı tarafından girilen 0,5-1,0 arasındaki çaprazlama olasılığından küçük olup olmadığı kontrol edilir. Küçükse çaprazlama olur, değilse ebeveyn kromozom çifti aynen gelecek nesle aktarılır.
 - b. Problemden iş sayısı tek ise kromozom, $((N-1)/2)$. elemandan, çift ise $((N-2)/2)$. elemandan bölünerek çaprazlama gerçekleştirilir. Bu işlem esnasında bir kromozomda bir iş numarasından yalnızca 1 tane olması gerektiği için, 2 tane olan işlerden 2.si silinir ve yerine kromozomda olmayan iş yazılır. Bu şekilde çaprazlamayla yeni kromozomlar oluşturulur.
- ii. Çaprazlama sonucu oluşan her bir kromozom için 0-1 arasında rastsal sayı üretilir. Bu rastsal sayıların her biri kullanıcı tarafından girilen 0-0,01 arasındaki mutasyon olasılığıyla karşılaştırılır. Rastsal sayının bu olasılıktan küçük olduğu durumlarda kromozom mutasyona uğrar. Diğer durumlarda çaprazlanan kromozom aynen yeni nesle aktarılır.
- iii. Bütün bu işlemler sonucu yeni popülasyon oluşmuş olur. Kullanıcı tarafından programa girilen iterasyon (nesil) sayısı değerine ulaşana kadar iteratif aşamalar tekrar tekrar döndürülür. İterasyon sayısı tamamlandığında, popülasyonun 1. sırasındaki kromozom bizim en uygun sıramız olur ve ekrana sıralama ve F değeri yazdırılır.

4.3. Önerilen Meta-sezgisel Algoritmanın Test Edilmesi

Daha önce de belirtildiği gibi meta-sezgisel algoritmalar, problemin en iyi çözümünü bulmayı garanti etmezler. Bu durumda da istenen, algoritmanın en iyi çözüme

mümkün olduğu kadar çok yaklaşmasıdır. Algoritma en iyi çözüme ne kadar çok yaklaşırsa performansının da o kadar yüksek olduğu söylenebilir.

Yukarıda anlatılan algoritma, MATLAB Version 7.7.0.471 (R2008B) programında kodlanmış ve en iyi çözümü bilenen problemler üzerinde test edilmiştir. Test problemleri, Eren ve Güner'in [10] 2007 yılında yayınladıkları makaledeki deneysel tasarım veri setine göre, işlem süreleri ortalaması (P_i) 100 ve standart sapması 25 olacak şekilde normal dağılıma göre, hazırlık zamanları (S_{ij}) 0 ile 19 arasında değişen ve teslim tarihleri (d_i) toplam işlem sürelerinin %40 ve %60'ı arasında değişen düzgün dağılıma göre Minitab 15.1.20.0 programında hazırlanmıştır. Farklı olarak iş sayıları (N) 4,5,6,7,8 ve 10 olarak alınmış, her bir iş kümesi için de erken bitirme ve geç bitirme cezaları (α, β) (0,25-0,50) ve (0,50-0,50) olarak alınmıştır.

Geçiş matrisleri Floyd-Warshall algoritması kullanılarak MATLAB Version 7.7.0.471 (R2008B) programında yazılan kodlarla üçgensel matrislere dönüştürülmüş ve elde edilen test problemleri Coleman'ın [100] bahsettiği matematiksel modelle Gams 23.5'de kodlanarak en iyi çözümler elde edilmiştir.

Test problemleri önerilen meta-sezgisel algoritmada 100'er kez çalıştırılmıştır. Eren ve Güner'in [10] sezgisel yöntem hatası formülü kullanılarak, her bir problemin her bir çalışmasında elde edilen sonucun % sezgisel yöntem hatası hesaplanmış ve her bir problem için ortalama sezgisel yöntem hatası bulunmuştur. Sonuçlar Çizelge 4.3'deki gibidir.

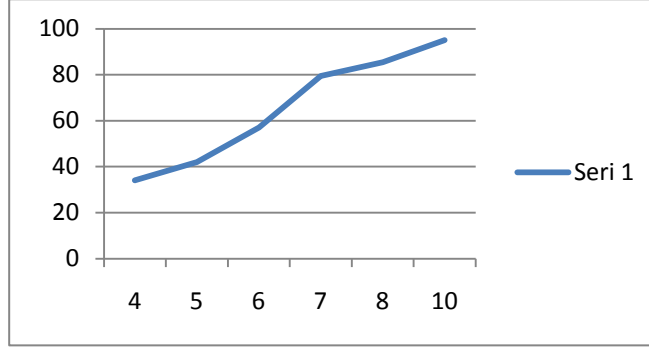
Çizelge 4.3'e bakıldığında problemin boyutu büyüdükçe önerilen algoritmanın ortalama sezgisel hatası da büyümektedir. Ortalama sezgisel hatanın yanı sıra en küçük ve en büyük sezgisel hatalar da artış göstermektedir. Küçük boyutlu problemlerde en iyiye ulaşma oranı %60'larda iken; problemin boyutu büyüdükçe, bu oranın %0'a yaklaştığı da görülmektedir.

Çözülen problemlerde yasaklı arama algoritmasının çözümüyle, genetik algoritmanın çözümü incelendiğinde problemin boyutu büyüdükçe genetik algoritmanın da işlevselliğinin arttığı görülmektedir. Şöyle ki; önerilen algoritmada, yasaklı arama algoritması sadece bir ilk çözüm bulma amacıyla, genetik algoritma ise; bu ilk çözümü geliştirip en iyiye yaklaşma amacıyla kullanılmaktadır.

Çizelge 4.3: Önerilen meta-sezgisel algoritmanın deney sonuçları.

Prob. No	Problem Parametreleri	Algoritma Parametreleri		En Küçük Sezgisel Hata	Ortalama Sezgisel Hata	En Büyük Sezgisel Hata	En İyiye Ulaşma Sayısı
		TA	GA				
1	N=4 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,25-0,50)	İterasyon Say.: 4 Tabu Listesi Uz.: 4	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%0,00	%0,59	%2,44	57
2	N=4 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,50-0,50)	İterasyon Say.: 4 Tabu Listesi Uz.: 4	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%0,00	%0,37	%1,63	59
3	N=5 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,25-0,50)	İterasyon Say.: 5 Tabu Listesi Uz.: 5	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%0,00	%2,07	%7,63	60
4	N=5 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,50-0,50)	İterasyon Say.: 5 Tabu Listesi Uz.: 5	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%0,00	%1,34	%3,72	59
5	N=6 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,25-0,50)	İterasyon Say.: 6 Tabu Listesi Uz.: 6	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%0,00	%2,88	%9,80	20
6	N=6 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,50-0,50)	İterasyon Say.: 6 Tabu Listesi Uz.: 6	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%0,00	%2,29	%9,32	32
7	N=7 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,25-0,50)	İterasyon Say.: 7 Tabu Listesi Uz.: 7	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%0,00	%3,63	%8,67	7
8	N=7 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,50-0,50)	İterasyon Say.: 7 Tabu Listesi Uz.: 7	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%0,05	%3,06	11,28	0
9	N=8 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,25-0,50)	İterasyon Say.: 8 Tabu Listesi Uz.: 7	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%0,00	%8,18	%17,83	2
10	N=8 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,50-0,50)	İterasyon Say.: 8 Tabu Listesi Uz.: 7	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%0,00	%6,61	%32,68	4
11	N=10 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,25-0,50)	İterasyon Say.: 10 Tabu Listesi Uz.: 7	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%4,34	%13	%19,18	0
12	N=10 Pi ~ N(100,25) Sij ~ U(0,19) di ~ U(0,4ΣPi,0,6ΣPi) (α,β)=(0,50-0,50)	İterasyon Say.: 10 Tabu Listesi Uz.: 7	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%3,39	%11,71	%20,31	0

Şekil 4.1’de de görülebileceği gibi her bir problem için yapılan 200 deney sonucunda, problem N=4 olduğunda %34, N=5 olduğunda %42, N=6 olduğunda %57, N=7 olduğunda %79,5, N=8 olduğunda %85,5, N=10 olduğunda da %95’lik bir iyileştirme oranıyla problem büyüklüğü arttıkça genetik algoritmanın işlevselliğinin arttığı gözlenmiştir.



Şekil 4.3: İş sayısı ve GA'nın işlevselliği arasındaki ilişki.

Yukarıda bahsedilen incelemelerde sadece bazı problem parametreleri değiştirilmiş ve problemin boyutuna göre yüzde değerlendirmeler yapılmıştır. Önerilen algoritmanın parametrelerinde yasaklı arama algoritmasının iterasyon sayısı ve tabu listesi haricinde herhangi bir parametre değişikliği yapılmamıştır. Bir sonraki adımda problemin çözümü üzerinde, problem parametrelerinin ve algoritma parametrelerinin etkilerini gözlemleyebilmek için istatistiksel bir çalışma yapılacaktır.

4.3.1. İstatistiksel Analiz

4.3.1.1. Problem Parametrelerinin Değişim Durumu

Çizelge 4.3'te aynı problemin ceza katsayıları olan (α, β) parametreleri değiştirildiğinde ve ceza parametreleri aynı kalıp iş sayısı değiştirildiğinde ortalama sezgisel hata oranlarında bazı değişimler görülmektedir. Bu değişimlerin anlamlı olup olmadığını verilerin varyans analizini yaparak inceleyebiliriz. Hipotezi şöyle kuralım:

$$H_0: \mu_{\text{iş sayısı}} = 0$$

$$H_1: \mu_{\text{iş sayısı}} \neq 0;$$

$$H_0: \mu_{\text{ceza}} = 0$$

$$H_1: \mu_{\text{ceza}} \neq 0;$$

Çizelge 4.3'te de görülen ortalama sezgisel hata verilerini kullanarak, %95 güven düzeyinde Minitab 15.1.20.0 programında yapılan varyans analizinin sonuçları Çizelge 4.4'te verilmiştir.

Çizelge 4.4: Önerilen algoritmanın deneyleri için varyans analizi.

Faktör	Serbestlik Derecesi	Kareler Toplamı	Ortalama Kare	F ₀ Değeri	P Olasılığı
İş Sayısı	5	197,862	39,5724	312,46	0,000
Ceza Değeri	1	2,058	2,0584	16,25	0,010
Hata	5	0,633	0,1266		
Toplam	11	200,554			

Çizelge 4.4' deki olasılık değerlerine bakıldığında $\alpha=0,05$ anlamlılık düzeyinde hipotezler şöyle yorumlanır:

$P(F_0>312,46)=0,000 < 0,05$ olduğundan H_0 hipotezi reddedilir, yani iş sayısı ortalama sezgisel hata üzerinde istatistiksel olarak etkilidir.

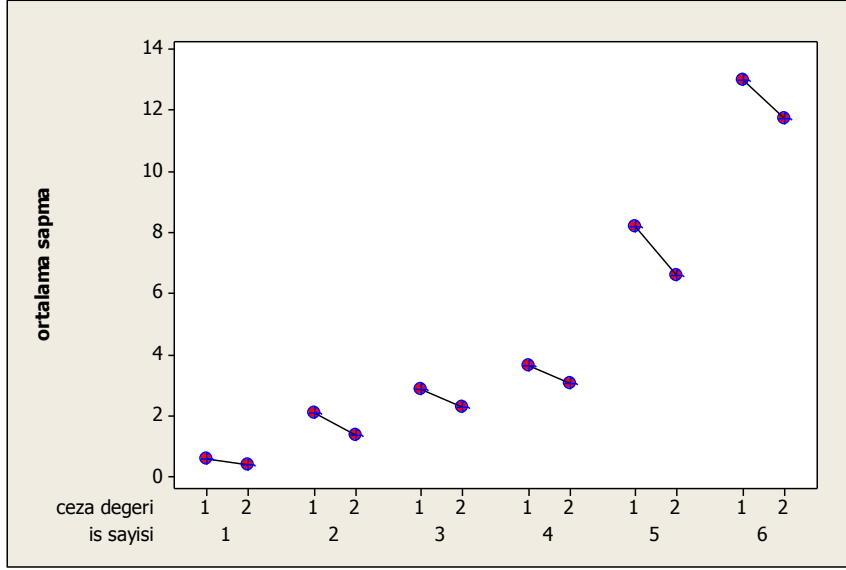
$P(F_0>16,25)=0,010 < 0,05$ olduğundan H_0 hipotezi reddedilir, yani ceza değeri ortalama sezgisel hata üzerinde istatistiksel olarak etkilidir.

