

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

**SIRA BAĞIMLI HAZIRLIK SÜRELERİNİN SİPARİŞ TİPİ
ÜRETİM ÇİZELGESİ PERFORMANSI ÜZERİNDEKİ ETKİSİ**

**Hazırlayan
Orhun KARACAN**

**Danışman
Doç. Dr. İbrahim DOĞAN**

Yüksek Lisans Tezi

**Haziran 2019
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

**SIRA BAĞIMLI HAZIRLIK SÜRELERİNİN SİPARİŞ TİPİ
ÜRETİM ÇİZELGESİ PERFORMANSI ÜZERİNDEKİ ETKİSİ
(Yüksek Lisans Tezi)**

**Hazırlayan
Orhun KARACAN**

**Danışman
Doç. Dr. İbrahim DOĞAN**

**Haziran 2019
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.

Adı-Soyadı: Orhun KARACAN

İmza:



YÖNERGEYE UYGUNLUK

“Sıra Bağımlı Hazırlık Sürelerinin Sipariş Tipi Üretim Çizelgesi Performansı Üzerindeki Etkisi” adlı Yüksek Lisans tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi ’ne uygun olarak hazırlanmıştır.

Tezi Hazırlayan

Orhun KARACAN

Danışman

Doç. Dr. İbrahim DOĞAN

Endüstri Mühendisliği ABD Başkanı

Prof. Dr. Mithat ZEYDAN

Doç. Dr. İbrahim DOĞAN danışmanlığında **Orhun KARACAN** tarafından hazırlanan “**Sıra Bağımlı Hazırlık Sürelerinin Sipariş Tipi Üretim Çizelgesi Performansı Üzerindeki Etkisi**” adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü **Endüstri Mühendisliği** Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

27 / 06 / 2019

JÜRİ:

Danışman : Doç. Dr. İbrahim DOĞAN

Üye : Doç. Dr. Ercan ŞENYİĞİT

Üye : Doç. Dr. Neslihan DEMİREL

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun 04/07/2019 tarih ve 2019/38-06 sayılı kararı ile onaylanmıştır.



04 / 07 / 2019

Prof. Dr. Mehmet AKKURT

Enstitü Müdürü

ÖNSÖZ / TEŞEKKÜR

Tez çalışmamda değerli görüşleri ile bana yol gösteren tez danışmanım Doç. Dr. İbrahim DOĞAN' a teşekkürü bir borç bilirim.

Bu süreçte ve hayatım boyunca bana her konuda destek olan sevgili aileme ve özellikle eşim Tülay KARACAN' a teşekkür eder, sevgilerimi sunarım.

Orhun KARACAN
Kayseri, Haziran 2019

SIRA BAĞIMLI HAZIRLIK SÜRELERİNİN SİPARİŞ TİPİ ÜRETİM ÇİZELGESİ PERFORMANSI ÜZERİNDEKİ ETKİSİ

Orhun KARACAN

Erciyes Üniversitesi, Fen Bilimleri Enstitüsü

Yüksek Lisans Tezi, Haziran 2019

Danışman: Doç. Dr. İbrahim DOĞAN

ÖZET

Üretim sistemleri çok fazla değişkeni ve talebi olan karmaşık yapılardır. Her sistemin kendine has yapısı vardır ve problemlere bu yapılara uygun çözüm önerileri geliştirmek gerekir. Bu tez çalışmasında sıra bağımlı hazırlık sürelerinin tek makine çizelgeleme modeli üzerindeki etkisi incelenmiş ve çözüm önerileri getirilmiştir. Sıra bağımlı hazırlık süreleri çizelge performansını etkileyen ve verimliliği artıran en önemli unsurdur. Ele alınan problem için matematiksel programlama modeli geliştirilmiş ve küçük ölçekli problemler için en uygun çözümler bulunmuştur. Büyük ölçekli problemleri çözmek için genetik algoritma kullanılmıştır ve genetik algoritmaya başlangıç popülasyonları üretmek için dört adet sezgisel algoritma geliştirilmiştir. Genetik algoritma ve geliştirilen sezgisel algoritmalar, 360 adet farklı senaryo ve veri setinde test edilerek karşılaştırmaları yapılmıştır. Devamında ise bir firmada üretilen bir çizelge üzerinde denemeler yapılmıştır. Farklı iterasyon setlerinde genetik algoritma çalıştırılmıştır ve firmadan alınan çizelge performansı ile karşılaştırılmıştır.

Anahtar Kelimeler: Tek Makine Çizelgeleme, Sıra Bağımlı Hazırlık Süreleri, Genetik Algoritma, NP-Zor, Toplam Ağırlıklı Erken Bitirme, Toplam Ağırlıklı Gecikme, Sezgisel Algoritmalar.

EFFECT OF SEQUENCE – DEPENDENT SETUP TIMES ON JOB SHOP PRODUCTION SCHEDULING PERFORMANCE

Orhun KARACAN

Erciyes University, Graduate School of Natural and Applied Sciences

M.Sc. Thesis, June 2019

Supervisor: Assoc. Prof. Dr. İbrahim DOĞAN

ABSTRACT

Production systems are complex structures with many variables and demands. Each system has its own structure and it is necessary to develop solutions to problems in accordance with these structures. In this thesis, single machine scheduling problem with sequence-dependent setup times has been examined and solution suggestions have been made. Sequence-dependent setup times are one of the important factors that affect schedule performance and increase the productivity. In this study, mathematical programming model has been developed for the problem and the optimum solutions have been found for small scale problems. Genetic algorithms have been used to solve large scale problems and four heuristic algorithms are developed to generate initial populations for the genetic algorithm. Genetic algorithm and heuristic algorithms are tested and compared under 360 different scenarios. Additionally, real data is taken from company and tested with improved models. Furthermore, genetic algorithms have been run for different iterations and compared with the real schedule performance.

Key Words: Single Machine Scheduling, Sequence-Dependent Setup Times, Genetic Algorithm, NP-Hard, Total Weighted Earliness, Total Weighted Tardiness, Heuristic Algorithm.

İÇİNDEKİLER

SIRA BAĞIMLI HAZIRLIK SÜRELERİNİN SİPARİŞ TİPİ ÜRETİM ÇİZELGESİ PERFORMANSI ÜZERİNDEKİ ETKİSİ

	<u>Sayfa</u>
BİLİMSEL ETİĞE UYGUNLUK	i
YÖNERGEYE UYGUNLUK.....	ii
KABUL VE ONAY SAYFASI	iii
ÖNSÖZ / TEŞEKKÜR	iv
ÖZET	v
ABSTRACT.....	vi
İÇİNDEKİLER	vii
KISALTMALAR	xi
TABLolar LİSTESİ.....	xii
ŞEKİLLER LİSTESİ	xiii
GİRİŞ	1

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR TARAMASI

1.1 Çizelgeleme Problemlerinin Temel Özellikleri ve Performans Kriterleri ..6	
1.1.1 Makine Bilgileri (α).....	8
1.1.2 İşlem Karakteristikleri (β).....	11
1.1.3 Amaç Fonksiyonları (γ)	11

1.2	Çizelgeleme Modelleri	12
1.2.1	Dinamik Çizelgeleme	13
1.2.2	Statik Çizelgeleme	13
1.2.3	Deterministik Modeller.....	14
1.2.4	Stokastik Modeller	14
1.2.5	Tek Makine Modeli	14
1.2.6	Paralel Makine Modeli	14
1.3	Tek Makine Çizelgeleme Modelleri	15
1.3.1	$1 \parallel \sum T_j$ ve $1 \parallel \sum w_j T_j$ Problemi	15
1.3.2	Farklı Teslim Tarihli E/G Problemleri	17
1.3.3	Sıra Bağımlı Hazırlık Süresi İçeren Modeller	18
1.3.3.1	Sıra Bağımlı Hazırlık Süresi İçeren E/G Problemleri	20
1.4	Çözüm İçin Değerlendirme	21

2. BÖLÜM

GENETİK ALGORİTMA

2.1	Başlangıç Popülasyonu	24
2.2	Uygunluk Fonksiyonu	25
2.3	Yeni Nesil Seçme Mekanizması	25
2.4	Çaprazlama Yöntemleri	26
2.4.1	Tek Noktalı Çaprazlama	27
2.4.2	İki Noktalı Çaprazlama	27
2.4.3	OX – LOX – PMX Çaprazlama	27
2.4.4	Pozisyona Dayalı Çaprazlama	28
2.5	Mutasyon Yöntemleri	28
2.6	Algoritmayı Sonlandırma	29

3. BÖLÜM

TEL ÇEKME BÖLÜMÜNDEKİ ÇİZELGELEME PROBLEMİNİN OPTİMİZASYONU

3.1	Problemin Tanımı ve Formülasyonu	31
3.1.1	Firmadaki İşlerin Temel Yapısı.....	31
3.1.2	Problemin Analizi	34
3.2	Problem İçin Matematiksel Model	35
3.2.1	Matematiksel Modelde Örnek Uygulamalar	37
3.2.2	Mevcut Problem İçin Matematiksel Model Gösterimi	39
3.3	Problemde Kullanılacak Temel Veriler ve Gösterimleri	41
3.4	Problem İçin Genetik Algoritma.....	43
3.4.1	Genetik Algoritma ve Çözüm Metodolojisinin Adımları	43
3.4.2	Çaprazlama Operatörü	45
3.4.3	Mutasyon Operatörü	46
3.4.4	Başlangıç Popülasyonu İçin Sezgisel Yöntemler ve Adımları.....	46
3.4.4.1	Bir Numaralı Sezgisel Algoritma (S1)	47
3.4.4.2	İki Numaralı Sezgisel Algoritma (S2).....	50
3.4.4.3	Üç Numaralı Sezgisel Algoritma (S3)	53
3.4.4.4	Dört Numaralı Sezgisel Algoritma (S4).....	57
3.5	GA ve Sezgisel Algoritmaların Analizi	61
3.5.1	Sentetik Deneyler İçin Senaryolar	61
3.5.2	Veri Setlerinin Sezgisel Algoritmalar ile Analizi	62
3.5.3	Veri Setlerinin GA ile Analizi	65
3.6	Gerçek Veri Seti ile Çalışmalar	68
3.6.1	Deney Tasarımı	68
3.6.2	Örnek Veri Setinin Çeşitli İterasyon Sayılarında Çalıştırılması	74

4. BÖLÜM

TARTIŞMA - SONUÇLAR VE ÖNERİLER	76
KAYNAKLAR	78
ÖZGEÇMİŞ.....	83



KISALTMALAR

NP-Zor	Polinomal Olmayan Zor
TGP	Toplam Gecikme Problemi
TAGP	Toplam Ağırlıklandırılmış Gecikme Problemi
E/G	Erkenlik/Geçlik
GA	Genetik Algoritma
LPT	En Uzun İşlem Süresi (Longest Processing Time)
OX	Sıralı Çaprazlama (Order Crossover)
LOX	Doğrusal Sıralı Çaprazlama (Linear Order Crossover)
PMX	Paçalı Eşleştirilmiş Çaprazlama (Partially Mapped Crossover)
S1	Bir Numaralı Sezgisel Algoritma
S2	İki Numaralı Sezgisel Algoritma
S3	Üç Numaralı Sezgisel Algoritma
S4	Dört Numaralı Sezgisel Algoritma
DD	Düzgün Dağılım Verileri
ND1	Normal Dağılım 1 Verileri
ND2	Normal Dağılım 2 Verileri
ORT	Ortalama
STD	Standart Sapma
GA1	Bir Numaralı Sezgisel Algoritma İle Çalışan Genetik Algoritma
GA2	İki Numaralı Sezgisel Algoritma İle Çalışan Genetik Algoritma
GA3	Üç Numaralı Sezgisel Algoritma İle Çalışan Genetik Algoritma
GA4	Dört Numaralı Sezgisel Algoritma İle Çalışan Genetik Algoritma

TABLOLAR LİSTESİ

Tablo 1.1 Çizelgeleme problemlerinin sınıflandırılması (Doğan 2013: 4).....	7
Tablo 3.1 Sıra bağımlı hazırlık sürelerinin gösterimi (dakika).....	37
Tablo 3.2 Eldeki işler için üretim sürelerinin gösterimi (dakika).....	37
Tablo 3.3 Eldeki işler için başlangıç hazırlık sürelerinin gösterimi (dakika).....	37
Tablo 3.4 Sıra bağımlı hazırlık sürelerinin gösterimi (dakika).....	38
Tablo 3.5 Eldeki işler için üretim sürelerinin gösterimi (dakika).....	38
Tablo 3.6 Eldeki işler için başlangıç hazırlık sürelerinin gösterimi (dakika).....	39
Tablo 3.7 Bir numaralı temel veriler tablosu.....	41
Tablo 3.8 İki numaralı temel veriler tablosu.....	42
Tablo 3.9 Sıra bağımlı hazırlık süreleri (Saat).....	42
Tablo 3.10 Veri türetme tablosu.....	62
Tablo 3.11 Sezgisel algoritmalar ile veri analizi.....	63
Tablo 3.12 Sezgisel algoritmaların birbirleri ile ilişkisi.....	64
Tablo 3.13 GA ile veri analizi.....	66
Tablo 3.14 GA sonuçlarının birbirleri ile ilişkisi.....	67
Tablo 3.15 Deney tasarımı düzeneği.....	69
Tablo 3.16 Deney tasarımı sezgisel algoritmalar sonuçları.....	70
Tablo 3.17 Deney tasarımı GA sonuçları.....	72
Tablo 3.18 Gerçek verilerin GA’da çalıştırılması.....	75
Tablo 3.19 Gerçek verilerin mevcut çizelge performansı ile karşılaştırılması.....	75

ŞEKİLLER LİSTESİ

Şekil 1.1 Üretim sistemlerinde bilgi akış diyagramı (Pinedo, 2008: 5).....	6
Şekil 1.2 Özdeş paralel makineler yerleşimi (Ak, 2012: 6).	9
Şekil 1.3 Esnek akış tipi yerleşim (Ak, 2012: 7).....	10
Şekil 1.4 Esnek atölye tipi üretim ortamı (Ak, 2012: 8).	10
Şekil 1.5 Çizelgeleme modelleri (Dağ, 2012: 10).....	13
Şekil 1.6 Paralel makineli bir model (Ceran, 2006: 11).....	15
Şekil 1.7 Çizelgeleme problemlerinin hazırlık süresine göre sınıflandırılması.	19
Şekil 2.1 Genetik algoritma genel akış şeması (Bolat vd. 2004).	24
Şekil 2.2 Tek noktalı çaprazlama	27
Şekil 2.3 İki noktalı çaprazlama	27
Şekil 2.4 İki nokta çaprazlama yöntemi örnekleri (Sevaux ve Peres, 2003).....	28
Şekil 2.5 Pozisyona dayalı çaprazlama (Ak, 2012: 58).	28
Şekil 2.6 Rastgele seçilen iki genin yer değiştirmesi.....	29
Şekil 2.7 Araya yerleştirme yöntemi.....	29
Şekil 3.1 Çok telli çelik halat örneği.....	31
Şekil 3.2 Halat katmanlarını oluşturan toronun önden görünümü	31
Şekil 3.3 Farika içi iş akış şeması	32
Şekil 3.4 Örnek bir çok telli halat üretim iş emri.....	33
Şekil 3.5 Çaprazlama operatörü gösterimi	45
Şekil 3.6 Keyfi iki genin yer değiştirmesi yöntemi.....	46
Şekil 3.7 S1 için akış diyagramı.....	49
Şekil 3.8 S2 için akış diyagramı.....	52
Şekil 3.9 S3 için akış diyagramı.....	56
Şekil 3.10 S4 için akış diyagramı.....	60
Şekil 3.11 Sezgisel algoritmaların S1 algoritması ile oranlarının kutu grafiği	64
Şekil 3.12 Sezgisel modeller için çalışma zamanları	65
Şekil 3.13 GA'ların GA1 algoritması ile oranlarının kutu grafiği	67
Şekil 3.14 Genetik algoritmalar için çalışma zamanları	68
Şekil 3.15 Sezgisel algoritmaların deney düzeylerinde analizi.....	71
Şekil 3.16 GA sonuçlarının RGA sonucu ile karşılaştırılması.....	73

Şekil 3.17 Sezgisel algoritmalar ve deney düzeylerinin GA'da analizi.....74



GİRİŞ

Bilim ve teknolojideki hızlı gelişmeler, sınırların ve gümrük duvarlarının kalkmasıyla yoğunlaşan rekabet, sermayenin ve iş gücünün eskiye göre daha kolay yer değiştirebilmesi firmaların kâr edebilmelerini ve hayatta kalabilmelerini oldukça zorlaştırmaktadır. Üretim firmalarının bu yoğun rekabet ortamında hayatta kalabilmeleri için iyi bir ürün geliştirme ve pazarlamanın yanı sıra, müşteri taleplerini zamanında karşılayan, malzeme, işçilik ve zaman israfını ortadan kaldıran, değişken taleplere hızlı şekilde karşılık verebilen iyi bir üretim çizelgeleme sistemine ihtiyaçları vardır.

Çizelgeleme imalat ve servis sistemlerinde karar verme süreçlerinde kullanılan çok önemli bir prosestir. Bir firmada çizelgeleme kaynakların verilen zaman aralıklarını da göz önünde bulundurarak görevlere tahsisi ile ilgilenir ve bir veya daha fazla amacı optimize etmeye çalışır (Pinedo, 2008: 1).

Bir başka tanıma göre çizelgeleme; makine, iş gücü, hammadde, yarı mamul, enerji ve zamana gibi kaynakların, matematiksel modeller ve sezgisel yöntemler kullanılarak etkin şekilde planlanmasıdır (Muştu, 2013: 1; Baker ve Trietsch, 2009: 3, 6).

Çizelgeleme modellerinin müşteri taleplerini zamanında karşılarken üstesinden gelmesi gereken bazı kısıtlar ve zorluklar vardır. Her üretim sisteminin kendine has problemleri vardır ve planlamacıların çizelgelerini oluştururken üstesinden gelmeleri gereken problemlerin bazılarında aşağıda bahsedilmiştir.

İş istasyonlarının aşırı yüklenmesi; sürekli değişen ve çeşitlenen ürün gamı üretim hatlarında aynı anda birçok ürünün üretilmesini zorunlu kılmıştır. Bu durum ise iş istasyonlarında üretilecek ürünlerin üretim zamanlarının değişik olmasına ve üst üste binen zaman birikmeleri ile iş istasyonlarında beklemelere ve yığılmalara sebep olmaktadır.

Zamanında üretim sıkıntısı; artık müşteri taleplerindeki çeşitlilikle birlikte müşteri taleplerinin zamanında karşılanması da iyice önem kazanmıştır. Birçok ürünün tek bir sistem üzerinde planlanması bazı gecikmeleri kaçınılmaz kılmaktadır.

Hazırlık süreleri ile oluşan kayıp zamanlar; sürekli değişen kalıplar iş istasyonlarında zaman kayıplarına sebep olmaktadır.

Hurda ve fire oranlarında artışlar; çok sık ürün değişimi beraberinde hurda ve fire miktarı artışını da getirmektedir.

Hammadde ve yarı mamul stok düzeyleri; çoklu üretim sistemlerinde en önemli sorunlardan biri de, hammadde ve yarı mamul stoklarının minimize edilmesidir. Birçok ürünün aynı anda üretilecek olması sistemde bir belirsizliğe sebep olmakta ve bu yüzden sistem kendini güvenceye alabilmek için hammadde ve yarı mamul stoklarını artırmak istemektedir.

Tüm bu süreçler ve problemler endüstriyel tesislerde üretim çizelgeleme yönteminin önemini ortaya çıkarmıştır. Çizelgeleme problemlerinde işler belirli bir zaman aralığında, eldeki kısıtlar göz önünde bulundurularak, bir veya daha fazla amaç fonksiyonuna göre sıralamaya tabi tutulur. Çizelgelemenin zorlukları kaynakların birden fazla iş için kullanılmasıyla ortaya çıkar. Bu durum için fayda maliyet hesapları yapılarak planlamacılar tarafından bir karar alma süreci oluşturulması gerekir. Doğru bir çizelgeleme; kaynakların verimli kullanılmasını ve maliyetlerde aşağı yönlü bir gelişme yaşanmasını sağlar.

Bu tezde sıra bağımlı hazırlık sürelerinin sipariş tipi üretim çizelgesi performansı üzerindeki etkisi incelenmiştir. Tezin ana motivasyon kaynağı, çelik halat imalatı yapan bir firmanın tel çekme bölümünde üretim çizelgeleme yaparak iç müşterilerin taleplerini zamanında karşılamaktır. Firmada hali hazırda basit bir çizelgeleme mantığı kullanılmaktadır, fakat bu çizelgelemeyi oluşturmak zaman almaktadır ve çok miktarda hata barındırabilmektedir. Firmadaki tel çekme bölümü, yaptığı üretimlerle yine aynı firma içerisindeki başka makineleri besleyen bir darboğaz alanıdır. Bu darboğaza çözüm üretebilmek için tek makine çizelgeleme yöntemi kullanılacak, başlangıçta teslim süreleri ve miktarları belli olan işler hazırlık süreleri de göz önünde bulundurularak çizelgelemeye tabi tutulacaktır. Çalışmanın amacı, tanımlanan problem için toplam gecikmeyi, toplam erken bitirmeyi ve toplama hazırlık süresini ağırlıkları oranında değerlendirerek minimize etmektir. Ele alınan problemde hazırlık süresinin vurgulanmasının sebebi, iyi yapılmamış bir planlamada bu sürelerin çok yüksek miktarlara ulaşması, zaman, emek, enerji ve fırsat maliyetlerine sebep olmasıdır.

Tezin birinci bölümü genel bilgiler ve literatür çalışmasına ayrılacaktır. Burada çizelgeleme türleri ve genel karakteristiklerinden bahsedilecek, bu çizelgeleme türlerine getirilen çözüm yöntemleri incelenecektir. Çizelgeleme literatürü; belirgin parametrelili türlerden (deterministik) değişken parametrelili türlere (stokastik), makinelerin tek olduğu sistemlerden çok olduğu sistemlere, işlerin geliş sürecinin durağan olduğu sistemlerden (statik) dinamik sistemlere varan türden problemler ihtiva eder (Eren ve Güner, 2002). Literatür çalışmasının son bölümünde erkenlik/geçlik (E/G) problemleri incelenecektir ve bu problemler sıra bağımlı hazırlık süreleri ile birlikte değerlendirilecektir.

Tezin ikinci bölümünde literatürde geçen, genel olarak kabul gören ve matematiksel modellerin çözüm üretilmediği problemlerde iyi sonuçlar veren genetik algoritmalarından bahsedilecektir. Burada kromozom yapıları, çaprazlama yöntemleri ve mutasyonlarla ilgili bilgiler verilecektir.

Üçüncü bölümde ise tezin ana motivasyon kaynağını oluşturan gerçek problem için çözüm yöntemleri geliştirilerek sonuçlara ulaşılmaya çalışılacaktır. İlk olarak probleme optimum çözüm üretebilmek için matematiksel model kullanılacaktır. Problemin yapısı NP-zor olduğu için ve matematiksel model buna makul sürelerde çözüm üretemeyeceği için genetik algoritmadan yararlanılacaktır. Genetik algoritmanın performansını artırmak amacıyla ona başlangıç çözümleri üretebilecek, farklı yapılarda ve özelliklerde dört adet sezgisel yöntem geliştirilecektir. Sezgisel algoritmalarından beklenen, rastgele üretilen başlangıç popülasyonlarını kullanan GA'ya göre daha düşük iterasyon sayılarında, daha istikrarlı ve uygunluk değeri yüksek sonuçlar elde etmektir. Sonrasında bu modeller birbiri ile ve rastgele üretilen başlangıç popülasyonları ile belirlenen performans kriterlerine göre kıyaslanacaktır. En nihayetinde genetik algoritma ile geliştirilen sezgisel algoritmaların bir kombinasyonu değerlendirilecektir. Firmadan alınan gerçek veriler için çözümler oluşturulacaktır ve bu çözümler ile firmada hali hazırda uygulanan çizelgeleme mantığı karşılaştırılacaktır.

Son olarak dördüncü bölümde ise çalışmanın sonuçları değerlendirilecek ve yeni öneriler getirilecektir.

1. BÖLÜM

GENEL BİLGİLER ve LİTERATÜR TARAMASI

Çizelgeleme; üretim, hizmet, inşaat gibi faaliyet alanlarındaki işlerin belirli bir sırada ve zaman çizelgesinde gerçekleşmesi için eldeki verilerin kullanılması ve çeşitli performans kriterlerini yerine getirerek program yapılmasıdır. İşletmelerde yapılması gereken işler çok fazladır fakat bu işleri gerçekleştirmek için elde bulunan kaynaklar kısıtlıdır. Çizelgelemede bu kısıtlı kaynaklara dayanarak işlem yapma zorunluluğu vardır.