Şekil 4.2'deki grafiğe bakıldığında iş sayısındaki artışın ve ceza değerindeki değişimin ortalama sapmayı üstel olarak artırdığı görülür.

Yukarıdaki analizin sonucunda problemin iş sayısı parametresiyle ceza değerleri parametresinin önerilen algoritmanın ortalama sezgisel hatası üzerinde etkisinin olduğu anlaşılmıştır. Şimdi de problem parametreleri sabit alındığında, algoritma parametrelerinin ortalama sezgisel hata üzerindeki etkisini inceleyelim.

4.3.1.2. Yasaklı Arama Algoritması Parametrelerinin Değişim Durumu

Meta-sezgisel algoritmaların kullanılma amacı, büyük boyutlu problemleri daha kısa sürede ve en iyiye en yakın olarak çözmektir. Bu yüzden, veri seti içerisindeki en büyük boyutlu test problemini kullanacağız. Öncelikle, başlangıç çözümünü bulmak için kullandığımız yasaklı arama algoritmasının parametrelerinin değişiminin hem yasaklı arama algoritması ortalama sezgisel hatası, hem de tüm algoritmanın ortalama sezgisel hatası üzerindeki etkisini inceleyelim.



Şekil 4.4: İş sayısı ve ceza değerine bağlı olarak OSH'nin değişimi.

İş sayısı $N=10$, ceza değerleri $(\alpha, \beta)=(0,5-0,5)$ olan 12. problemin verilerini kullanarak yasaklı arama algoritmasının iterasyon sayısını 10, 20 ve 50 olarak, tabu listesi uzunluğunu 7 ve 10 olarak değiştirdiğimizde, yasaklı arama algoritması ortalama hatası ve tüm algoritmanın ortalama hatası üzerindeki etkisini incelemek için deneyler yapılmıştır. Yapılan deneylerin sonuçları Çizelge 4.5'teki gibidir. Şekil 4.3'teki grafik, Çizelge 4.5'teki verilerin düzenlenmiş halidir. Grafiğe bakıldığında iterasyon sayısı değişimlerinin OSH ve YAOH'ı, tabu listesi uzunluğu değişimlerine göre daha fazla etkilediği görülebilir.

Bu değişimlerin OSH ve YAOH üzerindeki etkilerinin varlığını belirlemek için verilere göre %95 güven düzeyinde varyans analizi yapılmalıdır. Hipotezi şöyle kuralım:

$$H_0: \mu_{\text{iterasyonYAOH}}=0$$

$$H_1: \mu_{\text{iterasyonYAOH}} \neq 0;$$

$$H_0: \mu_{\text{iterasyonOSH}}=0$$

$$H_1: \mu_{\text{iterasyonOSH}} \neq 0;$$

$$H_0: \mu_{\text{tabulistesiyAOH}}=0$$

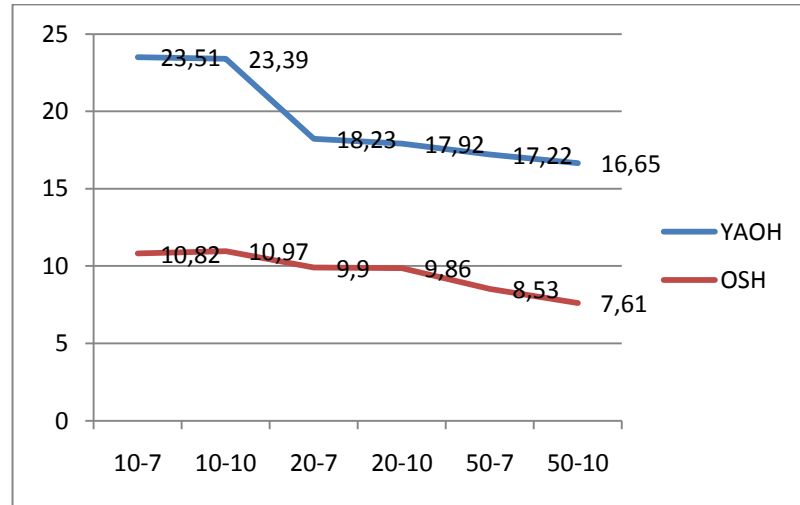
$$H_1: \mu_{\text{tabulistesiyAOH}} \neq 0;$$

$$H_0: \mu_{\text{tabulistesiosH}}=0$$

$$H_1: \mu_{\text{tabulistesiosH}} \neq 0;$$

Çizelge 4.5: YAA parametre değişiminin deney sonuçları.

Problem Parametreleri	Algoritma Parametreleri		Yasaklı Arama Ortalama Hatası (YAOH)	Ortalama Sezgisel Hata (OSH)
	Yasaklı Arama	Genetik Algoritma		
$N=10$ $P_i \sim N(100,25)$ $S_{ij} \sim U(0,19)$ $d_i \sim U(0,4\Sigma P_i, 0,6\Sigma P_i)$ $(\alpha, \beta) = (0,50-0,50)$	İterasyon Say.: 10 Tabu Listesi Uz.: 7	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%23,51	%10,82
	İterasyon Say.: 20 Tabu Listesi Uz.: 7		%17,22	%8,53
	İterasyon Say.: 50 Tabu Listesi Uz.: 7		%16,65	%7,61
	İterasyon Say.: 10 Tabu Listesi Uz.: 10		%23,39	%10,56
	İterasyon Say.: 20 Tabu Listesi Uz.: 10		%17,92	%9,86
	İterasyon Say.: 50 Tabu Listesi Uz.: 10		%16,65	%7,61



Şekil 4.5: YAA parametre değişiminin deney sonuçlarının grafiği.

Çizelge 4.5'te de görülen ortalama sezgisel hata verilerini kullanarak, %95 güven düzeyinde Minitab 15.1.20.0 programında yapılan varyans analizinin sonuçları Çizelge 4.6 ve Çizelge 4.7'e verilmiştir.

Çizelge 4.6: YAA iterasyon sayısı ve tabu listesi uzunluğunun YAA'nın kendi çözümü üzerindeki etkisini gösteren deney sonuçları.

Faktör	Serbestlik Derecesi	Sıralı Kareler Toplamı	Düzenlenmiş Kareler Toplamı	Düzenlenmiş Ortalama Kare	F ₀ Değeri	P Olasılığı
YAA İterasyon Sayısı	2	48,424	48,424	24,212	948,86	0,001
Tabu Listesi Uzunluğu	1	0,167	0,167	0,167	6,53	0,125
Hata	2	0,051	0,051	0,026		
Toplam	5	48,641				

Çizelge 4.7: YAA iterasyon sayısı ve tabu listesi uzunluğunun tüm algoritma çözümü üzerindeki etkisini gösteren deney sonuçları.

Faktör	Serbestlik Derecesi	Sıralı Kareler Toplamı	Düzenlenmiş Kareler Toplamı	Düzenlenmiş Ortalama Kare	F ₀ Değeri	P Olasılığı
YAA İterasyon Sayısı	2	7,1977	7,1977	3,5989	34,32	0,028
Tabu Listesi Uzunluğu	1	0,2481	0,2481	0,2481	2,37	0,264
Hata	2	0,2097	0,2097	0,1049		
Toplam	5	7,6555				

Çizelge 4.6' daki olasılık değerlerine bakıldığında $\alpha=0,05$ anlamlılık düzeyinde hipotezler şöyle yorumlanır:

$P(F_0>294,78)=0,001 < 0,05$ olduğundan H_0 hipotezi reddedilir, yani YAA'nın iterasyon sayısı YAA'nın ortalama sezgisel hatası üzerinde istatistiksel olarak etkilidir.

$P(F_0>38,93)=0,125 > 0,05$ olduğundan H_0 hipotezi kabul edilir, yani tabu listesi uzunluğu YAA'nın ortalama sezgisel hatası üzerinde istatistiksel olarak etkili değildir.

Çizelge 4.7' deki olasılık değerlerine bakıldığında $\alpha=0,05$ anlamlılık düzeyinde hipotezler şöyle yorumlanır:

$P(F_0>22,97)=0,028 < 0,05$ olduğundan H_0 hipotezi kabul edilir, yani YAA'nın iterasyon sayısı tüm algoritmanın ortalama sezgisel hatası üzerinde istatistiksel olarak etkilidir.

$P(F_0 > 4,47) = 0,264 > 0,05$ olduğundan H_0 hipotezi kabul edilir, yani tabu listesi uzunluğu tüm algoritmanın ortalama sezgisel hatası üzerinde istatistiksel olarak etkili değildir.

Bu sonuçları tekrar gözden geçirmek gerekirse $N=10$ büyüklüğünde bir problemin başlangıç çözümünde kullanılan YAA'nın tabu listesi uzunluğu 7'den 10'a değiştirildiğinde bu değişimin YAA'nın kendi çözümü ve tüm algoritmanın çözümü üzerinde etkili olmadığı görülmektedir. Aynı problemde YAA'nın iterasyon sayısı 10'dan 20'ye ve 50'ye çıkarıldığında YAA'nın kendi çözümü üzerinde ve tüm algoritmanın çözümü üzerinde etkili olduğu görülmektedir.

4.3.1.3. Genetik Algoritma Parametrelerinin Değişim Durumu

Genetik algoritma parametrelerinden bazılarının değişimi durumunda ortalama sezgisel hatanın bu değişimlerden etkilenip etkilenmediğini araştırmak için, öncelikle diğer parametreleri sabit olan bir problem belirleyelim. Bu analize ait verileri seçerken, veri setindeki 5. bölümdeki örnek olayın parametrelerine en uygun parametrelere sahip olan problemi seçmeye özen gösterilmiştir.

İş sayısı $N=10$ olan, ceza katsayıları $(\alpha, \beta) = (0,25-0,50)$ olan 11. problemi göz önüne alalım. Yasaklı arama algoritması parametrelerini de, iyi çözüm elde etmek için varyans analizi testinden çıkan sonuçları Şekil 4.3'e bakıp yorumlayarak, iterasyon sayısını büyük olan değer yani 50, tabu listesi uzunluğunu da küçük olan değer yani 7 alalım. Genetik arama algoritmasında değişimlerini gözlemlemek istediğim parametreler 100 ve 300 olarak değişen iterasyon sayısı, 89 ve 99 olarak değişen çaprazlama olasılığı ve 0,01 ve 0,05 olarak değişen mutasyon olasılığıdır. Buna göre yapılan deneylerde Çizelge 4.8'deki sonuçlar elde edilmiştir.

Bu değişimlerin OSH üzerindeki etkilerinin varlığını belirlemek için verilere göre %95 güven düzeyinde varyans analizi yapalım. Hipotezi şöyle kuralım:

$$H_0: \mu_{\text{iterasyon}} = 0$$

$$H_1: \mu_{\text{iterasyon}} \neq 0;$$

$$H_0: \mu_{\text{çaprazlamaolasiligi}} = 0$$

$$H_1: \mu_{\text{çaprazlamaolasiligi}} \neq 0;$$

$$H_0: \mu_{\text{mutasyonolasiligi}} = 0$$

$H_1: \mu_{\text{mutasyonolasılığı}} \neq 0;$

Çizelge 4.8’de görülen ortalama sezgisel hata verilerini kullanarak, %95 güven düzeyinde Minitab 15.1.20.0 programında yapılan varyans analizinin sonuçları Çizelge 4.9’da verilmiştir.