Etkin ve başarılı bir çizelgeleme yapabilmek için üretim kaynaklarının yapısını, kabiliyetlerini ve sınırlarını iyi belirlemek gerekir (Baker ve Trietsch, 2009: 3). Ayrıca işletmede bulunan işlerin tam olarak neyi istediği de iyi belirlenmelidir. İşlerin sisteme giriş zamanı, sistemden en son çıkması gereken zamanları, gecikmelerin oluşturacağı maliyetler, hazırlık süreleri, hazırlık sürelerinin sıralamaya bağlı olup olmadığı gibi kriterlerin iyi teşhis edilmesi ve buna göre amaç fonksiyonlarını oluşturmak gerekecektir. İdeal bir amaç fonksiyonu kurabilmek için çizelgelemenin oluşturabileceği tüm maliyetlerin göz önünde bulundurulması gerekirken, gerçek hayat problemlerinde tüm maliyet unsurlarının tanımlanması ve amaç fonksiyonlarına entegre edilmesi mümkün değildir. Bundan dolayı önem düzeyi yüksek olan ve kolay belirlenebilen maliyet unsurları amaç fonksiyonlarına dahil edilirken, tali maliyet unsurları ise sabit kabul edilmekte veya göz ardı edilmektedir (Baker ve Trietsch, 2009: 4).

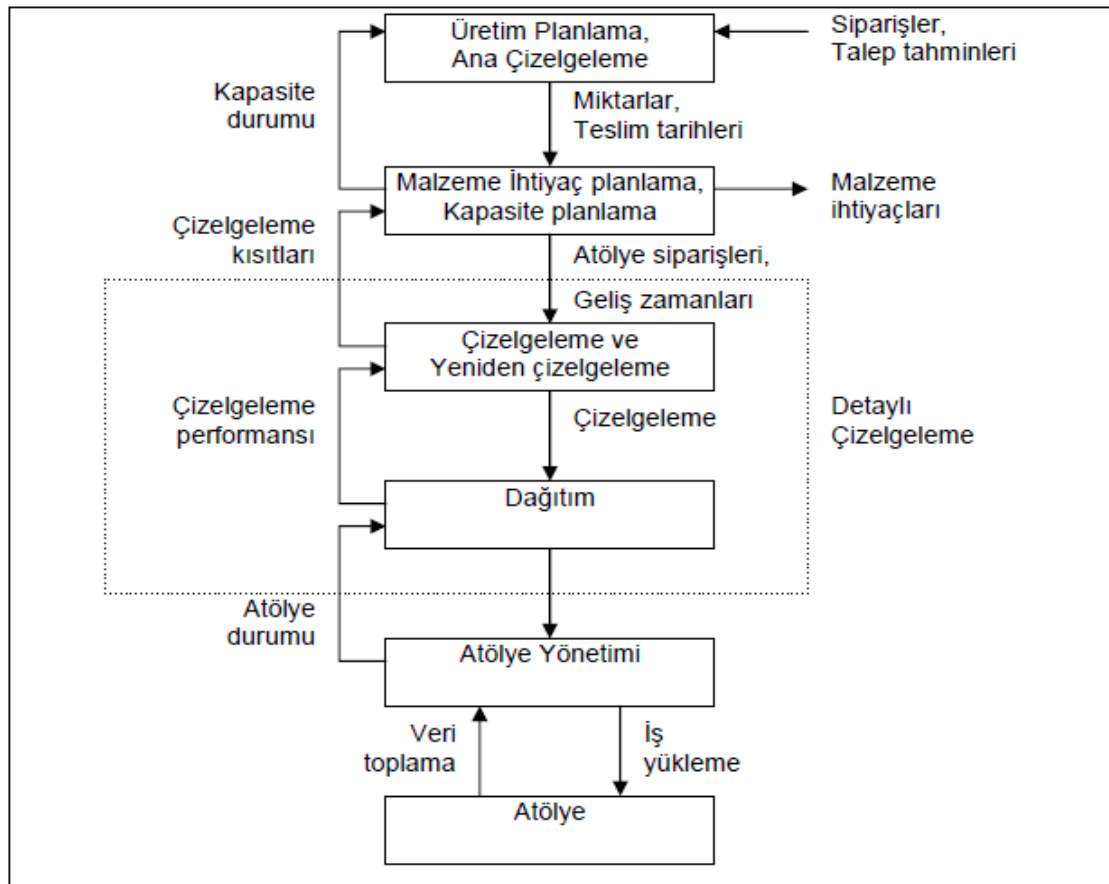
Çizelgeleme modelleri kombinatoriyal optimizasyon problemlerinden olduğu için en iyi çözüme ulaşmak çok zordur. Genellikle boyutu küçük olan problemler için optimum çözümler bulunabilmektedir (Eren ve Güner, 2002). İki amaçlı ve tek makineli problemlerle ilgili Van Wassenhove ve Gelders (1980), (pseudo-polinom algoritma) ile çözüm yöntemi geliştirmiştir. Cheun ve Bulfin (1990) ise (polinom algoritma) ile çözüm yöntemi geliştirmiştir. Bu algoritmaların her ikisi de iki amaçlı

problem için etkin çözüm yöntemleri üretmektedir. Çizelgeleme modellerini üretim kaynaklarının türüne göre ya da görevlerin doğasına göre belirleyebiliriz. Örneğin model tek makine mi içeriyor yoksa iki makine mi? İki makine içeriyorsa işler tek rotadan mı akacak yoksa iki ayrı rota mı oluşacak? Eğer işler çizelgelendikten sonra değişme olmuyorsa bu sisteme statik sistem, zaman içerisinde yeni işler çıkıyor ve sisteme dahil ediliyorsa buna da dinamik sistem denmektedir. Statik sistemler dinamik sistemlere göre daha kolay olduğundan geleneksel olarak üzerinde daha çok çalışılmaktadır. Son olarak tüm koşullar planlamanın başında kesin olarak biliniyorsa bu sistemlere deterministik sistemler, eğer tüm koşullar planlamanın başında kesin olarak bilinmiyorsa bu sistemlere de stokastik sistemler denmektedir (Baker ve Trietsch, 2009: 4).

Çizelgeleme problemlerinde makine kapasitesi ve bazı işlerin işlenmesi sırasında ortaya çıkan teknolojik kısıtlar olmak üzere iki farklı uygunluk kısıtı bulunmaktadır. Çizelgeleme problemleri atama ve sıralama işlemlerinden oluşmaktadır. Dolayısıyla çizelgeleme problemlerini çözme işlemi aşağıdaki sorulara cevap arama işlemidir (Baker ve Trietsch, 2009: 4).

- Hangi işler için hangi kaynaklar kullanılacak?
- Görevler ne zaman yapılacaktır?

Sisteme giriş yapan işler teslim tarihleri hesaplanmış bir şekilde hazır bulunmalıdır. Bu işler, işletme içerisinde bulunan makinelerde belirli bir sıra gözetilerek üretilecektir. Operasyonlar esnasında zaman çizgisinin bazı bölgelerinde iş yoğunluğunda artış olacak ve bu yüzden bazı işlerin üretiminde gecikmeler oluşacaktır. Aynı şekilde zaman çizgisinin bazı bölgelerinde ise kapasiteyi dolduracak kadar iş olmayacaktır ve ileri tarihli işler bu bölgelere kaydırılarak çizelge doldurulacaktır ya da makineler boş bırakılacaktır. Bununla birlikte makine arızası ya da işlem sürelerinin beklenenden uzun sürmesi gibi durumlar da çizelge yapılırken göz önünde bulundurulmalıdır (Muştu, 2013:6).



Şekil 1.1 Üretim sistemlerinde bilgi akış diyagramı (Pinedo, 2008: 5)

Bir işletmede çizelgeleme işletmenin içyapısı, kabiliyetleri ve hedefleri ile ilişkili bir şekilde yürütülebilir. Tam da bu noktada çizelgeleme, uzun periyotlarda gerçekleşecek ürün gruplarının ve kaynak dağıtımlarının optimizasyonu için stok seviyelerini, satış tahminlerini ve kaynak ihtiyaçlarını göz önünde bulundurur (Muştu, 2013: 7).

1.1 Çizelgeleme Problemlerinin Temel Özellikleri ve Performans Kriterleri

Literatürde çok sayıda üretim çizelgeleme probleminin olmasının sebebi; makine sayıları ve bunların birbiri ile ilişkileri, iş sayısı ve bunların birbirleri ile ilişkileri, işlerin sistemdeki akış şekilleri gibi seçeneklerin çok olmasıdır.

Graham vd. (1979)' nin yaptıkları çalışmada kullandıkları gösterimler günümüzde hala çalışmalara referans oluşturmaktadır. Çizelgeleme problemlerinde makinelerin ve işlerin sayısı sonlu olarak kabul edilmektedir. İşlerin sayısı n ile gösterilir ve birden çok iş varsa j alt indisi kullanılır J_j ($j=1,2,\dots,n$). Makinelerin sayısı ise m ile gösterilir ve makine sayısı birden fazla ise i alt indisi kullanılır M_i ($i=1,2,\dots,m$). Bu şekilde indisler

kullanılarak yapılan gösterimlerle bir iş sadece bir makinede işlem görebilir ve bir makine belirli bir zamanda sadece bir işle meşgul olabilir. Çizelgeleme problemlerinde en çok kabul gören gösterim biçimi $\alpha | \beta | \gamma$ üçlü yapıdaki gösterim biçimidir. Bu gösterimde α alanı makine bilgilerini ve üretim ortamının yapısını içermektedir. β alanı işlerin karakteristiklerinin detaylarını açıklamaktadır. γ alanı ise ele alınacak amaç fonksiyonunu temsil etmektedir (Koçanlı vd., 2012; Pinedo, 2008: 14). i makinesi ve j işi ile ilgili diğer terimler aşağıdaki gibidir.

Tablo 1.1 Çizelgeleme problemlerinin sınıflandırılması (Doğan 2013: 4).

<i>Makine Ortamı (α)</i>	<i>Kısıtlar (β)</i>	<i>Amaç (γ)</i>
I – Tek makine	r_j – Serbest kalma zamanı	Toplam ağırlıklı tamamlanma zamanı
P_m – Özdeş paralel makineler	d_j – Teslim zamanı	Enbüyük sapma
Q_m – Benzer makinalar	$prmp$ – Bölünebilir	Geciken iş sayısı
R_m – Farklı makinalar	$prec$ – Öncelikli	Toplam ağırlıklı gecikme
F_m – Akış tipi üretim	S_{jk} – Sıralamaya bağlı hazırlık süresi	Enbüyük taşıma süresi
J_m – Atölye tipi üretim	$fmls$ – İş aileleri	Toplam hazırlık maliyeti
O_m – Açık üretim	$batch(b)$ – Parti	Toplam hazırlık süresi
FF_m – Esnek akış tipi üretim	$brkdown$ – Arıza	Toplam akış süresi
FJ_m – Esnek atölye tipi üretim	M_j – Makine uygunluğu	En büyük tamamlanma zamanı
AF – Montaj akış tipi üretim	$prmu$ – Permütasyon	A – Makinenin hazır olması
	$block$ – Bloke olma	D – Toplam dağıtım süresi
	nwt – Bekleme yok	Z_{cost} – Çizelgeleme maliyeti
	$rcrc$ – Tekrarlı sirkülasyon	Z_{due} – Teslim zamanı atama maliyeti
	nbr – İş sayısı kısıtı	Z_{bottle} – Darboğaz oranını faydası
	n_j – İşlem sayısı kısıtı	$Z_{pjh-cost}$ – Üretme ve stokta tutma maliyeti
	p_j – İşlem süresi kısıtı	Z_q – Memnuniyet derecesi
	d_j – Teslim zamanı kısıtı	
	ST_{si} – Sıralamadan bağımsız hazırlık süresi	
	SC_{si} – Sıralamadan bağımsız hazırlık maliyeti	
	ST_{sd} – Sıralamaya bağlı hazırlık süresi	

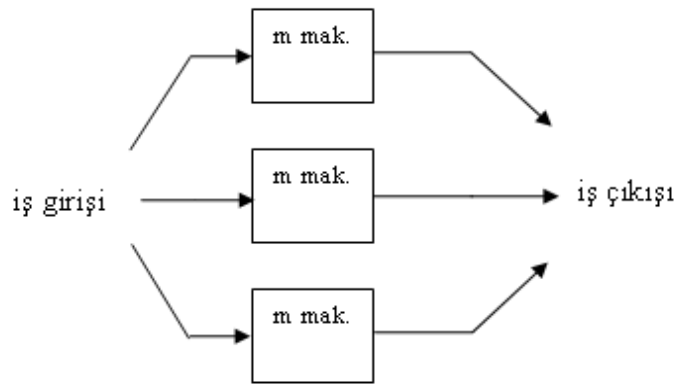
$ST_{sd,b}$ – Sıralamaya bağlı parti veya aile hazırlık süresi	N_t – Toplam geciken iş sayısı
$SC_{sd,b}$ – Sıralamaya bağlı parti veya aile hazırlık maliyeti	I_{sum} – Toplam makine boş kalma zamanı
R_{si} – Sıralamadan bağımsız serbest kalma zamanı	
R_{sd} – Sıralamaya bağlı serbest kalma zamanı	W_w – Toplam ağırlıklı bekleme zamanı
$R_{si,b}$ – Sıralamadan bağımsız parti veya aile serbest kalma zamanı	
$R_{sd,b}$ – Sıralamaya bağlı parti veya aile serbest kalma zamanı	F_{ave} – Ortalama akış zamanı
lsm – Parti gruplandırma	W_{ave} – Ortalama bekleme zamanı
$stch$ – Rassal	$C_{ave,w}$ – Ortalama ağırlıklı tamamlanma zamanı
$retr$ – Tekrarlı üretim	
$ovlp$ – İşlerin kesişmesi	$T_{ave,w}$ – Ortalama ağırlıklı gecikme zamanı
SC_{sd} – Sıralamaya bağlı hazırlık maliyeti	
$ST_{si,b}$ – Sıralamadan bağımsız parti veya aile hazırlık süresi	

- **İşlem zamanı (p_{ij}):** Bu gösterim j işinin i makinesindeki işlem süresini temsil etmektedir. İşlerin işlem süreleri makinelerden bağımsız ise i alt indisi kullanılmayabilir.
- **Serbest kalma süresi (r_j):** Bu gösterim j işinin hazır olma zamanını yani j işinin en erken başlayabilme zamanını göstermek için kullanılmaktadır.
- **Teslim tarihi (d_j):** j işinin tamamlanma zamanını yani müşteriye söz verilen termin zamanını göstermektedir. Teslim zamanından sonra işlerin tamamlanmasına müsaade vardır fakat bu durum gecikmeler için ceza uygulanmasını gerektirecektir.
- **Ağırlık (w_j):** j işinin diğer işlere göre önem düzeyini göstermektedir.
- **Tamamlanma zamanı (C_{ij}):** Tamamlanma zamanı j işinin i makinesindeki işleminin bittiği zamanı gösterir.

1.1.1 Makine Bilgileri (α)

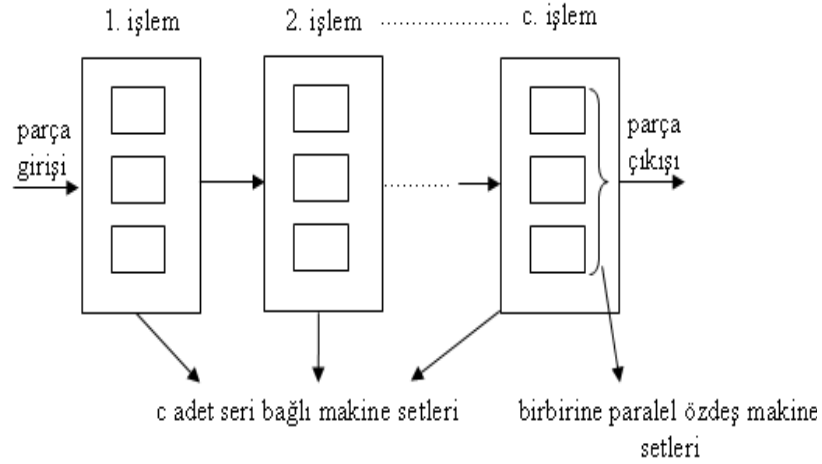
α simgesi ile tanımlanabilecek bazı üretim ortamları aşağıdaki gibidir (Pinedo, 2008: 14, 15).

- **Tek Makine (I):** Olası üretim ortamlarının en yalın hali olan tek makineyi ifade eder. Tek makine, diğer tüm karmaşık üretim ortamlarının özel bir durumudur. Karmaşık sistemlerin anlaşılabilmesi için tek makine modelinin iyi kavranması gerekmektedir. Her sistemin tek makineye indirilebilme durumu söz konusudur. Sistem içerisindeki olası darboğaz noktaları tek makine sistemi ile modellenebilir ve tüm sisteme bu şekilde yön verilebilir.
- **Özdeş Paralel Makineler (P_m):** Aynı özelliklere sahip paralel m adet makine bulunur. j işi tek bir işlemden oluşmaktadır ve herhangi bir makinelerde aynı sürede tamamlanabilmektedir.



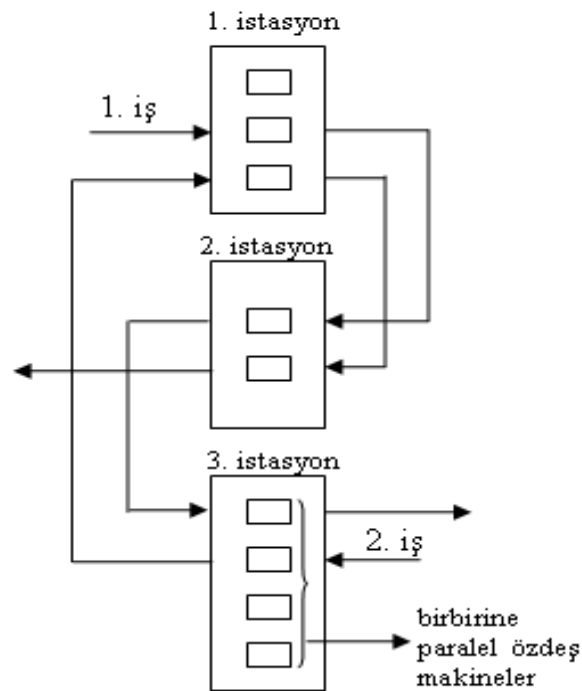
Şekil 1.2 Özdeş paralel makineler yerleşimi (Ak, 2012: 6).

- **Hızları Farklı Paralel Makineler (Q_m):** Hızları farklı m adet paralel makine söz konusudur. Mevcut işler herhangi bir makinede işlem görebilir. İşlerin makinelerdeki hızları değişiklik göstermektedir.
- **Akış Tipi (F_m):** Arka arkaya dizili m adet makine vardır ve işler bu makinelerin tamamında aynı rotayı takip ederek işlem görmek zorundadır. Bir iş bir makinede tamamlandıktan sonra diğer makinede işlem görmek için sıraya girmektedir. Montaj hatları bu üretim ortamı tipine örnek olarak verilebilir. Çeşitli çalışmalarda bu konu üzerine örnek uygulamalar ve çözüm yöntemleri geliştirilmiştir (Boysen, vd., 2009, McMullen ve Tarasewich 2003).
- **Esnek Akış Tipi (FF_c):** Paralel makine ile akış tipi üretim ortamlarının birleştirilmiş halidir. Birbirine seri bağlı makinelerden oluşan farklı makine grupları bulunmaktadır.



Şekil 1.3 Esnek akış tipi yerleşim (Ak, 2012: 7).

- **Atölye Tipi (J_m):** Her işin kendisine ait rotalarının olduğu birden fazla makinenin bulunduğu sistemlerdir.
- **Esnek Atölye Tipi Çizelgeleme (FJ_c):** Her işin kendine has rotaları vardır ve bu işler her makinede işlem görmek zorunda değildir. Bir makinede işlem gören iş de, birçok makinede işlem gören iş de aynı anda sistemde bulunabilmektedir.



Şekil 1.4 Esnek atölye tipi üretim ortamı (Ak, 2012: 8).

1.1.2 İşlem Karakteristikleri (β)

β simgesi ile tanımlanabilecek bazı üretim ortamları aşağıdaki gibidir (Pinedo, 2008: 15, 16, 17).

- **Sıra Bağımlı Hazırlık Zamanları (s_{jk}):** s_{jk} j işi ile k işi arasındaki sıraya bağımlı hazırlık sürelerini ifade etmektedir. İşlerin farklı işlerin arkasından gelmesi ile hazırlık süreleri farklılaşabilmektedir. k işi kuyruktaki ilk iş ise k işi için ayar zamanını s_{0k} gösterir, j işi kuyruktaki son iş ise s_{j0} temizlik zamanını gösterir. Eğer j işi ile k işi arasındaki ayar zamanı makineye bağlı ise i alt indisi eklenir ve s_{ijk} şeklinde gösterilir.
- **Yarıda Kesme ($prmp$):** Bir makinede üretimine başlanan işe tamamlanmadan ara verilip başka bir işe başlanması durumudur.
- **İşlem Önceliği Kısıtı ($prec$):** Bazı durumlarda işler arasında öncelikler bulunabilir. Bazı işler kendisinden önce başka işler üretilmeden işleme başlayamazlar.
- **Hazır Bulunma Zamanı (r_j):** İşlerin kendileri için belirlenen zamandan önce işleme başlayamaması durumudur.

Yukarıda belirtilen durumlar dışında farklı uygulamalar da bulunmaktadır.

1.1.3 Amaç Fonksiyonları (γ)

Çizelgeleme işlemlerinde işletmelerin yapılarına ve taleplerine göre birçok farklı amaç fonksiyonu oluşmaktadır. Genel olarak teslim zamanına göre, kaynak kullanımına göre ve akış zamanına göre belirlenmektedir. Bu temel amaçlar genel olarak birbiriyle ters düşebilmektedir. Optimize edilecek amaç fonksiyonları ile ilgili örnek çeşitleri aşağıdaki gibidir (Pinedo, 2008: 18, 19).

- **Maksimum Tamamlanma Zamanı (C_{max}):** Sistemden çıkacak son işin tamamlanma zamanını temsil eder. Bu fonksiyonun minimizasyonu makine kullanım süresini yükseltecektir.
- **Maksimum Gecikme (L_{max}):** İşlerin teslim tarihleri ile o işlerin sistemden çıkış zamanları arasındaki farkın en büyüğünü ifade eder.
- **Toplam Tamamlanma Zamanı ($\sum C_j$):** İşlerin sistemden çıkış zamanlarının toplamını ifade eder.