Çizelge 4.8: GA parametre değişiminin deney sonuçları.

Problem Parametreleri	Algoritma Parametreleri		Ortalama Sezgisel Hata (OSH)
	Yasaklı Arama	Genetik Algoritma	
N=10 $P_i \sim N(100,25)$ $S_{ij} \sim U(0,19)$ $d_i \sim U(0,4\Sigma P_i, 0,6\Sigma P_i)$ $(\alpha, \beta) = (0,25-0,75)$	İterasyon Say.: 50 Tabu Listesi Uz.: 7	İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%11,49
		İterasyon Say.:300 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,01	%6,84
		İterasyon Say.:100 Çaprazlama Ols.:0,99 Mutasyon Ols.: 0,01	%11,02
		İterasyon Say.:300 Çaprazlama Ols.:0,99 Mutasyon Ols.: 0,01	%6,65
		İterasyon Say.:100 Çaprazlama Ols.:0,99 Mutasyon Ols.: 0,05	%10,15
		İterasyon Say.:300 Çaprazlama Ols.:0,99 Mutasyon Ols.: 0,05	%6,72
		İterasyon Say.:100 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,05	%10,81
		İterasyon Say.:300 Çaprazlama Ols.:0,89 Mutasyon Ols.: 0,05	%6,93

Çizelge 4.9’ daki olasılık değerlerine bakıldığında $\alpha=0,05$ anlamlılık düzeyinde hipotezler şöyle yorumlanır:

$P(F_0 > 301,17)=0,000 < 0,05$ olduğundan H_0 hipotezi reddedilir, yani GA’nın iterasyon sayısı ortalama sezgisel hata üzerinde istatistiksel olarak etkilidir.

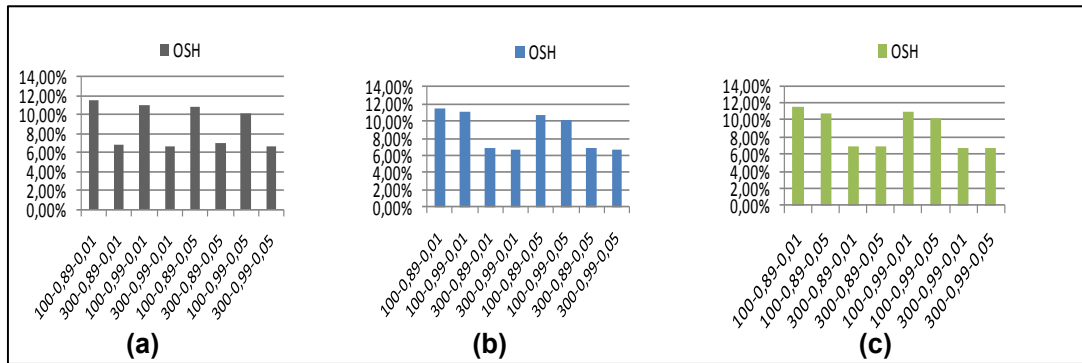
$P(F_0 > 2,65)=0,179 > 0,05$ olduğundan H_0 hipotezi kabul edilir, yani çaprazlama olasılığı ortalama sezgisel hata üzerinde istatistiksel olarak etkili değildir.

Çizelge 4.9: GA iterasyon sayısı, çaprazlama ve mutasyon olasılığının OSH üzerindeki etkisini gösteren deney sonuçları

Faktör	Serbestlik Derecesi	Sıralı Kareler Toplamı	Düzenlenmiş Kareler Toplamı	Düzenlenmiş Ortalama Kare	F ₀ Değeri	P Olasılığı
GA İterasyon Sayısı	1	33,334	33,334	33,334	301,17	0,000
Çaprazlama Olasılığı	1	0,293	0,293	0,293	2,65	0,179
Mutasyon Olasılığı	1	0,242	0,242	0,242	2,19	0,213
Hata	4	0,441	0,441	0,110		
Toplam	7	34,309				

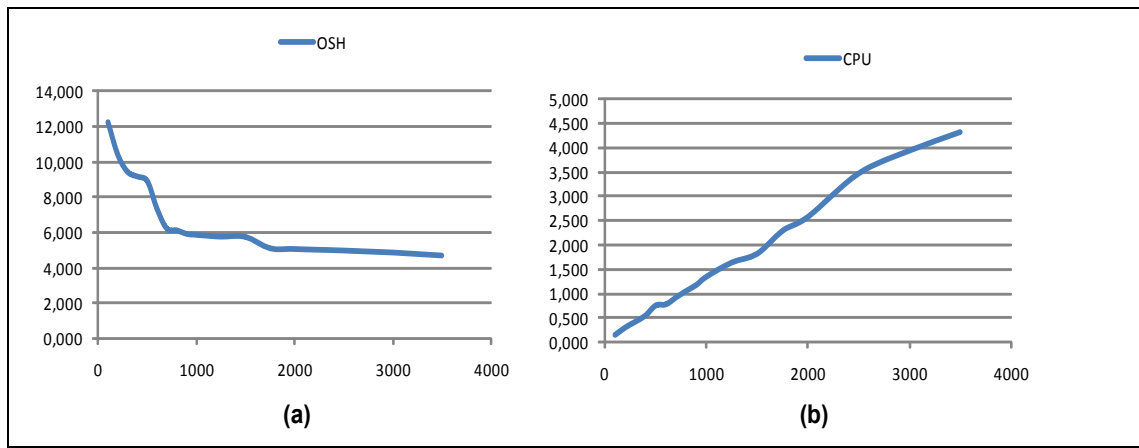
$P(F_0 > 2,19) = 0,213 > 0,05$ olduğundan H_0 hipotezi kabul edilir, yani mutasyon olasılığı ortalama sezgisel hata üzerinde istatistiksel olarak etkili değildir.

Varyans analizinden elde edilen sonuçlara göre, genetik algoritmanın iterasyon sayısını 100'den 300'e çıkarıldığında ortalama sezgisel hata üzerinde etkili olduğu görülmüştür. Şekil 4.4.a'ya bakıldığında, çaprazlama olasılığı ve mutasyon olasılığının aynı olduğu, iterasyon sayısının değiştiği sütunlarda iterasyon sayısı 100'den 300'e artırıldığında ortalama sezgisel hata üzerinde gözle görülebilir bir iyileşme sağladığı söylenebilir. Şekil 4.4.b'ye bakıldığında diğer parametreler sabit iken çaprazlama olasılığının %89'dan %90'a çıkarılması ortalama sezgisel hata üzerinde gözle görülebilir bir iyileşme sağlamadığı, Şekil 4.4.c'ye bakıldığında da diğer parametreler sabit iken mutasyon olasılığının %1'den %5'e çıkarıldığında ortalama sezgisel hata üzerinde bir iyileşme sağlamadığı söylenebilir. Burada mutasyon olasılığının daha yüksek seviyelerine bakılmamıştır, çünkü mutasyon olasılığının artması algoritmayı rastsal arama algoritmasına dönüştürmektedir.



Şekil 4.6: GA parametrelerinin değişim durumlarında OSH değişim grafiği.

Yukarıdaki verilere bakıldığında iterasyon sayısının artırılması ortalama sezgisel hatanın düşürülmesinde, dolayısıyla algoritmanın performansının artırılmasında önemlidir. Diğer bir taraftan da algoritmanın çözüm zamanı (CPU) da bizim için önemli bir performans ölçütüdür. Yukarıda ele alınan 11. problemi yasaklı arama algoritmasının tüm parametreleri sabit olarak (iterasyon sayısı 50, tabu listesi uzunluğu 7), genetik algoritmanın da çaprazlama ve mutasyon olasılıkları sabit olarak (çaprazlama olasılığı %89, mutasyon olasılığı %1) artan iterasyon sayısı ile Intel(R) Core (TM) 2 Duo CPU 2,00 GHz işlemciye sahip bir dizüstü bilgisayarda çözüldüğünde Şekil 4.5'teki grafikler elde edilmiştir.



Şekil 4.7: GA iterasyon sayısının değişim durumlarında OSH ve CPU değişim grafikleri.

Şekil 4.5'teki grafikler incelendiğinde problem 100 iterasyonla çözülürken %13 civarında olan ortalama sezgisel hatanın iterasyon sayısı 3500'e çıkarıldığında %4,5 civarına düştüğü söylenebilir. Yapılan deneyler sonucu 100 iterasyondan 1000 iterasyona kadar ortalama sezgisel hatanın hızlı bir düşüş gösterdiği, 1000 iterasyondan sonraki her artırımda daha az bir düşüş eğilimi gösterdiği görülmüştür. Örneğin; 2000 iterasyon yapıldığında ortalama sezgisel hata, yaklaşık %5 iken, 3500 iterasyon yapıldığında yaklaşık %4,7 olmaktadır. Fakat; 2000 iterasyon ile program 1 kez çalıştırıldığında 2,58 sn'de sonuç verirken, 3500 iterasyon ile çalıştırıldığında 4,32 sn'de sonuç vermektedir. Buna göre %67,44 uzun sürede %0,3'lük bir iyileştirme söz konusudur. Ancak; bu hususta seçim kullanıcıya göre de farklı değerlendirilebilir. Süre saniye veya yüksek sayıdaki koşumlarda dakika cinsinden olacağı için %67,44'lük bir oran, iyileşme söz konusu olduğu için göz ardı da edilebilir.

Son olarak algoritmayı kaç kez çalıştırılacağına, yani koşum sayısına karar verilmelidir. Bunun için yine 5. bölümde anlatılacak olan örnek vaka problemine en çok benzeyen test problemi olan 11. problem ele alınmış, yasaklı arama algoritması parametreleri olarak iterasyon sayısı 50, tabu listesi uzunluğu 7, genetik algoritma parametresi olarak iterasyon sayısı 2000, çaprazlama olasılığı %89, mutasyon olasılığı %1 kullanılmıştır. Problemin çözümü için algoritma 41 kez çalıştırılmış ve sezgisel hata ortalaması 5,063, standart sapması 2,868 olan örneklem elde edilmiştir. 0,95 güven düzeyinde sezgisel hata ortalamasının, 0,56 farkla yani %4,50 olarak tahmin edilebilmesi için örneklem büyüklüğü n'in değeri şu şekilde hesaplanır:

$L = (t_{1-\alpha, (n-1)} \cdot \frac{S_x}{\sqrt{n}})$ formülünde L yerine örneklem sezgisel hata ortalaması ile istenen sezgisel hata ortalaması arasındaki fark yazılır. Kitle varyansının bilinmediği durumlarda t- dağılımı kullanılır fakat; merkezi limit teoremine göre örneklem büyüklüğü arttıkça t-dağılımı normal dağılıma yaklaşır. Buradaki örneklem büyüklüğü olan 41, t-dağılımı yerine normal dağılım kullanılması için yeterli büyüklüktedir.

$$Z_{0,95}=1,65$$

$$S_x=2,868$$

$$\bar{X}=5,063$$

$\mu= 4,50$. Bu bilgiler ışığında şu işlemler yapılır:

$$\bar{X} - \mu = Z_{0,95} \cdot \frac{S_x}{\sqrt{n}} = (5,063 - 4,50) = 1,65 \cdot \frac{2,868}{\sqrt{n}}$$

$$\sqrt{n} = \frac{4,732}{0,563} = 8,405$$

$n = 71$ defa algoritma koşturulmalıdır.