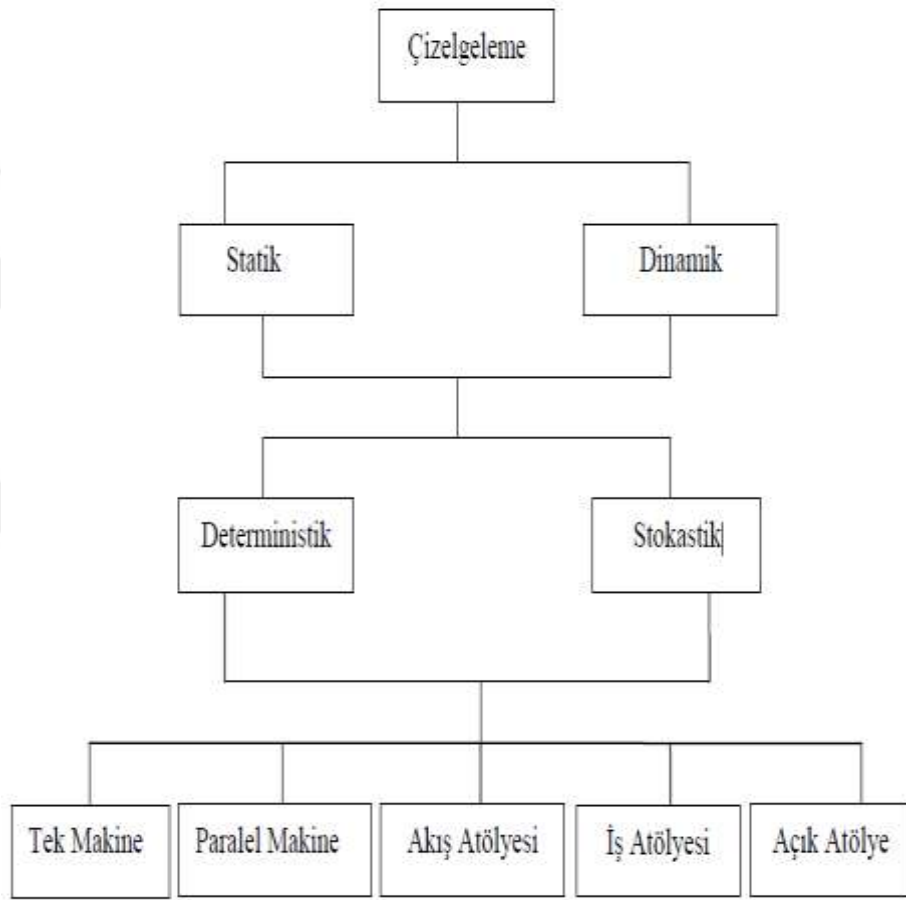
- **Ağırlıklandırılmış Toplam Tamamlanma Zamanı ($\sum w_j C_j$):** İşlerin yapısal olarak önem düzeylerinin farklı olduğu durumlar için kullanılır. Bu fonksiyon işlerin oluşturacağı stok maliyetinin bir ölçütünü verir.
- **Toplam Gecikme ($\sum T_j$):** Son teslim tarihinden sonra tamamlanan işlerin gecikme miktarlarının toplamını ifade etmektedir. Çeşitli çalışmalarda toplam gecikme fonksiyonu inceleme konusu olmuştur (Sen, vd. 2003, Koulamas, 1994).
- **Toplam Ağırlıklandırılmış Gecikme ($\sum w_j T_j$):** İşlerin önem düzeyine göre gecikme miktarlarının o işlerin ağırlıklarıyla çarpımı ile elde edilir. Çeşitli çalışmalarda ağırlıklandırılmış toplam gecikme fonksiyonu inceleme konusu olmuştur (Abdul-Razaq, vd. 1990, Sen, vd. 2003).
- **Toplam Geciken İş Sayısı ($\sum U_j$):** Teslim tarihinden sonra tamamlanan işlerin sayısını gösterir. Geciken işlerin gecikme miktarları ile ilgilenmez, sadece geciken iş sayısını minimize etmek ister.
- **Toplam Ağırlıklandırılmış Geciken İş Sayısı ($\sum w_j U_j$):** Teslim tarihlerinden sonra tamamlanan işlerin sayısının o işlerin ağırlıkları ile çarpılması ile elde edilir.
- **Toplam Erken Tamamlama ($\sum E_j$):** Eğer bir iş istenen zamandan erken tamamlanıyorsa ve bu erken tamamlanma bir maliyet unsuru oluşturuyorsa kullanılır. Erken tamamlanmaların minimize edilmeye çalışılması ara stok maliyetlerini düşürücü bir etki yaratacaktır.
- **Toplam Ağırlıklandırılmış Erken Tamamlama ($\sum w_j E_j$):** İşlerin erken tamamlanmasıyla ilgili bir maliyet unsuru bulunuyorsa ve bu işler arasında bir önem sıralaması veya önem düzeyi bulunuyorsa kullanılır.

Çizelgeleme problemlerinin gösterimine örnek verilecek olursa; $FF_c / r_j | \sum w_j T_j$ notasyonu; esnek akış tipi üretimde işlerin serbest kalma zamanına sahip olduğunu ve amaç fonksiyonunun ağırlıklandırılmış gecikmenin en küçüklenmesi olduğunu ifade etmektedir (Pinedo, 2008: 20).

1.2 Çizelgeleme Modelleri

Çizelgeleme modelleri stokastik, deterministik, çok makineli, tek makineli veya atölye tipi gibi değişik özelliklerde olabilmektedir. Dolaylı olarak göz önüne alınan bir diğer

ölçüt de problemin çözülebilirliğidir. Deterministik çizelgeleme problemleriyle ilgili araştırmaların büyük bir bölümü etkili (polinom zamanlı) çözüm algoritmalarına bağlıdır. Buna karşın birçok çizelgeleme problemi ise, polinom zamanlı algoritmalara sahip değildir. Bu tür problemler NP-Zor (Non-deterministic Polynomial Hard) olarak tanımlanırlar (Pinedo, 2008: 26; Dağ, 2012: 10). Klasik çizelgeleme modelleri ve sistemleri hiyerarşik olarak Şekil 2.5'te görüldüğü gibi sınıflandırılır.



Şekil 1.5 Çizelgeleme modelleri (Dağ, 2012: 10)

1.2.1 Dinamik Çizelgeleme

Çizelgelenecek işlerin zaman içerisinde değişiklik gösterdiği durumları ifade eder. Belli bir zaman çizgisi içerisinde işler rastgele bir şekilde sisteme giriş yapar. Statik modellere göre uygulamada ciddi zorluklar oluşturmaktadır.

1.2.2 Statik Çizelgeleme

Çizelgelenecek işlerin sayısı ve sisteme giriş zamanları önceden biliniyorsa ve değişmiyorsa bu sistemlere statik çizelgeleme problemleri denir. Statik modellerin

analizi genel durumların analizini kolaylaştırmaktadır ve çeşitli sezgisel yöntemlerin bulunmasında etkili olmuştur (Muştu, 2013: 19).

1.2.3 Deterministik Modeller

Eldeki çizelgelenecek işlerin işlem zamanları, geliş zamanları veya teslim zamanları gibi gerekli tüm parametreler biliniyorsa, bu tarz problemler deterministik olarak adlandırılır.

1.2.4 Stokastik Modeller

Eldeki çizelgelenecek işlerin işlem zamanları, geliş zamanları veya teslim zamanları gibi gerekli parametreler belirsiz ise, bu tarz problemler stokastik olarak adlandırılır. Gerçek hayatta rastlanan, makine bozulmaları, beklenmeyen siparişler ve düzensiz işlem zamanlarını içeren problemler stokastik problemlerdir (Dağ, 2012: 11).

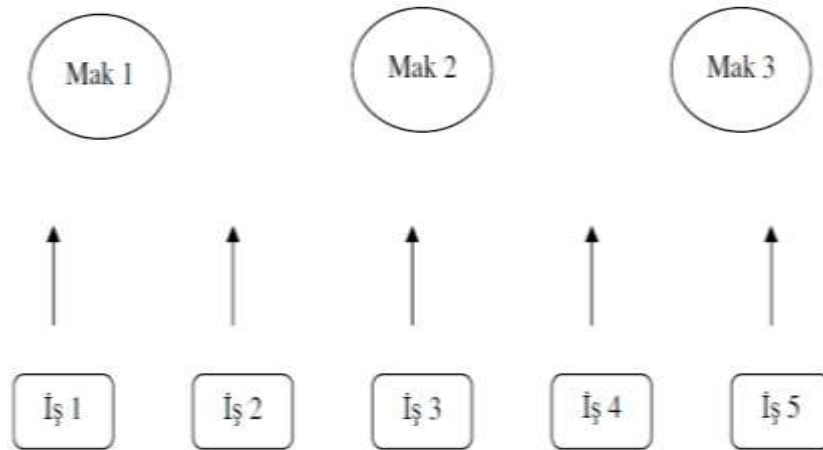
1.2.5 Tek Makine Modeli

Çizelgeleme problemlerinin en basit modelidir. Çizelgeleme sistemlerinin temelini oluşturur ve üzerinde en çok çalışılan modeldir. Basit olmakla birlikte diğer modelleri anlayabilmek için çok iyi kavranması gerekmektedir. n adet iş için $n!$ çözüm bulunabileceği için iş sayısı arttıkça çözüm üretme zorluğu katlanarak artmaktadır.

Problemin yapısı firma karakteristiklerine ve taleplerine bağlı olarak stokastik veya deterministik olabilir. Bu iki durum için de kullanılacak yöntemler benzer yöntemlerdir. Optimum sonuçlara ulaşabilmek için dal-sınır algoritmaları ve dinamik programlama modelleri kullanılırken, büyük problemler için genetik algoritma, tavlama benzetimi ve tabu arama gibi metasezgisel çözüm yöntemleri kullanılabilir.

1.2.6 Paralel Makine Modeli

En basit haliyle; paralel makine çizelgeleme modelinde n adet iş bulunur ve bu işler birbirine özdeş m adet makineden herhangi birinde işlenebilir. Başlangıç anında n adet iş ve m adet makine işleme hazır haldedir. İşler uygun olan makinelerde işlem görebilir. Temel paralel makine çizelgelemede işler birbirinden bağımsız ve makineler paralel kabul edilir (Baker ve Trietsch, 2009: 200). Şekil 2.6 da bir örnek gösterilmiştir.



Şekil 1.6 Paralel makineli bir model (Ceran, 2006: 11)

Tüm bu anlatılan çizelgeleme modelleri dışında akış tipi, atölye tipi ve açık atölye tipi çizelgeleme modelleri de bulunmaktadır.

1.3 Tek Makine Çizelgeleme Modelleri

Çalışmanın temelini oluşturduğu için bundan sonra literatür taramasına tek makine çizelgeleme modelleri üzerinden devam edilecektir. Makine çizelgeleme problemleri literatürde üzerinde oldukça fazla çalışılan problemlerden biridir. Smith (1956)'ın yapmış olduğu çalışmadan günümüze tek makine çizelgeleme modelleri üzerine de çok fazla çalışma yapılmıştır. Bu bölümde tek makine çizelgeleme problemlerinden olan toplam gecikme problemi (TGP) ve toplam ağırlıklandırılmış gecikme problemi (TAGP) ele alınacaktır. Sonrasında ise işleri erken bitirmenin de geç bitirmenin de cezalandırıldığı modellerden farklı teslim tarihli erkenlik/geçlik (E/G) problemi incelenecektir. İlk olarak $t=0$ anında tüm işlerin hazır bulunduğu işlem sürelerinin p_j , teslim tarihlerinin d_j ve ağırlıkların w_j farklı ve birbirinden bağımsız olduğu problem tipi incelenecektir ($1 \parallel \sum T_j$, $1 \parallel \sum w_j T_j$). Ardından E/G problemleri ele alınacak ve sonrasında ise sıra bağımlı hazırlık süreleri (s_{ijk}) içeren problemler incelenecektir. En son olarak ise sıra bağımlı hazırlık süreleri ile E/G problemleri entegre edilerek incelenecektir.

1.3.1 $1 \parallel \sum T_j$ ve $1 \parallel \sum w_j T_j$ Problemi

Çizelgeleme problemleri üzerinde en çok tek makine toplam gecikme problemi ve toplam ağırlıklı gecikme problemleri üzerine çalışmalar yapılmıştır. TAGP ile ilgili

olarak Abdul-Razaq vd. (1990), TGP ile ilgili olarak Koulamas (1994) ve her iki problem tipi ile ilgili olarak Sen vd. (2003)'nin çalışmaları en önde gelen çalışmalardır.

Tek makine çizelgelemede TGP ve TAGP yapısı şu şekildedir. n adet iş $t=0$ anında hazır bulunur, hiçbir işin öncelik kısıtı yoktur ve makine herhangi bir zamanda sadece bir işi işleyebilmektedir. İşler bölünemez, işler sadece bir kez işlem görür, makinelerin boşta kalmasına müsaade edilmez ve işler arka arkaya sıralanır. İşlerin işlem süreleri p_j , teslim tarihleri d_j ve ağırlıkları w_j farklı ve birbirinden bağımsızdır. Tüm bu kabuller altında bu işler için toplam gecikmenin ya da toplam ağırlıklı gecikmenin en küçük olduğu çözümler aranır (Muştu, 2013: 23).

Bu konuyla ilgili öncü çalışmalar; Schild ve Fredman (1961)'in TGP problemini dal sınır algoritması yöntemi ile çözdüğü ve Held ve Karp (1962)'in TGP problemini dinamik programlama yöntemi ile çözdüğü çalışmalardır. Bu çalışmalar da en iyi çözümü bulmaya yönelik öneriler ve yöntemler getirilmiştir, dolayısıyla büyük boyutlu problemler için uygun çözümler üretilmemiştir.

Elmaghraby (1968), kendinden önceki çalışmalarda geçen dinamik programlama yöntemine benzeyen bir şebeke modeli oluşturmuştur ve amaç fonksiyonunu en kısa yol problemine dönüştürmüştür. Ayrıca dal sınır algoritmasının dinamik programlamaya göre daha hızlı çözümler üretebildiğini göstermiştir. Sonraları bu çalışma üzerine deneyler yapan Shwimer (1972), Elmaghraby (1968) çalışmasında geçen 6 adım metodunun uygulamada programlamasının zor olacağını göstermiştir. Emmons (1969), TGP problemi için teorik bir çözüm yöntemi geliştirmiş ve işlerin işlem süreleri ile teslim tarihleri arasında bir ilişki kurmuştur. Geliştirdiği yöntemler ile daha büyük problemler için çözüm bulunabilmesini kolaylaştırmıştır.

Rinnooy Kan vd. (1975), TAGP için geliştirdikleri dal sınır algoritmasını Emmons (1969)'un modelini referans alarak yapmışlardır. Baker ve Schrage (1978), TGP problemi için zincir algoritması olarak anılan algoritmayı geliştirmişlerdir. Bu algoritma öncekilerle kıyaslandığında daha uygun zamanlarda çözümler üretebilmektedir. TGP için bir başka optimizasyon tekniği ise Sen vd. (1983)'in geliştirdiği ve MINIT olarak anılan tekniktir. Bilinen öncelik ilişkilerine dayalı olarak çalışan yöntem daha az hafıza kullanımı gerektirmektedir. Baker ve Schrage (1978)'in dinamik programlama yöntemi ile yaptıkları kıyaslamalarda, hafıza kısıtı olan durumlar için MINIT yönteminin diğer

durumlarda ilse dinamik programlama yönteminin tercih edilebilir olduğunu göstermişlerdir.

Rachamadugu (1987), iki komşu işin konumunu optimum bir sırayla karakterize eden bir koşul tanımlayarak, TAGP için etkin çözüm üretmeye çalışmıştır. Potts ve Van Wassenhove (1991), TAGP için genel sıralama yöntemleri ile birlikte sezgisel yöntemleri de incelemiştir. Yazarlar tavlama benzetimi gibi yöntemler kullanarak komşu değişimi uygulamış ve optimuma yakınsayan yöntemlerin etkisini göstermiştir.

Tansel ve Sabuncuoğlu (1997), TGP problemi için baskınlık kurallarının geometrik izahını yapmışlardır.

1.3.2 Farklı Teslim Tarihli E/G Problemleri

Günümüz şartlarında üretim sistemlerine giren işleri istenen teslim tarihinden önce tamamlamak da geç tamamlamak da bir maliyet unsuru oluşturmaktadır. Erken tamamlamak stok maliyeti, hurda ve fire gibi maliyetler oluştururken, geç tamamlanan işler ise müşteri memnuniyetsizliği ve cezalara sebep olabilmektedir (Li vd. 1995).

Yukarıdaki sorunlar göz önüne alındığında tam zamanında üretim kavramına olan ilgi artmış ve bu konuda önemli araştırmalar yapılmıştır. Tam zamanında üretim felsefesinin ile kurgulanan çizelgeleme problemlerinin en yaygın amaçlarından biri, erkenlik ve geçlik cezalarının en küçüklenmesidir. Bu probleme erkenlik/geçlik üretim çizelgeleme problemi denir. j işinin erkenlik ve geçliği olarak sırasıyla E_j ve T_j gösterimleri dikkate alınır; genel bir E/G probleminde j işinin erkenliği, E_j , $E_j = \max(0, d_j - C_j)$ ile, j işinin geçliği, T_j , $T_j = \max(0, C_j - d_j)$ ile tanımlanır. Her işe ilişkin birim erkenlik cezası α_j ve birim geçlik cezası β_j vardır. Ceza fonksiyonlarının doğrusal olduğu varsayılarak, bir çizelge için standart E/G amaç fonksiyonu $f(s) = \sum_{j=1}^n (\alpha_j E_j + \beta_j T_j)$ olarak yazılabilir (Özdemir, 2007: 8). Bu problem tipi için işlerin teslim tarihlerinin farklı olmasına ve işlerin teslim tarihlerinin aynı olmasına göre değişen durumlar mevcuttur. Bu tezde teslim tarihleri farklı olan durumlar incelenecektir.

Bu tür E/G modellerinde her iş kendine ait bir teslim tarihine sahiptir ve bu durum problemin çözümünü zorlaştırmaktadır. Koullamas (1996), $t=0$ anında işlerin üretim zamanları ve zaman pencereleri aralıkları belli olan deterministik bir E/G problemini

incelemiştir. Her iş için belirlenen zaman aralığı o işin işlem süresinden büyüktür. İşler bu zaman aralığından önce başlar veya sonra biterse sapma oranınca ceza alır. Eğer işler kendisi için belirlenen zaman aralığında tamamlanırsa herhangi bir ceza almazlar. Yazar, çeşitli sezgiseller kullanarak çözüm önerileri getirmiştir. Chang (1999), $t=0$ anında hazır bulunan ve tüm parametreleri belli olan işleri tam zamanında üretim mantığını baz alarak çizelgelemeye çalışmıştır. Dal sınır algoritması yöntemini kullanarak probleme çözüm üretmiştir.

Abdul-Razaq vd. (1990), teslim tarihi cezası içeren bu tip problemi çözmüş, fakat çizelgelerde boş zamanlar oluşmasına izin vermemişlerdir. Problemlerin çözümü için dal sınır algoritması önermişler ve iyi sınırlar elde etmek için değiştirilmiş bir dinamik programlama modeli ile çalışmışlardır. Elde ettikleri sonuçlara göre 20'den fazla iş içeren problemlere çözüm üretebilmek için çok fazla zamana ihtiyaç olduğunu göstermişlerdir (Özdemir, 2007: 23).

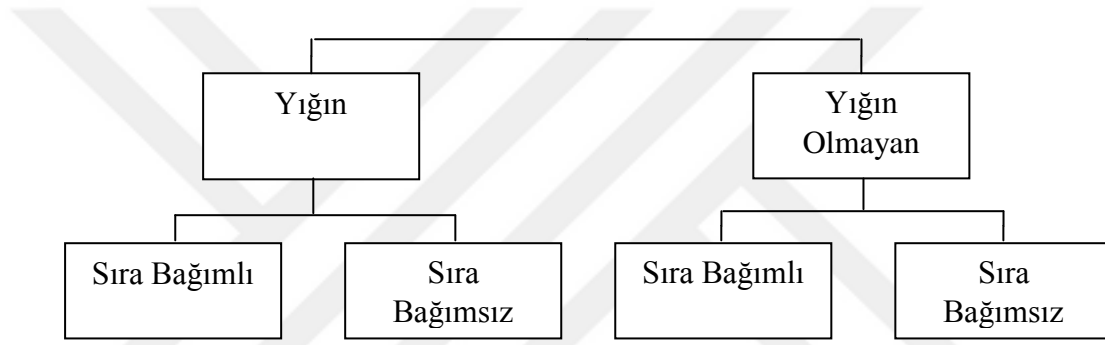
Valente ve Alves (2005), çalışmalarında tek makinede ağırlıklı erkenlik ve geçlik toplamını en küçükleme problemini ele almıştır. Boş zamana izin verilmeyen problem için filtreli ve iyileştirilmiş direkt arama algoritmaları sunmuşlar ve var olan komşuluk arama ve dağıtım kuralı sezgiselleriyle karşılaştırmışlardır. İşlemsel sonuçlar, iyileştirilmiş direkt arama algoritmalarının filtreli benzerlerinden daha üstün olduğunu ve öncelik-tabanlı filtreleme yönteminin kurallar-tabanlı alternatife göre üstünlüğünü kanıtlamıştır. En iyi çözümler, komşuluk arama algoritması ile elde edilmiştir fakat bu yöntem işlemsel olarak çok fazla zamana ihtiyaç duymaktadır ve içerdiği iş sayısı az olan örneklere uygulanabilmektedir. İyileştirilmiş direkt arama sezgiseli kaliteli ve daha hızlı bir şekilde çözümler üretmiştir, bu nedenle küçük problemleri bile çözmek için kullanılabilir (Özdemir, 2007: 24).

1.3.3 Sıra Bağımlı Hazırlık Süresi İçeren Modeller

Literatürün çoğunda makinelerin işleri işlemek üzere hazır bulunduğu veya herhangi bir hazırlık süresine ihtiyaç duymadığı varsayılır. Diğer durumlarda ise hazırlık sürelerinin işlem zamanının içine gömülü olarak verildiği düşünülür. Her iki durum da birbirinden çok farklı değildir. Oysa gerçek hayat problemlerinde çoğunlukla hazırlık süreleri mevcuttur ve bunlar çizelgeyi ciddi şekilde etkileyecek kadar önemli düzeydedir.

Tek makine problemlerinde $r_j=0$ olduğunda ve sıra bağımlı hazırlık süreleri olmadığında tamamlanma zamanı sıralamadan bağımsız ve işlem sürelerinin toplamına eşit olmaktadır. Sıra bağımlı hazırlık süreleri olduğunda ise tamamlanma zamanı çizelgeye bağlı olmaktadır. Bu durum sıra bağımlı hazırlık süreli problemleri NP- zor sınıfına sokmaktadır (Pinedo, 2008: 84).

Allahverdi vd. (1999), hazırlık süresi içeren çizelgeleme modelleri ile ilgili literatür taraması yapmışlardır. Çalışmaları yığın ve yığın olmayan diye iki kategoriye ayırmışlardır. Her iki kategorinin altında sıra bağımlı hazırlık süreli ve sıra bağımsız hazırlık süreli iki alt başlık oluşturmuşlardır. Bu gösterim Şekil 2.7’de gösterilmiştir.



Şekil 1.7 Çizelgeleme problemlerinin hazırlık süresine göre sınıflandırılması.

Tan ve Narasimhan (1997), yapmış oldukları çalışmada toplam gecikmeyi minimize etmek üzere tek bir amaç fonksiyonu kullanmışlardır. Etkin sezgisel yöntemlerden tavlama benzetimi algoritmasını kullanarak, inceledikleri problem için uygun işlem süreleri içerisinde etkin çözümler elde etmeyi başarmışlardır.

Allahverdi vd. (2008), hazırlık süresi içeren çizelgeleme modelleri ile ilgili ikinci kez literatür taraması yapmışlardır. Bu incelemelerinde ilk yaptıkları çalışmadan buyana her yıl ortalama 40 sayfa çalışmanın çizelgeleme problemlerinde hazırlık sürelerini kullandığını belirtmişlerdir. Yazarları bu çalışmada da literatürü yığın ve yığın olmayan diye iki ana başlığa ayırmıştır. Schaller vd (2000), iş aileleri arasında bulunan sıra bağımlı hazırlık sürelerini incelemiştir. Çizelge, tamamlanma zamanını minimize etmeye çalışmış ve bunun için çeşitli sezgisel yöntemler kullanılmıştır.

Lee ve Asllani (2004), yaptıkları çalışmada maksimum tamamlanma zamanı ve toplam geciken iş sayılarını minimize eden ikili amaç fonksiyonu oluşturmuşlardır. 0-1 karışık tamsayı programlama algoritması kullanarak inceledikleri problem için matematiksel

model geliřtirmişlerdir. Bu model ile 20 işe kadar olan çizelgelerde uygun zaman içerisinde optimum çözümler elde etmeyi başarmışlardır. Problem NP-zor sınıfına girdiđi için 20 işten fazla olan işlerde matematiksel model anlamsız kalmıştır ve genetik algoritma yöntemi kullanılarak yüksek sayılı işler için uygun zamanlarda çözümler elde edilmiştir.

Eren ve Güner (2006), çalışmalarında toplam ađırlıklı tamamlanma zamanı ile toplam ađırlıklı gecikmeyi eniyileyen ikili amaç fonksiyonu geliřtirmişlerdir. Bu problemin çözümü için tamsayılı programlama modeli geliřtirmişlerdir ve problem NP-zor sınıfında olduđu için 12 işe kadar uygun süreler içerisinde çözümler elde etmişlerdir. Tamsayılı programlama modelinin uygun çözümler üretmediđi durumlar için sezgisel yöntemlerden tabu arama algoritmasını kullanmışlardır ve 1000 işe kadar olan çizelgeler için uygun zamanda çözümler üretmişlerdir.

Gupta ve Smith (2006), yapmış oldukları çalışmada toplam gecikmeyi minimize eden bir amaç fonksiyonu oluşturmuşlardır ve yerel arama sezgisel algoritmalarını kullanarak çözümler üretmeye çalışmışlardır. Allahverdi (2014), hazırlık süresi içeren çizelgeleme modelleri ile ilgili üçüncü kez literatür taraması yapmıştır. Son çalışmasından bu yana hazırlık süresi içeriden çalışmalarda artış olmuş ve yıllık ortalama 60 sayfa çalışma miktarına ulaşılmıştır. Yazar, literatürü sınıflandırma yöntemi olarak yine yığınlı ve yığınlı olmayan iki ana başlık kullanmıştır.

Bigras vd. (2008), çalışmalarında tek makine sıra bađımlı hazırlık süreli ortamı inceleyerek, toplam tamamlanma zamanını ve toplam gecikme zamanını minimize edecek iki farklı amaç fonksiyonu oluşturmuşlardır. Bu problem için zaman bađımlı gezgin satıcı yaklaşımını kullanarak dal sınır algoritması yöntemi ile çözüm üretmeye çalışmışlardır.