4.3.2. İstatistiksel Analiz Sonuçları

Yapılan hesaplara göre 5. bölümde ele alınacak olan iş sayısı $N= 15$ büyüklüğünde ve değişken ceza katsayılarının, teslim tarihlerinin olduğu örnek olay probleminde kullanılacak olan parametrelere karar verilmiştir. Buna göre; yasaklı arama algoritması en az 50 iterasyon yapacak ve 50 iterasyondan sonra gelen iterasyonlarda arka arkaya 3 iterasyonda iyileşme olmadığı takdirde duracak. Yasaklı arama

algoritmasının tabu listesinin hafızası 7 olacaktır. Genetik arama algoritması 2000 iterasyonla, %89 çaprazlama olasılığıyla ve %1 mutasyon olasılığıyla çalışacaktır. Program en az 71 defa koşturularak, sonuçları incelenecektir.

5. GIDA SEKTÖRÜNDE BİR UYGULAMA ÇALIŞMASI

Bu bölümde gıda sektöründe faaliyet gösteren bir işletme içerisinde var olan hazırlık zamanı sıra bağımlı, farklı teslim tarihli ve farklı erken ve geç bitirme cezalarının olduğu, amacı tam zamanlı üretim sisteminin uygulanması olan tek makineli çizelgeleme problemi incelenecektir.

5.1. İşletmenin Tanıtımı

Probleme geçmeden önce kısaca işletmeyi tanıyalım. Gıda sektöründe 1996 yılından beri faaliyet gösteren ABC Gıda San. ve Tic. A.Ş. süt ve süt ürünleri üretimini gerçekleştirmektedir. 93,500 m² kapalı alana sahip fabrikası, lokasyon olarak şirkete yeterli miktarda yüksek kalitede çiğ süt toplam imkânı sağlamaktadır. Yapılan yaklaşık 150 milyon Dolar yatırımla süt alımdan bitmiş ürün yüklemesine kadar tam otomasyona sahip entegre bir tesis kurulmuştur.

ABC Gıda günlük 1.850 tonu aşan ortalama fiili süt işleme miktarı ile Türkiye'nin en büyük süt ve süt ürünleri üreticisidir. UHT süt (tam yağlı-yarım yağlı), meyve aromalı süt, puding, thick cream, light süt, laktaz süt, yoğurt çeşitleri ayran, kefir, nektar, krem şanti, kapuçino, salep, peynir çeşitleri, tereyağ, ketçap, mayonez gibi ürünler üretmektedir. ABC Gıda yurtiçi üretimin dışında, Amerika Birleşik Devletleri, Almanya, Angola, Azerbaycan, Bosna, Doğu Türkistan, Fildişi, Filistin, Gürcistan, İngiltere, Katar, Kıbrıs, Kırgızistan, Kuveyt, Singapur, Suudi Arabistan, Ürdün ve Yunanistan'a ihracat yapmaktadır. ABC Gıda San. ve Tic. A.Ş. yaklaşık 1200 çalışanıyla üretim tesisinde 7/24 3 vardiya faaliyet göstermektedir.

5.2. Problemin Tanımı

Bu bölümde ele alınan problem, taze peynir üretim tesisinde dolum yapan ve “Dosomat 1” olarak isimlendirilen makinada, ana üretim planına uygun olarak siparişlerin tam zamanında tesliminin gerçekleştirilmesi için en uygun iş sıralamasının oluşturulmasıdır.

5.2.1. Taze Peynir Üretimi

Taze peynir üretimi için yapılan işlemleri şöyle sıralayabiliriz:

- i. Taze peynir için otomasyondan alınan %20 yağlı pastörize süt taze peynir proses kısmında 4 adet bulunan 10 ton kapasiteli tanklara alınır.
- ii. Kimyasal analizleri yapılan süte %1 oranında kültür eklendikten sonra inkübasyona bırakılır. (L için 32 °C’de 5 saat, F için 22 °C’de 8 saat) Ph takibi yapılan süte, 4.6- 4.7 ph aralığında kırılma işlemi uygulanır.
- iii. Soğutucuda 8°C’ye kadar soğutularak aynı tanka alınır. Oluşan yarı mamul artık, krema adını almıştır ve akış hızını stabil hale getiren balans tankından geçer, ısıtıcı eşanjöre gelerek 82 °C’ye kadar ısıtılır.
- iv. Isınan krema “Reaksiyon Tankı” adı verilen delikli yapıya sahip bir çeşit separatörde teleme peynir altı suyu ayrışımı yapılır.
- v. Katkılama işleminin yapıldığı her biri 1250 kg kapasiteli 3 adet blenderda katkı maddeleri eklenir.
- vi. Borulu ve sıyrıcılı pastörizatörde 82°C’ de pastörize edilir. Daha sonra soğutma işlemine tabi tutulan teleme homojenizatörde iki kademeli basınç uygulanarak (L için 200-40 bar, F için 125-20 bar) homojenize hale getirilir.
- vii. Dolumdan önce ürünün dinlendirildiği 500 kg kapasiteli Creaming Tank’ta bekletilir.
- viii. Bekleme esnasında dolum makinasında ana üretim programına göre gerekli yıkama ve kalıp değişimi işlemleri yapılır, ardından mamul dolum makinasına basılır. Dolum makinasında, ambalajlara doldurulup paketleme işlemi otomatik olarak gerçekleştirilir.
- ix. Son olarak ambalajlanan ürün kolilenerek sevkiyata hazır hale getirilir.

5.2.2. Taze Peynir Üretimi İçin Ana Üretim Planı

Taze peynir üretimi için öncelikle siparişe göre aylık ana üretim planı hazırlanmaktadır. Bu planda hangi üründen ne kadar üretileceği ve işletmede sevkiyatlar haftalık yapıldığı için hangi ürünün hangi hafta sevk edileceği görülmektedir.

Çizelge 5.1’de firmanın Aralık 2010 tarihli ana üretim planı mevcuttur. Üretim planlamayı yapan kişi bu planı ve diğer tüm kısıtları göz önünde bulundurarak tam zamanlı üretim felsefesine göre işleyen bir sıralama çıkarmak zorundadır.

Çizelge 5.1: Aralık 2010 ana üretim planı.

No:	Çeşit	Kod	Veri (KG)				Toplam Plan
			P49	P50	P51	P52	
1	F	0xx01-06	10.511	-	-	-	10.511
2		0xx01-09	-	76.062	-	-	76.062
3		0xx11-03	6.140	-	-	-	6.140
4		0xx10-08	739	-	-	-	739
5		0xx06-01	-	27.097	-	-	27.097
	Toplam F		17.390	103.159	-	-	129.680
6	L	0xx14-02	9.157	-	-	-	9.157
7		0xx47-03	-	-	-	158.028	158.028
8		0xx01-05	42.390	-	-	-	42.390
9		0xx02-08	-	36.862	-	-	36.862
10		0xx25-07	8.465	-	-	-	8.465
11		0xx14-09	4.610	-	-	-	4.610
12		0xx03-05	1.301	-	-	-	1.300
13		0xx25-04	867	-	-	-	867
14		0xx14-04	867	-	-	-	867
15		0xx06-00	-	-	121.935	-	121.935
	Toplam L		67.657	36.862	121.935	158.028	384.484

Ana üretim planı dışındaki en önemli kısıtlardan biri kapasite kısıtıdır. Dolum makinasının günlük kapasitesi üretilen peynir çeşidine göre değişmektedir. Peynirin gramajı ne kadar yüksek olursa, dolum makinasının da günlük üretim kapasitesi o kadar artmaktadır. 100 gr’lık ambalajlar için 10.081 kg/gün, 180 gr’lık ambalajlar için 18.146 kg/gün, 200gr için 20.163 kg/gün, 275gr için 23.763 kg/gün ve 300 gr’lık ambalajlar için 24.195 kg/gün’lük kapasiteye sahiptir. Ayrıca işçilerin molaları ve

CIP olarak isimlendirilen rutin temizlik süreleri günde 4 saatlik bir duruşa neden olmaktadır. Günlük net 20 saat çalışır durumda olan makinede, her bir ürün için o ürüne ait kapasite değerlerine göre operasyon süreleri hesaplanır. Her bir üründen diğerlerine geçerken CIP süreleri ve kalıp değişim süreleri toplamı hazırlık zamanı olarak daha önceden belirlenmiştir. Çizelge 5.2’de Çizelge 5.1’deki ana üretim planına göre hazırlanmış, geçiş matrisini de içeren veri matrisi görülmektedir.

2010 Aralık ayı ana üretim programına göre 15 farklı çeşit ürünün, istenen miktarda ve istenen teslim zamanında hazır olması gerekmektedir. İş tanımları önceden bilinmektedir ve tüm işlerin “0” anında yapılmaya hazır olduğu düşünülür. Taze peynir dolum makinesi darboğaz oluşturduğundan, bu makinadaki sıralamaya göre önceki makinaların çizelgelemeleri düzenlenmektedir. İşler arasında herhangi bir öncüllük ardıllık zorunluluğu yoktur. Hazırlık zamanları işlerin sıralamasına göre değişir ve işlem sürelerinden ayrı tutulur. Taze peynir dolum makinesi duruşları için zorunlu zamanlar yoktur ve Çizelge 5.2’deki veriler bu makine duruşları hesaba katılarak belirlenmiştir. (Makinenin günlük olarak net 20 saat çalıştığı bilinmektedir.) İşler yapılmaya başlandıktan sonra yarıda kesilmediği varsayılmıştır.

Bu koşullar altında üretimde sıralama yapılmak istenmektedir. Üretimde erken üretip stoklama ve geç üretme istenmeyen bir durumdur. Dolayısıyla erken üretim ve geç üretim cezalandırılmaktadır. Bu cezalar, ürünlerin raf ömürleriyle de ilişkilendirilir. Üretimde kullanılan hammaddenin süt olmasından dolayı, piyasaya mümkün olan en taze haliyle çıkması istenmektedir. Örneğin; raf ömrü kısa olan bir ürünün istenen zamandan erken üretilmesi ile raf ömrü daha uzun olan bir ürünün istenen zamandan erken üretilmesi durumundaki cezaları aynı olmayacaktır. Ayrıca ihraç edilen ürünlerin raf ömürleri, yurtiçine verilen ürünlerin raf ömürlerinden farklı olabilmektedir. Ürün, termin süresinden daha geç üretildiğinde ise bir sonraki sevkiyat ile birleştirilmekte, bu mümkün değilse de bekletilmeyip sevk edilmektedir. Bu durumda ceza katsayılarını belirlemek oldukça tecrübe gerektirmektedir.

Problemin amaç fonksiyonunda kullanılan erkenlik ve geçlik ceza katsayıları, ürünlerin erken tamamlandıkları veya geciktikleri süreyle çarpılmak üzere işletmenin taze peynir üretim tesisinin planlamasını yapan tecrübeli planlamacısı tarafından öngörülmüştür. Çizelge 5.3’te ürünler için erkenlik ve geçlik ceza katsayıları matrisi verilmiştir. Tüm bu veriler kullanılarak 4. bölümde anlatılan algoritma ile taze peynir dolum makinasının iş sıralaması yapılmıştır.

Çizelge 5.2: Ürünler arası geçiş hazırlık zamanları matrisi, ürün operasyon ve termin süreleri.