1.3.3.1 Sıra Bađımlı Hazırlık Süresi İçeren E/G Problemleri

Coleman (1992), tek makinede n işin çizelgelenmesi için 0/1 karışık tamsayılı programlama modelini kullanmıştır. Çalışmanın amacı sıraya bađımlı hazırlık süreleri içeren problemler için ađırlıklı erkenlik ve geçliđi en küçüklemektir. Cezaların işe bađımlı oldukları varsayılmıştır ve işlerin teslim tarihleri birbirinden farklıdır. Coleman (1992), tamsayılı programı test etmiş ve 8 işe kadarki örnek problemler için en iyi

çözümler bulabilmiştir. İşlerin sayısı arttıkça tamsayılı programlamanın çözüm üretebilmek için yüksek işlem zamanına ihtiyacı olmuştur (Özdemir, 2007: 29).

Chen (1997), yığın sıraya bağımlı hazırlık süreleri içeren E/G problemini iki örnek problem için incelemiştir. İlkinde iş yığınları belli bir ortak teslim tarihine ve eşit olmayan cezalara sahiptir. Sadece iki iş yığını içeren bir problem olması ve iki teslim tarihinin kısıtlandırılmamış derecede büyük olması durumları için problemin NP-zor olduğunu göstermiştir. Makalenin diğer bölümünde teslim tarihlerinin karar değişkenleri olarak alınması dışında diğer değişkenler aynı kalacak şekilde problem ele alınmıştır. Yazar, problem için bazı eniyileyici özellikler sunmuş ve iki yığın işli problemi çözebilmek için bir polinom dinamik programlama algoritması kullanmıştır. (Özdemir, 2007: 29).

Kolahan ve Liang (1998), yapmış oldukları çalışmada tek makine problemini sıra bağımlılık kısıtı ile incelemiştirlerdir. Çalışmada, toplam ağırlıklı erken bitirmeyi ve toplam ağırlıklı gecikmeyi minimize edecek şekilde bir amaç fonksiyonu oluşturulmuştur. Problemin kombinatoryal yapısı gereği NP- zor sınıfında olduğu belirtilmiş ve uygun zaman aralığında çözüm üretebilmek için tabu arama yaklaşımı kullanılmıştır.

Rabadi vd. (2004), yapmış oldukları çalışmada aynı teslim tarihine sahip işleri incelemiştirlerdir. Tek makinede, sıra bağımlılık kısıtı altında, toplam erken bitirme ve toplam gecikme amaç fonksiyonlarını aynı anda minimize etmeye çalışmışlardır. Yazar, erken bitirmenin ve geç bitirmenin ağırlıklarını eşit kabul etmişlerdir. Problem yapısı gereği NP-zor kabul edilmiş ve dal sınır algoritması yöntemi ile probleme çözüm üretilmeye çalışılmıştır.

1.4 Çözüm İçin Değerlendirme.

Problemin temel yapısı aşağıdaki gibidir.

- Tek makinede çizelgeleme yapılacaktır.
- Sıra bağımlı hazırlık süreleri vardır.
- İşlerin teslim tarihleri farklıdır.
- Erken bitirmede her iş için kendi ağırlığı olan bir ceza vardır.

- Erken bitirme zamanı karesi alınarak fonksiyona dahil edilecektir. Böylelikle erken bitirmelere katlanarak artan bir ceza maliyeti uygulanmış olacaktır.
- Geç bitirme için her iş için sabit ve aynı olan bir katsayı cezası vardır.
- Sıra bağımlı hazırlık süreleri de tüm işler için aynı katsayı ile amaç fonksiyonunda bulunacaktır.
- Amaç fonksiyonu üç farklı amacı minimize etmek için kurgulanacaktır.

Bu şartlar ve kısıtlar altında amaç fonksiyonu $f(s) = \sum_{j=1}^n (\alpha_j E_j^2 + \beta T_j + \mu s_j)$

şeklinde olacaktır ve problemin gösterimi $\mathbf{1} | s_{jk} | \sum_{j=1}^n (\alpha_j E_j^2 + \beta T_j + \mu s_j)$ formunda olacaktır.

- α_j : İşleri erken tamamlamanın ağırlığıdır. İki unsur içermektedir, birincisi işlerin kilogram cinsinden miktarlarına bağlı bir elde bulundurma maliyeti, diğeri ise tellerin sarılı olduğu makaraların boşta beklemelerinden kaynaklı bir elde bulundurma maliyetidir.
- β : işlerin geç tamamlanmasının ağırlığıdır ve işlerin miktarından bağımsızdır. Kendisinden sonraki makineye yarı mamul üreten makinedeki gecikmelerin diğerk makinede oluşturacağı beklemeyle ilişkili bir fırsat maliyeti miktarıdır.
- μ : hazırlık zamanlarının ağırlığıdır ve işlerin miktarından bağımsızdır. Makinede hazırlık zamanlarından dolayı oluşan atıl kalmayla ilişkili bir fırsat maliyeti miktarıdır.

2. BÖLÜM

GENETİK ALGORİTMA

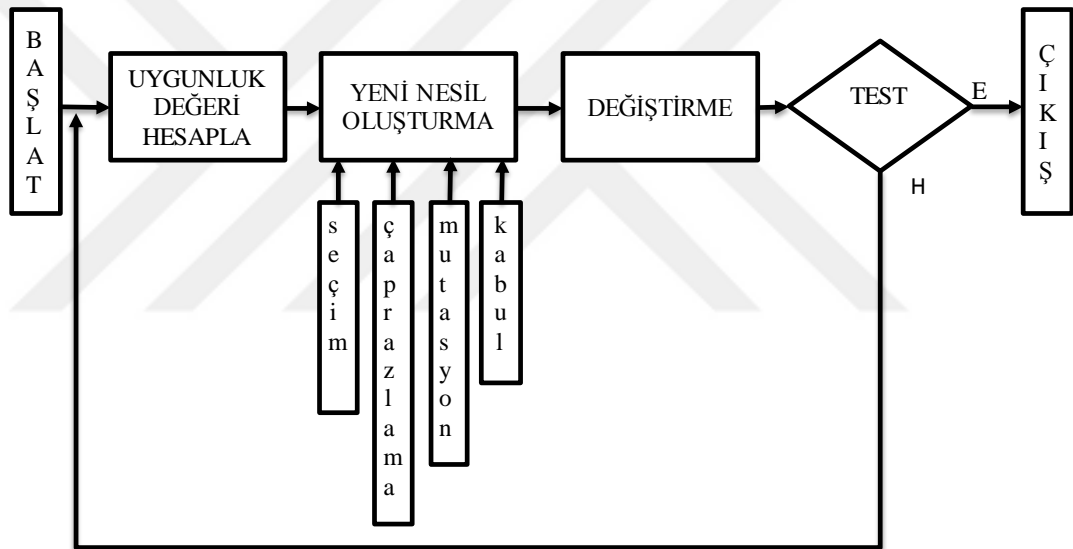
Genetik algoritma (GA) biyoloji bilimindeki genetik ve doğal seleksiyon konularını örnek alan, bir çeşit eniyileme ve arama tekniğidir. GA, birçok bireyden oluşan popülasyonların belirtilen seçim kuralları çerçevesinde, uygunluğunu maksimize etmeye çalışır (Haupt ve Haupt, 2004: 22).

GA ilk olarak Holland (1975) tarafından kullanılmıştır. Biyolojideki genetik kavramından etkilenilen John Holland, genetik süreçleri bilgisayar ortamında gerçekleştirmek istemiştir. Çoğalma, çiftleşme, mutasyon vb. genetik süreçlerin kullanılmasıyla, öğrenebilen ve bilgi düzeyini geliştirebilen mekanik yapılar oluşturulabildiğini görmüştür ve geliştirdiği yöntemin adı Genetik Algoritma olarak yerleşmiştir (Ceran, 2006: 113).

GA kromozom adı verilen bireylerin başlangıç popülasyonunu oluşturulmasıyla başlar. Genlerden oluşan her kromozom aday çözümdür. Amaç fonksiyonunda istenen kriterleri sağlayan en iyi bireyler bir sonraki popülasyona (nesle) aktarılır. Uygunluk dereceleri düşük olan bireylerin genlerini bir sonraki nesle aktarmasına izin verilmez. Elenen kromozomların yerine iyi kromozomlardan seçilen bireylerle çaprazlama ve mutasyon operatörleri kullanılarak yeni bireyler üretilir. Belirlenen amaç fonksiyonu kriterlerini sağlayıncaya kadar ve durdurma kriterlerine ulaşıncaya kadar bu adımlar tekrar edilir (Ak, 2012: 49-50). GA'nın genel olarak adımları aşağıdaki gibidir:

- GA'ya başla.
- Başlangıç popülasyonunu oluştur, her birey için uygunluk değerlerini sırala ve en iyi sonuçları sırala. En iyi sonuç durdurma kriterini sağlıyorsa GA'yı sonlandır.

- Çaprazlama işlemini uygula. Oluşan yeni bireyler için uygunluk değerlerini hesapla ve en iyi sonuçları sırala. En iyi sonuç durdurma kriterini sağlıyorsa GA'yı sonlandır.
- Mutasyon işlemini uygula. Mutasyon işlemi sonucunda ortaya çıkan bireyler için uygunluk değerlerini hesapla ve en iyi sonuçları sırala. En iyi sonuç durdurma kriterini sağlıyorsa GA'yı sonlandır.
- Belirlenen kurala göre bireyler seç ve yeni popülasyonu oluştur.
- Belirlenen nesil sayısına ulaşılmamışsa çaprazlama adımına git ve tüm işlemleri yeniden uygula.
- GA'yı sonlandır.



Şekil 2.1 Genetik algoritma genel akış şeması (Bolat vd. 2004).

2.1 Başlangıç Popülasyonu

GA, başlangıç popülasyonu olarak adlandırılan bir grup kromozomla işe başlar. Genel olarak literatürde başlangıç çözümü en basit haliyle rastsal olarak oluşturulmaktadır. Lakin başlangıç çözümü GA sonuçlarını ciddi şekilde etkilemektedir ve bu yüzden başlangıç çözümleri için uygun sezgiseller kullanarak daha iyi sonuçlar elde etmek mümkündür. Başlangıç popülasyonunun büyüklüğü için herhangi bir standart yoktur, problemin tipine göre planlama operatörleri tarafından belirlenebilir.

Hallah (2007), 3 farklı yöntemden elde ettiği iş sıralamalarını başlangıç popülasyonuna dahil etmiştir. Yöntemlerden ilkinde, optimal olarak çözülmüş birçok probleme ait

işlerin işlem zamanları p_i , teslim tarihleri d_i , gevşeklik $d_i - p_i$ ve buldukları pozisyonlar arasındaki korelasyon hesaplanmıştır. İşlem zamanının $-0,8$ katsayısıyla, gevşekliğin de $0,4$ katsayısıyla baskın oldukları görülmüş ve işlem süresi büyük olan veya erken tamamlanma süresi küçük olan işin ilk sıralarda yer alması gerektiği kural olarak uygulanmıştır. İkinci yöntemde; erken teslim tarihi, en kısa işlem süresi, en uzun işlem süresi (longest processing time – LPT) ve gevşekliğin artan sıralaması şeklinde teknikler uygulanmıştır. Son olarak, dinamik bir atama tekniğiyle erken tamamlanma sürelerini artan sırayla çizelgeleyen üçüncü bir yöntem kullanılmıştır (Muştu, 2013: 54).

2.2 Uygunluk Fonksiyonu

GA yapısı gereği her zaman güçlü ve iyi çözümler vermiş bireylerin yeni nesiller oluşturmasına izin vermektedir. Kaliteli uygunluk değerine sahip bireylerin kullanılması ile kaliteli nesiller üretileceği öngörülmektedir. Bu süreçte çeşitliliğe de önem vererek daha iyi sonuçlar üretebilecek genetik yapıların da kaybolması önlenmelidir. Literatürdeki birçok uygulamada uygunluk fonksiyonu olarak problemin amaç fonksiyonu kullanılmaktadır. Amaç fonksiyonu uygunluk fonksiyonu olarak kullanıldığında tüm bireyler için eşit değerlendirme yapılmış olmaktadır.

Öte yandan literatürde uygunluk fonksiyonu oluştururken farklı yöntemler de kullanılmıştır. Khorshidian vd. (2011), yapmış oldukları çalışmada uygunluk fonksiyonu olarak $M=1000$ sayısından amaç fonksiyonu değerini çıkarmayı yöntem olarak belirlemişlerdir ve çıkan sonucu uygunluk değeri olarak almışlardır.