Ürün Kodu	Ürün No	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Aralık'10 Sipariş Miktarı	Ürün Bazında Operasyon Süreleri (saat)	Aralık 2010 Ürün Termin Süreleri (di)
04314-02	1	0	0,25	0,25	4	4	4	4	2	2	2	2	4	2	2	2	9.157	18,17	140
04347-03	2	0,25	0	0,25	4	4	4	4	2	2	2	2	4	2	2	2	158.028	174,17	560
04301-05	3	0,25	0,25	0	4	4	4	4	2	2	2	2	4	2	2	2	42.391	42,05	140
04301-09	4	4	4	4	0	2	2	0,25	4	4	4	4	2	4	4	4	76.062	75,45	280
04306-01	5	4	4	4	2	0	2	2	4	4	4	4	0,25	4	4	4	27.097	22,4	280
04311-03	6	4	4	4	2	2	0	2	4	4	4	4	2	4	4	4	6.140	12,18	140
04301-06	7	4	4	4	0,25	2	2	0	4	4	4	4	2	4	4	4	10.511	10,43	140
04306-00	8	2	2	2	4	4	4	4	0	0,25	0,25	0,25	4	0,25	0,25	0,25	121.935	111	420
04302-08	9	2	2	2	4	4	4	4	0,25	0	0,25	0,25	4	0,25	0,25	0,25	36.863	31,03	280
04325-07	10	4	4	4	4	4	4	4	4	4	0	4	4	4	0,25	4	8.465	7,12	140
04314-09	11	4	4	4	4	4	4	4	4	4	4	0	4	4	4	0,25	4.610	3,9	140
04310-08	12	4	4	4	2	0,25	2	2	4	4	4	4	0	4	4	4	739	0,6	140
04303-05	13	2	2	2	4	4	4	4	0,25	0,25	0,25	0,25	4	0	0,25	0,25	1.301	1,08	140
04325-04	14	4	4	4	4	4	4	4	4	4	0,25	4	4	4	0	4	867	0,72	140
04314-04	15	4	4	4	4	4	4	4	4	4	4	0,25	4	4	4	0	867	0,72	140

Çizelge 5.3: Ürünlerin erkenlik ve geçlik cezaları.

Ürün No:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Erkenlik Cezası	2	4	5	5	5	3	3	5	4	4	4	4	4	4	4
Geçlik Cezası	8	6	5	5	5	7	7	5	6	6	6	6	6	6	6

5.3. Problemin Çözümü

Problemin çözümü için kullanılan yöntemde, 4. bölümde de anlatıldığı gibi, önce yasaklı arama algoritmasıyla başlangıç çözümü bulunmuş ardından bu çözüm genetik algoritmayla geliştirilerek optimum sıralamanın elde edilmesi hedeflenmiştir. Yine 4. bölümde bu algoritmaların parametre seçimleriyle ilgili detaylı bilgi verilmiştir. Yasaklı arama algoritması, tabu listesi hafızası 7 ve iterasyon sayısı en az 50 olmak üzere, 50 iterasyondan sonra yapılan iterasyonları arka arkaya gelen üç tanesinde iyileşme olmazsa durdurulması, elde edilen çözümün genetik algoritmanın başlangıç çözüm kümesine alınmasına kadar çalışmaktadır. Ardından çaprazlama olasılığı 0,89, mutasyon olasılığı 0,01 olan ve seçkinlik yöntemiyle en iyi kromozomun bir sonraki iterasyona aynen aktarıldığı genetik algoritma, 2000 iterasyonla çalışmaktadır. Bu algoritma her bir deneyde en az 71 defa olacak şekilde koşturulmuş ve sonuçlar Çizelge 5.4’de verilmiştir.

Çizelge 5.4’e bakıldığında her biri 10’ar kez olmak üzere 75, 80, 85 ve 90 koşumluk toplam 40 deney yapılmıştır. Bu deneylerden elde edilen en küçük ve en büyük amaç fonksiyonu değerleri ile amaç fonksiyonu değerlerinin ortalaması ve koşum başına ortalama çözüm süreleri hesaplanmıştır. Yapılan deneylerde en küçük amaç fonksiyonu değerlerine bakıldığında 8 defa 2140,57 değerinin tekrarlandığını ve diğer değerler içerisinde en küçüğü olduğu görülmüştür. Amaç fonksiyonu 2140,57 olan iş sıralaması 3-6-1-7-10-11-13-12-15-14-4-9-5-8-2 şeklindedir ve yapılan toplam $75 \times 10 + 80 \times 10 + 85 \times 10 + 90 \times 10 = 3300$ koşumda bulunan en iyi değerdir.

Bulunan en iyi sezgisel çözüm sonucunu 2140,57, en iyi çözüm olarak alırsak; 40 deneyde bulunan en iyi çözümlerin ortalama hatası da %0,9 olmaktadır.

Koşum sayısı çok daha büyük bir sayı olduğunda, ortalama amaç fonksiyonu değerinin hangi aralıkta olabileceğini %95 güven düzeyinde tahmin edelim.

Çizelge 5.4: Deney verileri çizelgesi.

Deney No	Koşturma Sayısı	En Küçük Amaç Fonksiyonu Değeri	Amaç Fonksiyonu Değerlerinin Ortalaması	En Büyük Amaç Fonksiyonu Değeri	Ortalama CPU Zamanı (sn)
1	75	2148,76	2890,381	3583,34	7,685
2	75	2153,16	2934,411	3404,43	7,774
3	75	2140,57	2898,289	3507,81	7,605
4	75	2179,56	2924,736	3530,33	7,203
5	75	2140,57	2888,767	3616,45	7,346
6	75	2170,81	3025,759	3667,48	7,746
7	75	2153,16	2879,114	3513,94	7,297
8	75	2140,57	2979,719	3556,07	7,658
9	75	2145,37	2846,906	3452,62	7,447
10	75	2154,41	2884,966	3535,86	7,311
11	80	2144,41	2966,947	3448,94	7,866
12	80	2144,41	2837,854	3497,34	7,981
13	80	2201,48	2975,981	3542,3	7,83
14	80	2140,57	2955,226	3520,09	7,879
15	80	2144,92	2901,161	3556,07	7,921
16	80	2158,92	2868,747	3448,94	7,653
17	80	2158,92	2981,958	3492,21	7,464
18	80	2153,16	2869,779	3524,71	7,44
19	80	2220,88	2884,301	3494,85	7,534
20	80	2178,6	2915,649	3550,68	7,439
21	85	2148,76	2941,652	3690,07	7,64
22	85	2178,6	2533,984	3558,23	7,451
23	85	2140,57	2850,083	3509,21	7,621
24	85	2232,42	2943,214	3522,23	7,413
25	85	2153,16	2860,717	3547,47	7,59
26	85	2144,41	2941,138	3574,45	7,653
27	85	2140,57	2891,217	3612,89	7,558
28	85	2140,57	2861,185	3607,31	7,61
29	85	2153,6	2939,503	3430,4	7,225
30	85	2232,42	2983,451	3561,01	7,459
31	90	2144,92	2978,685	3547,89	7,783
32	90	2145,88	2995,303	3801,05	7,64
33	90	2170,81	2879,36	3413,95	7,562
34	90	2144,92	2858,276	3583,21	7,678
35	90	2144,41	2960,092	3640,26	7,772
36	90	2153,16	2976,43	3453,79	7,654
37	90	2170,81	2895,4	3556,07	7,56
38	90	2158,92	2967,84	3583,21	7,801
39	90	2154,41	2908,285	3625,23	7,683
40	90	2140,57	2961,11	3591,140	7,655

Tüm koşumlar için örneklem ortalama değeri $\bar{X}=2910,939$ ve standart sapması $S_x=353,307$ 'dir. Örneklem sayısı $n=3300$ için merkezi limit teoreminden, t-dağılımının normal dağılıma yaklaştığını söyleyebiliriz. %95 güvenle $Z_{0,975}=1,96$ olduğunda göre, Ortalama amaç fonksiyonu değerinin aralığı hesaplanır.

$$\bar{X} - Z_{1-\frac{\alpha}{2}} \cdot \frac{\sigma}{\sqrt{n}} = 2910,939 - 1,96 \cdot \frac{353,307}{\sqrt{3300}} = 2898,88$$

$$\bar{X} + Z_{1-\frac{\alpha}{2}} \cdot \frac{\sigma}{\sqrt{n}} = 2910,939 + 1,96 \cdot \frac{353,307}{\sqrt{3300}} = 2922,994$$

Yukarıdaki hesaplamalara göre koşum sayısı sonsuza yaklaşırsa bulunan amaç fonksiyonu değerlerinin ortalaması $[2898,88;2922,994]$ aralığında olacaktır.

Her bir koşum için Intel(R) Core (TM) 2 Duo CPU 2,00 GHz işlemciye sahip bir dizüstü bilgisayarda geçen ortalama süre 7,62 sn olmaktadır.

6. SONUÇ VE ÖNERİLER

Bu çalışmada, hazırlık zamanı sıra bağımlı olan teslim süreleri, işlem süreleri ve erken bitirme ile geç bitirme cezaları işlere göre farklılık gösteren, amacı işlerin tam zamanında teslim edilmesi olan tek makineli ortamda iş sıralaması problemi ele alınmıştır. Üretimde sıralama ve çizelgeleme konusunda detaylı bir literatür çalışması yapılmış olup, bu çalışmaya konu olan problemin yaygın olarak çalışılmadığı görülmüştür. Literatürde küçük boyutlu ve üçgensel özellik taşımasıyla gezgin satıcı problemine benzeyen problemlerin çözümü için kullanılan bir matematiksel model incelenmiş ve buna karşılık yine küçük boyutlu problemlerin çözümü için kullanılan ve koşulsuz çalışmanın konusu olan tüm problemlere uygulanabilen bir matematiksel model önerilmiştir. Literatürdeki büyük boyutlu sıralama ve çizelgeleme problemleri için kullanılan meta-sezgisel yöntemler araştırılmıştır. İncelenen yöntemler içerisinde, yasaklı arama algoritmasının başlangıç çözümünü oluşturduğu ve genetik algoritmanın da bu çözümü iyileştirerek nihai çözümün elde edildiği bir algoritma geliştirilmiş ve kodları MATLAB Version 7.7.0.471 (R2008B) programında yazılmıştır. Yazılan algoritma literatürdeki verilere göre oluşturulan 12 farklı problem üzerinde deneylerle analiz edilmiş ve algoritma parametreleri, gıda sektöründe faaliyet gösteren bir işletmeden alınan problem verilerine uygun olarak belirlenmiştir. İşletmeden verileri alınan problem, önerilen meta-sezgisel algoritma ile çözdürülmüş ve sonuçları değerlendirilmiştir.

6.1. İstatistiksel Sonuçlar

4 işten 10 işe kadar hazırlanan 12 test problemi literatürde var olan matematiksel model ile çözümlenerek en iyi sonuçlar elde edilmiştir. Önerilen meta-sezgisel algoritma bu veriler ışığında test edilerek şu sonuçlar elde edilmiştir:

- i. İş sayısı problemin çözüm kalitesi üzerinde önemli bir etkidir. Algoritmanın, parametrelerine göre de değişen çözüm kalitesi, aynı algoritma parametreleriyle düşük sayıdaki iş için ortalama sezgisel hata %0,5 civarında iken, problem boyutu büyüdükçe %12,5 seviyelerine çıkmaktadır. Algoritma

- parametreleri uygun şekilde düzenlendiğinde ise; büyük boyutlu problemler için ortalama sezgisel hatanın %6,65 seviyelerine indiği gözlenmiştir.
- ii. Önerilen meta-sezgisel yöntemde yasaklı arama algoritmasının bulunduğu çözüm genetik algoritmayla iyileştirilmektedir. Bu durumun da iş sayısına bağlı olarak değiştiği gözlenmiştir. Şöyle ki; 4 işlik bir problemde genetik algoritma, yasaklı algoritmanın çözümünde %34'lük bir iyileştirme yapabilirken 10 işlik bir problemde %95'lik bir iyileştirme yapabildiği gözlenmiştir. Bunun sebebi, genetik algoritma yasaklı arama algoritmasına göre daha geniş bir alanda tarama yapabilmektedir. Problem küçük boyutlu iken, her iki algoritmanın taradığı alanlar birbirine yakın olacaktır. Problem boyutu büyüdükçe çözüm uzayı da büyür ve genetik algoritmanın etkinliği de artmaya başlar.
 - iii. Yasaklı arama algoritması parametrelerinden iterasyon sayısı ve tabu listesi uzunluğunun (hafızasının) ortalama sezgisel hata üzerindeki etkisi incelendiğinde belirli seviyelerdeki artışlara göre iterasyon sayısının etkili olduğu gözlenmiş ve grafikler incelendiğinde de iterasyon sayısının 10'dan 50'ye artırılması durumunda ortalama sezgisel hatanın bundan olumlu etkilendiği gözlenmiştir.
 - iv. Genetik algoritma parametrelerinden iterasyon sayısı, çaprazlama olasılığı ve mutasyon olasılığının ortalama sezgisel hata üzerindeki etkisi incelendiğinde belirli seviyelerdeki artışlara göre iterasyon sayısının etkili olduğu gözlenmiştir ve grafikler incelendiğinde iterasyon sayısının 100'den 300'e artırılması durumunda ortalama sezgisel hatanın bundan olumlu etkilendiği gözlenmiştir.
 - v. Yukarıda yapılan analizler sonucu, ortalama sezgisel hata üzerinde etkisi olmayan parametreler eski seviyelerinde bırakılıp, etkili parametreler olumlu olan düzeylere çekilerek, en büyük boyutlu test problemi üzerinde %95 güven düzeyinde, ortalama sezgisel hatayı %4,5 seviyesinde tutabilmek için 71 adet koşum yapılması gerektiği hesaplanmıştır.
 - vi. Örnek olay problem verileri üzerinde yapılan toplam 3300 koşumluk 40 deney sonucunda, elde edilen en küçük amaç fonksiyon değeri en iyi değer olarak kabul edildiği takdirde, her bir deney için en iyiden sapma, ortalama %0,9 olarak hesaplanmıştır.

- vii. Yapılan 3300 adet kořumda 2910,939 olan ama fonksiyonu deęerleri ortalamasının, kořumlar sonsuza gittięinde ise; bulunacaęı aralık %95 gvenle [2898,88;2922,994] olarak hesaplanmıřtır.

6.2. neriler

Bundan sonrakialıřmalarda, ele alınan probleminzmn daha iyileřtirmek iin bualıřmada var olan algoritma geliřtirilebilir ya da dięer meta-sezgisel yntemlerden de faydalanılabilir.

alıřmanın rnek olay incelemesinde ele alınan ceza katsayıları, birok faktre gre bir uzman tarafından daha nceki tecrbelere baęlı olarak belirlenmiřtir. Ceza katsayıları belirlenirken bulanık kmeler de kullanılarakzmn gerek hayata entegrasyonu artırılabilir.

Ele alınan problemde iřlerin tamamlanmadan yarıda kesilebileceęi kabulü yapılarakzm metodu geliřtirilebilir.

KAYNAKLAR

- [1] **Güner, E., Karaca, M. E.**, 2004. Tam Zamanında Üretim Sisteminde Tedarikçi İlişkileri Ve En İyi Parti Büyüklüğü Üzerine Bir Uygulama, *Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, **19(4)**, 443-454.
- [2] **Allahverdi, A., Gupta, Jatinder N. D., Aldowaisan, T.**, 1999. A Rewiev of Scheduling Research Involving Setup Considerations, *The International Journal of Management Science*, **27**, 219-239.
- [3] **Naderi, B., Fatemi Ghomi, S. M. T., Aminnayeri, M.**, 2010. A High Performing Metaheuristic For Job Shop Scheduling With Sequence Dependent Setup Times, *Applies Soft Computing*, **10**, 703-710.
- [4] **Liu, C.Y., Chang, S. C.**, 2000. Scheduling Flexible Flowshops with Sequence Dependent Setup Effects, *IEEE Transactions on Robotics and Automation*, Vol. **16**, no. 4.
- [5] **Naderi, B., Zandieh, M., Fatemi Ghomi, S. M. T.**, 2009. Scheduling Sequence-Dependent Setup Time Job Shops With Preventive Maintenance, *International Journal of Advanced Manufacturing Technology*, **43**, 170-181.
- [6] **Gavett, William J.**, 1965. Three Heuristics Rules For Sequencing Jobs To A Single Production Facility, *Management Science*.
- [7] **Lockett, A. G., Muhleman, A. P.**, 1971. A Scheduling Problem Involving Sequence Dependent Changeover Times.
- [8] **Choobineh, Fred F., Mohebbi, E., Khoo, H.**, 2006. A Multi-Objective Tabu Search For A Single Machine Scheduling Problem With Sequence Dependent Setup Times, *European Journal of Operational Research*.
- [9] **Moghaddas, R., Houshmond, M.**, 2008. Job-Shop Scheduling Problem With Sequence Dependent Setup Times, *Proceedings of The International Multiconference of Engineers and Computer Scientists*, **2**, 19-21.
- [10] **Eren, T., Güner, E.**, 2007. Sıra Bağımlı Hazırlık Zamanlı İki Ölçütlü Çizelgeleme Problemi: Toplam Tamamlanma Zamanı Ve Maksimum Erken Bitirme, *Erciyes Üniversitesi FBE Dergisi*, **23(1-2)**, 95-105.
- [11] **Bruno, J., Sethi, R.**, 1978. Task Sequencing in A Batch Environment With Setup Times, *Found. Control Eng.*
- [12] **Tang, Christopher S.**, 1990. Scheduling Batches on Parallel Machines With Major and Minor Setups, *European Journal of Operational Research*, **4(2)**, 329-341.
- [13] **Cheng, T.C.E., Chen, Z.L.**, 1993. Parallel Machine Scheduling with Batch Setup Times, *Operations Research*, **42(6)**, 1171-1174.

- [14] **Schutten, J.M.J., Leussink, R.A.M.**, 1996. Parallel Machine Scheduling with Release Dates, Due Dates and Family Setup Times, *International Journal of Production Economics*, **46-47**, 119-125.
- [15] **Glassey, C.R.**, 1967. Minimum Changeover Scheduling of Several Products on One Machine, *Operations Research*.
- [16] **Schwimer, J.**, 1972. On The N-Job, One Machine, Sequence-Dependent Scheduling Problem with Tardiness Penalties: A Branch and Bound Solution, *Management Science*, **18(6)**.
- [17] **Kan, Rinnooy A.G., Lageweg, B.J., Lenstra, J.K.**, 1975. Minimizing Total Costs in One-Machine Scheduling, *Operations Research*, **23(5)**, 908-927.
- [18] **Baker, Kenneth R.**, 1999. Heuristic Procedures for Scheduling Job Families with Setups and Due Dates, *Naval Research Logistics*, **46**, 978-991.
- [19] **Pan, Jason C.-H., Chen, J.-S., Cheng, H.-L.**, 2001. A Heuristic Approach for Single Machine Scheduling With Due Dates and Class Setups, *Computers & Operations Research*, **28 (2001)**, 1111-1130.
- [20] **Sourd, F., Kedad-Sidhoum, S.**, 2003. The One-Machine Problem with Earliness and Tardiness Penalties, *Journal of Scheduling*, **6**, 533-549.
- [21] **Nazif, H., Lee, L.S.**, 2009. A Genetic Algorithm on Single Machine Scheduling Problem to Minimise Total Weighted Completion Time, *European Journal of Scientific Research*, **35(3)**, 444-452.
- [22] **Johnson, S.M.**, 1953. Optimal Two and Three Stage Production Schedule with Setup Times Included.
- [23] **Allahverdi, A.**, 1997. Scheduling in Stochastic Flowshops with Independent Setup, Processing and Removal Times, *Computers & Operations Research*, **24(10)**, 955-960.
- [24] **Sule, Dileep R., Huang, Kin Y.**, 1983. Sequency on Two and Three Machines with Setup Processing and Removal Times Separated, *International Journal of Production Research*, **21(5)**, 723-732.
- [25] **Ben-Daya, M., Al-Fawzan, M.**, 1998. A Tabu Search Approach for The Flowshop Scheduling Problem, *European Journal of Operational Research*, **109**, 88-95.
- [26] **Taillard, E.**, 1990. Some Efficient Heuristic Methods for The Flowshop Sequencing Problem, *European Journal of Operational Research*, **47**, 65-74.
- [27] **Obgu, F.A., Smith, D.K.**, 1990. The Application of The Simulated Annealing Algorithm to The Solution of The $n/m/C_{\max}$ Flowshop Problem, *Computers & Operations Research*, **17(3)**, 243-253.
- [28] **Allahverdi, Ali**, 2000. Minimizing Mean Flowtime in a Two Machine Flowshop with Sequence Independent Setup Times, *Computers & Operations Research*, **27**, 111-127.
- [29] **Gupta, J.N.D., Schaller J.E.**, 2006. Minimizing Flowtime in a Flow-Line Manufacturing Cell with Family Setup Times, *Journal of Operational Research society*, **57(2)**, 163-176.

- [30] **Yenisey, Mutlu M., Yağmahan, B.,** 2006. Akış Tipi Çizelgeleme Problemi İçin KKE Parametre Eniyileme, *İTÜ Dergisi/d Mühendislik*, **5:2,2**, 133-141.
- [31] **Yenisey, Mutlu M., Yağmahan, B.,** 2010. A Multi-objective Ant Colony System Algorithm For Flowshop Scheduling Problem, *Expert System with Applications*, **31**, 1361- 1368.
- [32] **Laarhoven, Peter J.M., Aarts, Emile H.L., Lenstra, Jan K.,** 1992. Job Shop scheduling by Simulated Annealing, *Operations Research*, **40(1)**, 113-125.
- [33] **Colorni, A., Dorigo, M., Maniezzo, V., Trubian, M.,** 1994. Ant System for Job-Shop Scheduling, *Belgian Journal of Operations Research, Statistic and Computer Science*, **34(1)**, 39-53.
- [34] **Sotskov, Yuri N., Tautenhahn, T., Werner, F.,** 1999. On The Application of Insertion Techniques for Job Shop Problems with Setup Times, **33(2)**, 209-245.
- [35] **Gonçalves, J., Mendes, S., Resende, M.,** 2002. A Hybrid Genetic Algorithm for Jobshop Scheduling, *AT&T Labs Research Technical Report*.
- [36] **Hemamalini, T., Senthilvel, A.N., Somasunsadaram, S.,** 2010. Deep Memory Greedy Search and Allocation of Job Shop Scheduling for Optimizing Shop Floor Performance, *Asian Journal of Industrial Engineering*, **2**, 9-16.
- [37] **Deane, Richard H., White, Emmett R.,** 1975. Balancing Workloads and Minimizing Set-up Costs in the Parallel Processing Shop, *Operational Research Quarterly*, **26(1)**, 45-53
- [38] **Guinet, A.,** 1993. Scheduling Sequence-Dependent Jobs on Identical Parallel Machines to Minimize Completion Time Criteria, *International Journal of Production Research*, **31(7)**, 1579-1594.
- [39] **Lee, Young H., Pinedo, M.,** 1997. Scheduling Jobs on Parallel Machines with Sequence Dependent Setup Times, *European Journal of Operational Research*, **100**, 464-474.
- [40] **Eom, D.-H., Shin, H.-J., Kwun, I.-H., Shim, J.-K., Kim, S.-S.,** 2002. Scheduling Jobs on Parallel Machines with Sequence Dependent Family Setup Times, *The International Journal of Advanced Manufacturing Technology*, **19**, 926-232.
- [41] **Paula, Mateus R., Ravetti, Martin G., Mateus, Geraldo R., Pardalo, Panos M.,** 2007. Solving Parallel Machines Scheduling Problems with Sequence Dependent Setup Times Using Variable Neighborhood Search, *IMA Journal Management Mathematics*, **18(2)**, 101-115.
- [42] **Gavett, William J.,** 1965. Three Heuristic Rules For Sequencing Jobs to a Single Production Facility, *Management Science*, **11(8)**.
- [43] **Lockett, A.G., Muhlemann, A.P.,** 1971. A Scheduling Problem Involving Sequence Dependent Changeover Times, *Operations Research*, **20(4)**, 895-902.

- [44] **White, Charles H., Wilson, Richard C.,** 1977. Sequence Dependent Setup Times and Job Sequencing, *International Journal of Production Research*, **15(2)**, 191-202.
- [45] **Ragatz, Gary L.,** 1993. A Branch and Bound Method for Minimum Tardiness Sequencing on a Single Processor with Sequence Dependent Setup Times, *Proceedings of the 24th Annual Meeting of the Decision Science Institute*.
- [46] **França, P., Mendens A., Moscato, P.,** 2000. A Memetic Algorithm for the Total Tardiness Single Machine Scheduling Problem, *European Journal of Operational Research*, **132**, 224-242.
- [47] **Cheng, T.C.E., Ng, C.T., Yuan, J.J.,** 2003. The Single Machine Batching Problem with Family Setup Times to Minimize Maximum Lateness is Strongly NP-Hard, *Journal of Scheduling*, **6**, 483-490.
- [48] **Sourd, F.,** 2005. Earliness-Tardiness Scheduling with Setup Consideration, *Computers and Operations Research*, **32**, 1849-1865.
- [49] **Choobineh, Fred F., Mohebbi, E., Khoo, H.,** 2006. A Multiobjective Tabu Search for A Single Machine Scheduling Problem with Sequence-Dependent Setup Times, *European Journal of Operational Research*, **175**, 318-337.
- [50] **Gupta, Skylab L., Smith Jeffrey S.,** 2006. Algorithms for Single Machine Total Tardiness Scheduling with Sequence Dependent Setups, *European Journal of Operational Research*, **175**, 722-739.
- [51] **Lin, S.-W., Ying, K.-C.,** 2007. Solving Single-Machine Total Weighted Tardiness Problems with Sequence Dependent Setup Times by Metaheuristics, *International Journal of Advanced Manufacturing Technology*, **34(11-12)**, 1183-1190.
- [52] **Liao, C.-J., Juan, H.-C.,** 2007. An Ant Colony Optimization for Single Machine Tardiness Scheduling with Sequence-Dependent Setups, *Computers & Operations Research*, **34**, 1899-1909.
- [53] **Corwin, Burton D., Esogbue, Augustin O.,** 1974. Two Machine Flowshop Scheduling Problems with Sequence Dependent Setup Times: A Dynamic Programming Approach, *Naval Researchs Logistics Quarterly*, **21(3)**, 515-524.
- [54] **Gupta, J.N.D.,** 1986. Flowshop Scheduling with Sequence Dependent Setup Times, *Journal of Operations Research Society of Japan*, **29(3)**, 206-219.
- [55] **Gupta, J.N.D., DAS, S.R., Ghosh, S.,** 1995. Flowshop Scheduling with Sequence Dependent Setup Times, *Working Paper Department of Management*.
- [56] **Aldowaisan, T., Allahverdi, A.,** 1998. Total Flowtime in No-Wait Flowshops with Separated Setup Times, *Computers & Operations Research*, **25(9)**, 757-765.

- [57] **Ruiz, R., Maroto, C., Alcaraz, J.,** 2004. Solving the Flowshop Scheduling Problem with Sequence Dependent Setup Times Using Advanced Metaheuristics, *European Journal of Operational Research*, **165**, 34-54.
- [58] **Zandieh, M., Ghomi, Fatemi S.M.T., Hussein, Moattar S.M.,** 2006. An Immune Algorithm Approach to Hybrid Flow Shops Scheduling with Sequence Dependent Setup Times, *Applied Mathematics and Computation*, **180(1)**, 111-127.
- [59] **Wilbrecht, Jon K., Prescott, William B.,** 1969. The Influence Setup Time on Job Shop Performance, *Management Science*, **16(4)**.
- [60] **Srikar, Bellur N., Ghosh, S.,** 1986. A MILP Model for N-Job, M-Stage Flowshop with Sequence Dependent Set-Up Times, *International Journal of Production Research*, **24(6)**, 1459-1474.
- [61] **Kim, S.C., Bobrowski, P.M.,** 1994. Impact of Sequence-Dependent Setup Time on Job Shop Scheduling Performance, *International Journal of Production Research*, **32(7)**, 1503-1520.
- [62] **Low, Ching Y.,** 1995. Jobshop Scheduling Heuristics for Sequence Dependent Setups, *Computers & Industrial Engineering*, **29(1-4)**, 279-283.
- [63] **Cheung, W., Zhou, H.,** 2002. Using Genetic Algorithms and Heuristics for Job Shop Scheduling with Sequence-Dependent Setup Times, *Annals of Operations Research*, **107**, 65-81.
- [64] **Choi, I.-C., Choi, D.-S.,** 2002. A Local Search Algorithm for Jobshop Scheduling Problems with Alternative Operations and Sequence-Dependent Setups, *Computers & Industrial Engineering*, **42**, 43-58.
- [65] **Zhou, Y., Li, B., Yang, J.,** 2006. Study on Job Shop Scheduling with Sequence-Dependent Setup Times Using Biological Immune Algorithm, *International Journal of Advanced Manufacturing Technology*, **30**, 105-111.
- [66] **Naderi, B., Zandieh, M., Fatemi, Ghomi S.M.T.,** 2008. Scheduling Sequence-Dependent Setup Time Job Shops with Preventive Maintenance, *International Journal of Advanced Manufacturing Technology*, **43**, 170-181.
- [67] **Moghaddas, R., Houshmann, M.,** 2008. Job-Shop Scheduling Problem with Sequence Dependent Setup Times, *Proceeding of the International MultiConference of Engineers and Computer Scientists*.
- [68] **Naderi, B., Zandieh, M., Fatemi, Ghomi S.M.T.,** 2009. Scheduling Job Shop Problems with Sequence-Dependent Setup Times, *International Journal of Production Research*, **47(21)**, 5959-5976.
- [69] **Naderi, B., Ghomi, F., Aminnayeri, M.,** 2010. A High Performing Metaheuristic for Job Shop Scheduling with Sequence-Dependent Setup Times, *Applied Soft Computing*, **10**, 703-710.

- [70] **Akgül, Nergis F., Düzce, Çağrı M., Erdem, O., Kerimoğlu, A., Koçak, M., Karaoğlu, İ., Altıparmak, F.** Tusaş-Türk Havacılık ve Uzay Sanayi A.Ş.'de Paralel Makinalarda Çizelgeleme Problemi İçin Bir Çözüm Yaklaşımı, *MMO Endüstri Mühendisliği Dergisi*, **19(3)**, 35-47.
- [71] **Askin, Ronald G., Standridge, Charles R.**, 1993. Modeling And Analysis of Manufacturing Systems, *John Wiley and Sons Inc.*, p:19.
- [72] **Çalışkan, E., Acar, Hulusi H., Akay, Abdullah E.**, 2009. Odun Hammaddesi Taşımacılığında Metasezgisel Yöntemlerin Kullanımı, *Artvin Çoruh Üniversitesi Orman Fakültesi Dergisi*, **10(1)**, 19-28.
- [73] **Bredom, Alex V.**, 2001. Comparing Descent Heuristics and MetaHeuristics for Vehicle Routing Problem, *Computers and Operations Research*, **28(4)**, 289-315.
- [74] **Williems, T. M., Rooda, J. E.**, 1994. Neural Networks for Job-Shop Scheduling, *Control Engineering Practice*, **2(1)**, 31-39.
- [75] **Sabuncuoglu, İ., Gürgün, B.**, 1996. A Neural Network Model for Scheduling Problems, *European Journal of Operational Research*, **93(2)**, 288-299.
- [76] **Liang, R.-H., Hsu, Y.-Y.**, 2002. Scheduling of Hydroelectric Generation Using Artificial Neural Networks, *Generation, Transmission and Distribution IEEE Proceedings*, **141(5)**, 452-458.
- [77] **Lian, Z., Jiao, B., Gu, X.**, 2006. A Similar Particle Swarm Optimization Algorithm for Job-Shop Scheduling to Minimize Makespan, *Applied Mathematics and Computation*, **183**, 1008-1017.
- [78] **Liao, C.-J., Tseng, C.-T., Luarn, P.**, 2008. A Discrete Version of Particle Swarm Optimization for Flow-Shop Scheduling Problems, *Computers & Operations Research*, **34**, 3099-3111.
- [79] **Pan, Q.-K., Tasgetiren, M. F., Liang, Y.-C.**, 2008. A Discrete Particle Swarm Optimization Algorithm for the No-Wait Flow-Shop Scheduling Problem, *Computers & Operations Research*, **35**, 2807-2839.
- [80] **Yağmağan, B., Yenisey, M.**, 2008. Ant Colony Optimization for Multi-Objective Flow-Shop Scheduling Problem, *Computers & Industrial Engineering*, **54**, 411-420.
- [81] **Kirkpatrick, S., Gelatt, C.D., Vecchi, M. P.**, 1983. Optimization by Simulated Annealing, *Science*, **220**, 671-680.
- [82] **Temiz, İ.** 2010. Çok Kriterli Permütasyon Akış Tipi Çizelgeleme Problemi İçin Bir Tavlama Benzetimi Yaklaşımı, *Çankaya University Journal of Science and Engineering*, **7(2)**, 141-153.
- [83] **Gözüdeli, Y., Akçayol, Ali M.**, 2008. XML Veritabanı İçin Tavlama Benzetimi ile Sorgu Optimizasyonu, *Gazi Üniversitesi Bilişim Teknolojileri Dergisi*, **1(1)**, 13-22.

- [84] **Glover, F.** 1989. Tabu Search- Part I, *ORSA Journal on Computing*, **1(3)**.
- [85] **Glover, F.** 1990. Tabu Search- Part II, *ORSA Journal on Computing*, **2(1)**.
- [86] **Güden, H., Vakvak, B., Özkan, Barış E., Altıparmak, F., Dengiz, B.** Genel Amaçlı Arama Algoritmaları ile Benzetim Eniyilemesi: En iyi Kanban Sayısının Bulunması, *Makine Mühendisleri Odası Endüstri Mühendisliği Dergisi*, **16(1)**, 2-15.
- [87] **Özdemir, Müge H.**, 2007. Kısıtlandırılmış Teslim Tarihli ve Sıraya Bağımlı Hazırlık Süreli Tek Makine Çizelgeleme Problemlerinde Erkenlik ve Geçlik Toplamının Enküçüklenmesi, *Gazi Üniversitesi FBE Yüksek Lisans Tezi*, s:59.
- [88] **Candan, G.**, 2010. Esnek Üretim Çizelgeleme Problemlerinin Genetik Algoritma ve Bulanık Mantık Yöntemleri ile Çözümlemesi, *Sakarya Üniversitesi FBE Yüksek Lisans Tezi*, s:25,30.
- [89] **Emel, Gül G., Taşkın, Ç.**, 2002. Genetik Algoritmalar ve Uygulama Alanları, *Uludağ Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, **21(1)**, 129-152.
- [90] **Turğut, P., Gümüşçü, M., Arslan, A.**, 2002. Genetik Algoritmalar ve Çalışma Prensipleri, *GAP IV. Mühendislik Kongresi Bildiriler Kitabı*.
- [91] **Dorigo, M., Di Caro, G.**, 1999. Ant Colony Optimization: A New Meta-Heuristic, *Evolutionary Computation Proceedings of the 1999 Congress*.
- [92] **Kahnlı, A.**, 2003. Karınca Koloni Algoritması Kullanarak Aktif Filtre Tasarımı, *Elektrik Mühendisleri Odası Elektrik-Elektronik-Bilgisayar Mühendisliği 10. Ulusal Kongresi ve Fuarı Bildirileri*.
- [93] **Serin, S., Terzi, S.**, 2008. Ulaştırma Mühendisliğinde Karınca Kolonisinin Kullanılması, *Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu*.
- [94] **Duman, E., Akın, E.**, 2006. Yapay Bağışıklık Sistemi ile Bulanık Yaklaşımcı Kural Tabanlı Optimizasyonu, *Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu ELECO'06*.
- [95] **Aladağ, A.**, 2010. Tekrar İşlemeli Esnek Atölye Tipi Çizelgeleme Problemi İçin Yapay Bağışıklık Sistemi ile Bir Çözüm Yaklaşımı, *Eskişehir Osmangazi Üniversitesi FBE Yüksek Lisans Tezi*, s: 49.
- [96] **Kennedy, J., Eberhart, R.**, 1995. Particle Swarm Optimization, *Neural Networks IEEE International Conference Proceedings*, **4**, 1942-1948.
- [97] **Özsağlam, Mehmet Y.**, 2009. Parçacık Sürü Optimizasyonu Algoritmasının Gezgin Satıcı Problemine Uygulanması ve Performansının İncelenmesi, *Selçuk Üniversitesi FBE Yüksek Lisans Tezi*, s: 15.
- [98] **Ergezer, H., Dikmen, M., Özdemir, E.**, 2003. Yapay Sinir Ağları Ve Tanıma Sistemleri, *Pivolka*, **2(6)**, 14-17.
- [99] **Pinedo, M.**, 1995. Scheduling Theory, Algorithms, and Systems, *Prentice Hall*.

- [100] **Coleman, Jay B.**, 1992. Technical Note: A Simple Model for Optimizing The Single Machine Early/Tardy Problem with Sequence Dependent Setups, *Production and Operations Management*, **1(2)**.
- [101] **Rabadi, G.**, 1999. Minimizing The Total Earliness and Tardiness for a Single-Machine Scheduling Problem with Common Due Dates and Sequence-Dependent Setup Times, *Doktora Tezi, University of Central Florida*.
- [102] **Gözüdeli, Y., Akçayol, Ali M.**, 2007. Genetik Algoritma ile Web Sayfası Düzeninin Gerçek Zamanlı Optimizasyonu, *Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi*, **22(2)**, 431-439.

EKLER

EK A.1: Test Probleminin Verileri

Çizelge A.1: Test problemlerinin verileri.

Problem No	İş Sayısı	Teslim Süresi	İşlem Süresi	Ceza Katsayıları	Hazırlık Matrisi	En İyi Sıralama	En İyi Çözüm																																																	
1	4	257	104	(0,25-0,75)	<table><tr><td>Sij</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td>1</td><td>-</td><td>3</td><td>5</td><td>7</td></tr><tr><td>2</td><td>10</td><td>-</td><td>2</td><td>12</td></tr><tr><td>3</td><td>15</td><td>15</td><td>-</td><td>13</td></tr><tr><td>4</td><td>18</td><td>17</td><td>6</td><td>-</td></tr></table>	Sij	1	2	3	4	1	-	3	5	7	2	10	-	2	12	3	15	15	-	13	4	18	17	6	-	2-3-1-4	328,50																								
Sij		1	2	3		4																																																		
1		-	3	5		7																																																		
2		10	-	2		12																																																		
3	15	15	-	13																																																				
4	18	17	6	-																																																				
2	195	114	(0,50-0,50)	2-3-1-4	246																																																			
	220	124																																																						
	192	115																																																						
3	5	242	132	(0,25-0,75)	<table><tr><td>Sij</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td></tr><tr><td>1</td><td>-</td><td>7</td><td>8</td><td>6</td><td>4</td></tr><tr><td>2</td><td>6</td><td>-</td><td>14</td><td>4</td><td>5</td></tr><tr><td>3</td><td>12</td><td>6</td><td>-</td><td>4</td><td>3</td></tr><tr><td>4</td><td>12</td><td>12</td><td>14</td><td>-</td><td>3</td></tr><tr><td>5</td><td>13</td><td>16</td><td>18</td><td>4</td><td>-</td></tr></table>	Sij	1	2	3	4	5	1	-	7	8	6	4	2	6	-	14	4	5	3	12	6	-	4	3	4	12	12	14	-	3	5	13	16	18	4	-	3-2-5-4-1	334													
Sij		1	2	3		4	5																																																	
1		-	7	8		6	4																																																	
2		6	-	14		4	5																																																	
3	12	6	-	4	3																																																			
4	12	12	14	-	3																																																			
5	13	16	18	4	-																																																			
4	229	82	(0,50-0,50)	3-2-5-4-1	269																																																			
	229	118																																																						
	309	101																																																						
		303	87																																																					
5	6	334	117	(0,25-0,75)	<table><tr><td>Sij</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr><tr><td>1</td><td>-</td><td>12</td><td>13</td><td>12</td><td>9</td><td>8</td></tr><tr><td>2</td><td>10</td><td>-</td><td>13</td><td>8</td><td>9</td><td>7</td></tr><tr><td>3</td><td>12</td><td>19</td><td>-</td><td>9</td><td>9</td><td>9</td></tr><tr><td>4</td><td>8</td><td>11</td><td>18</td><td>-</td><td>14</td><td>10</td></tr><tr><td>5</td><td>3</td><td>13</td><td>4</td><td>3</td><td>-</td><td>0</td></tr><tr><td>6</td><td>3</td><td>15</td><td>8</td><td>5</td><td>4</td><td>-</td></tr></table>	Sij	1	2	3	4	5	6	1	-	12	13	12	9	8	2	10	-	13	8	9	7	3	12	19	-	9	9	9	4	8	11	18	-	14	10	5	3	13	4	3	-	0	6	3	15	8	5	4	-	3-6-4-2-5-1	517,75
Sij		1	2	3		4	5	6																																																
1		-	12	13		12	9	8																																																
2		10	-	13		8	9	7																																																
3	12	19	-	9	9	9																																																		
4	8	11	18	-	14	10																																																		
5	3	13	4	3	-	0																																																		
6	3	15	8	5	4	-																																																		
6	339	101	(0,50-0,50)	1-3-6-4-2-5	450,50																																																			
	291	95																																																						
	334	100																																																						
	368	118																																																						
		315	93																																																					

Çizelge A.1: Test problemlerinin verileri (devamı).

Problem No	İş Sayısı	Teslim Süresi	İşlem Süresi	Ceza Katsayıları	Hazırlık Matrisi								En İyi Sıralama	En İyi Çözüm	
7	7	299	98	(0,25-0,75)	Sij	1	2	3	4	5	6	7	1-2-6-4-5-7-3	640,25	
		1	-		7	3	7	7	7	11					
		2	4		-	1	0	0	0	4					
		3	14		10	-	10	6	10	10					
8		336	75	(0,50-0,50)	4	4	7	1	-	3	1	7	1-2-6-4-5-7-3	534,25	
		294	82		5	8	4	5	4	-	4	4			
		332	105		6	4	7	1	0	3	-	7			
					7	4	0	1	0	0	0	-			
9	8	441	88	(0,25-0,75)	Sij	1	2	3	4	5	6	7	8	8-5-7-2-3-6-1-4	699,50
		383	61		1	-	11	11	5	2	6	7	10		
		395	75		2	4	-	11	3	1	6	5	4		
		422	123		3	8	13	-	9	9	7	8	4		
10		351	79	(0,50-0,50)	4	1	7	12	-	3	7	2	11	8-7-5-2-3-6-1-4	590,50
		409	79		5	4	10	12	3	-	10	5	8		
		366	110		6	1	6	12	6	3	-	7	10		
		326	123		7	1	5	10	6	3	7	-	9		
				8	6	12	12	5	6	12	7	-			

Çizelge A.1: Test problemlerinin verileri (devamı).

Problem No	İş Sayısı	Teslim Süresi	İşlem Süresi	Ceza Katsayıları	Hazırlık Matrisi	En İyi Sıralama	En İyi Çözüm																																																																		
11	10	561	136	(0,25-0,75)	<table><tr><th>Sij</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th></tr><tr><td>1</td><td>-</td><td>4</td><td>3</td><td>6</td><td>6</td><td>0</td><td>4</td><td>0</td><td>9</td><td>1</td></tr><tr><td>2</td><td>9</td><td>-</td><td>8</td><td>6</td><td>2</td><td>3</td><td>3</td><td>9</td><td>8</td><td>6</td></tr><tr><td>3</td><td>5</td><td>9</td><td>-</td><td>11</td><td>11</td><td>5</td><td>9</td><td>5</td><td>14</td><td>6</td></tr><tr><td>4</td><td>5</td><td>9</td><td>8</td><td>-</td><td>11</td><td>5</td><td>6</td><td>5</td><td>11</td><td>6</td></tr><tr><td>5</td><td>7</td><td>8</td><td>6</td><td>12</td><td>-</td><td>7</td><td>1</td><td>7</td><td>6</td><td>4</td></tr></table>	Sij	1	2	3	4	5	6	7	8	9	10	1	-	4	3	6	6	0	4	0	9	1	2	9	-	8	6	2	3	3	9	8	6	3	5	9	-	11	11	5	9	5	14	6	4	5	9	8	-	11	5	6	5	11	6	5	7	8	6	12	-	7	1	7	6	4	5-7-8-6-4-10-6-9-1-3	984,25
Sij		1	2		3	4	5	6	7	8	9	10																																																													
1		-	4		3	6	6	0	4	0	9	1																																																													
2		9	-		8	6	2	3	3	9	8	6																																																													
3		5	9		-	11	11	5	9	5	14	6																																																													
4		5	9		8	-	11	5	6	5	11	6																																																													
5		7	8	6	12	-	7	1	7	6	4																																																														
		510	98		<table><tr><td>6</td><td>10</td><td>4</td><td>5</td><td>9</td><td>6</td><td>-</td><td>7</td><td>9</td><td>10</td><td>10</td></tr><tr><td>7</td><td>6</td><td>8</td><td>5</td><td>11</td><td>9</td><td>6</td><td>-</td><td>6</td><td>5</td><td>3</td></tr><tr><td>8</td><td>1</td><td>4</td><td>3</td><td>6</td><td>6</td><td>0</td><td>4</td><td>-</td><td>9</td><td>1</td></tr><tr><td>9</td><td>10</td><td>9</td><td>12</td><td>15</td><td>11</td><td>9</td><td>12</td><td>9</td><td>-</td><td>10</td></tr><tr><td>10</td><td>5</td><td>6</td><td>2</td><td>8</td><td>8</td><td>4</td><td>3</td><td>4</td><td>8</td><td>-</td></tr></table>	6	10	4	5	9	6	-	7	9	10	10	7	6	8	5	11	9	6	-	6	5	3	8	1	4	3	6	6	0	4	-	9	1	9	10	9	12	15	11	9	12	9	-	10	10	5	6	2	8	8	4	3	4	8	-	8-5-7-2-4-10-6-9-1-3	913,5											
6		10	4	5	9	6	-	7	9	10	10																																																														
7		6	8	5	11	9	6	-	6	5	3																																																														
8	1	4	3	6	6	0	4	-	9	1																																																															
9	10	9	12	15	11	9	12	9	-	10																																																															
10	5	6	2	8	8	4	3	4	8	-																																																															
	570	168																																																																							
	489	57																																																																							
	516	105																																																																							
	576	73		(0,50-0,50)																																																																					
	410	104																																																																							
	414	113																																																																							
	590	113																																																																							
12		549	51																																																																						

ÖZGEÇMİŞ

Ad Soyad: Sena KIR

Doğum Yeri ve Tarihi: Üsküdar /08.12.1987

Adres: Korucuk Mahallesi 1697 Kümeevler No: 9/8 Adapazarı/SAKARYA

Lisans Üniversite: Sakarya Üniversitesi (2005-2009)

Yayın Listesi:

- Yalçiner Yılmaz, A., Özçelik Över, T., **Kır, S.**, “E-signature and Time Stamping Concepts and Services Evolution in Turkey”, *2010 IMS Symposium, Bosnia Herzogovina*.