2.3 Yeni Nesil Seçme Mekanizması

GA'da seçim mekanizması ile bir sonraki nesli oluşturacak bireyler belirlenir. Burada amaç uygunluk fonksiyonu iyi olan bireylerin genlerini bir sonraki nesle aktarmasını sağlamaktır. Yüksek uygunluk değerine sahip bireylerin, çaprazlama ve mutasyon gibi genetik işlemlere girerek yeni nesil oluşturmada kullanılma olasılığı da yüksek olacaktır. Rastgele seçim yöntemi, elitizm yöntemi, rulet çarkı yöntemi, turnuva yöntemi, rank yöntemi ve belirli oranda elitizmle birlikte diğer yöntemlerin beraber kullanıldığı karışık yöntem bu konuda uygulanan birçok yöntemden bazılarıdır.

- **Elitizm Yöntemi:** Popülasyon içinde bulunan en iyi bireyin seçildiği, birden fazla birey seçilecekse uygunluk değerine göre yapılan sıralamada en iyi sonuçları veren bireylerden sırayla en iyilerin seçildiği sistemdir.
- **Rastgele Seçim Yöntemi:** Popülasyon içindeki bireylerden rastgele ve eşit olasılıkla seçim yapıldığı yöntemdir. Aynı bireyin birden fazla seçilmesi istenmiyorsa bazı düzenlemeler yapmak gerekecektir.
- **Rulet Çarkı Seçim Yöntemi:** Genetik algoritmada en sık karşılaşılan seçim yöntemi rulet çarkı seçim yöntemidir. Bu yöntemde, her bir birey için uygunluk fonksiyonu değerleri hesaplanır ve hesaplanan uygunluk fonksiyonu değerleri toplanır. Her bir kromozomun uygunluk fonksiyon sonucu, uygunluk fonksiyonu değerlerinin toplamına bölünerek [0-1] arasında bir değer elde edilir. Bu değerler her bir bireyin çaprazlamaya girme olasılığını gösterir. Bir daire üzerinde her kromozom için bulunan değerle orantılı bir dilim ayrılır. Çaprazlamaya girmesine karar verilen ebeveyn sayısı kadar teker çevrilir ve teker her döndürüldüğünde hangi dilimde durursa, o dilimin temsil ettiği birey seçilir. Rulet çarkı seçim yönteminde en iyi kromozomların seçileceğinin garantisi yoktur. Sadece iyi kromozomların seçilme olasılığı yüksektir (Ak, 2012: 53).
- **Turnuva Seçim Yöntemi:** Popülasyon içerisinde rastgele olarak seçilen n adet bireyden en iyi uygunluk değerine sahip bireyin seçildiği yöntemdir.
- **Karışık Yöntemler:** Yukarıda belirtilen yöntemlerin birlikte kullanılması ile seçim yapılan yöntemdir.

Bu yöntemlerin dışında farklı yöntem ve uygulamalar da mevcuttur.

2.4 Çaprazlama Yöntemleri

Çaprazlamadaki temel amaç, uygunluk değerleri göz önünde bulundurularak belirlenen kurallara göre seçilen ebeveynlerden çocuk bireyler üretmek ve böylece yeni nesillerde uygunluk değerlerinin daha yüksek olmasını sağlamaktır. Burada önemli olan diğer bir nokta da, çaprazlama yapmak için seçilen noktaların çaprazlama sonrasında elde edilecek çocuk kromozomların uygunluk değerlerini ne yönde etkileyeceğidir. Bu işlemi yaparken sonuçları önceden kestirmek mümkün değildir. Rastgele yapılan değişikliklerde sonucun iyi yönde ilerlemesi için belirli kriterler bulmak ve uygulamak gerekir. Bireyler içerisindeki genlerin yapısı ve etkileri araştırılarak, bu genlerde yapılan

değişikliklerle bireye bazı iyi özellikler kazandırılabilir. Çaprazlama sonucunda elde edilecek çocuk kromozomların uygunluk değerinin bir önceki ana kromozomlardan daha yüksek olmama olasılığı da vardır (Ak, 2012: 55-56).

2.4.1 Tek Noktalı Çaprazlama

İlk olarak Holland tarafından geliştirilmiştir ve uygulaması en kolay olan yöntemdir. Kromozom rastgele bir noktadan kesilir ve bu noktadan sonraki genler karşılıklı olarak değiştirilir.

$$\begin{array}{cc|c}
 \text{Ebeveyn 1} & 1\ 0\ 1\ 1 & 0\ 0\ 0\ 1 \\
 \text{Ebeveyn 2} & 0\ 0\ 0\ 1 & 1\ 0\ 1\ 1 \\
 \hline
 \text{Çocuk 1} & 1\ 0\ 1\ 1 & 1\ 0\ 1\ 1 \\
 \text{Çocuk 2} & 0\ 0\ 0\ 1 & 0\ 0\ 0\ 1
 \end{array}$$

Şekil 2.2 Tek noktalı çaprazlama

2.4.2 İki Noktalı Çaprazlama

Birden fazla çaprazlama noktası belirlenir ve bu noktalar arasında karşılıklı gen alışverişi yapılır. Çaprazlama noktasının artırılması GA performansında kötüleşmelere sebep olabilir.

$$\begin{array}{cc|c|c}
 \text{Ebeveyn 1} & 1\ 0 & 1\ 1\ 0\ 0 & 0\ 1 \\
 \text{Ebeveyn 2} & 0\ 0 & 0\ 1\ 1\ 0 & 1\ 1 \\
 \hline
 \text{Çocuk 1} & 1\ 0 & 0\ 1\ 1\ 0 & 0\ 1 \\
 \text{Çocuk 2} & 0\ 0 & 1\ 1\ 0\ 0 & 1\ 1
 \end{array}$$

Şekil 2.3 İki noktalı çaprazlama

2.4.3 OX – LOX – PMX Çaprazlama

OX yönteminde bölünme noktaları arasındaki genler birinci ebeveynden olduğu gibi gelir. İkinci ebeveynin ikinci bölünme noktasındaki genler sırasıyla kontrol edilir ve çocuk bireyde bulunmayan genler çocuk bireyin ikinci noktasından itibaren yazılır.

LOX yönteminin OX'ten farkı; başlangıç noktası olarak kromozomun ilk genin seçilmesidir.

PMX yönteminde çaprazlama yapılan bölgedeki genler ilk ebeveynden alınır, daha sonra ikinci ebeveynin ilk pozisyonundaki gen ilk ebeveynde bulunur ve bulunan bu genin pozisyonuyla aynı pozisyonda bulunan gen (ikinci ebeveynde) çocuk kromozomda yoksa ilk pozisyona atanır. Bu işlem çaprazlama alanı dışında kalan tüm

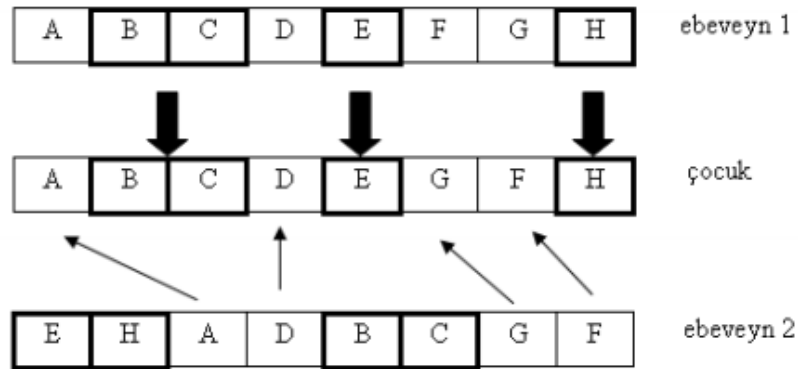
genler için sırayla tekrarlanır. Bu işlem esnasında çocuk kromozomda bulunan genin aynıyla karşılaşırsa, atanmamış bir gen bulunana kadar, ilgili pozisyon için bu işlemler tekrarlanır (Muştu, 2013: 58).

Ebeveyn 1	1	2	3	4	5	6	7
Ebeveyn 2	4	3	7	6	2	5	1
OX	6	2	3	4	5	1	7
LOX	7	6	3	4	5	2	1
PMX	6	7	3	4	5	2	1

Şekil 2.4 İki nokta çaprazlama yöntemi örnekleri (Sevaux ve Peres, 2003)

2.4.4 Pozisyona Dayalı Çaprazlama

Bu yöntemde rastgele seçilen genler birinci ebeveynden çocuğa aynı pozisyonu koruyarak aktarılır. Çocuk bireyde bulunmayan genler ise ikinci ebeveynden sırasıyla çocuk bireyde boş kalan yerlere aktarılır.

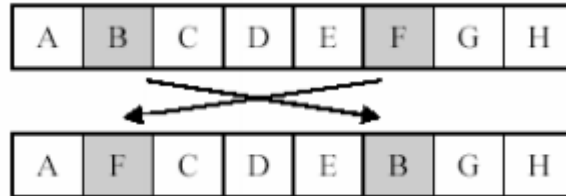


Şekil 2.5 Pozisyona dayalı çaprazlama (Ak, 2012: 58).

2.5 Mutasyon Yöntemleri

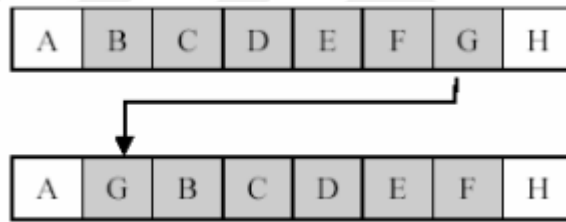
Mutasyon yöntemindeki temel amaç, GA'nın çaprazlamaları yaparken yerel optimumlara takılmasını önlemektir. Popülasyondaki çeşitliliği devam ettiren yöntem mutasyon yöntemidir. Kromozomlardaki bazı yapıları rastgele değiştirerek yeni yapılar oluşmasını sağlar.

- **Komşu İki Geni Değiştirme:** Rastgele seçilen komşu iki gen birbiriyle yer değiştirir.
- **Keyfi İki Geni Değiştirme:** Kromozom üzerinde rastgele seçilen iki gen birbiriyle yer değiştirir.



Şekil 2.6 Rastgele seçilen iki genin yer değiştirmesi

- **Araya Yerleştirme:** Bu işlemden seçilen bir gen rastgele seçilen bir yere yerleştirilir ve diğer genler oluşan boşluğa doğru kaydırılır.



Şekil 2.7 Araya yerleştirme yöntemi

2.6 Algoritmayı Sonlandırma

GA, belirlenen sayıda iterasyona ulaşıncaya sonlandırılır. Bununla birlikte sonlandırma işlemi için farklı kurallar da belirlenebilir.

3. BÖLÜM

TEL ÇEKME BÖLÜMÜNDEKİ ÇİZELGELEME PROBLEMİNİN OPTİMİZASYONU

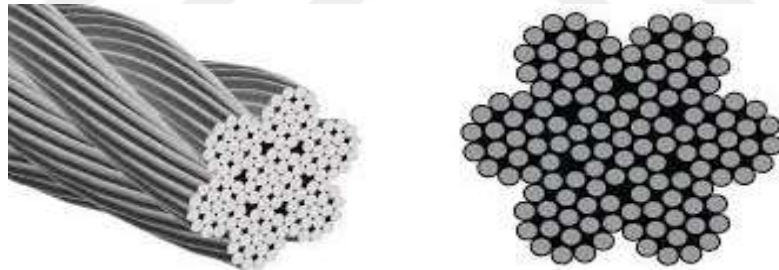
Çizelgeleme problemlerini çözebilmek için zaman içerisinde çok sayıda çözüm yöntemi geliştirilmiştir. Doğrusal programlama, dal sınır algoritması ve kesme metodu gibi algoritmalar deterministik karakterde ve ele alınan problem için en uygun çözümü sağlayabilecek yapıdadır. Kesin çözüm oluşturabilen bu algoritmalar problemin yapısı büyüdükçe ve problem karmaşıklıklaştıkça uygun zaman içerisinde çözüm üretememekte, problem daha da büyük olursa çözüm üretmek eldeki tekniklerle imkânsıza yakınsamaktadır. Bu durum zaman içerisinde sezgisel çözüm yöntemlerini öne çıkarmıştır. Sezgisel yöntemler çok büyük problemlere çok kısa süreler içerisinde tatmin edici sonuçlar verebilmekte, her zaman en iyi sonuca ulaşılmasa bile en iyiye yakın sonuçlar elde edilebilmektedir. Sezgisel çözüm yöntemlerinden biri olan GA'nın çizelgeleme problemlerinde kullanımı ise yapılarının esnek ve çeşitli çizelgeleme problemlerine uyarlanabilir olmalarından dolayı oldukça rahattır. Günümüze kadar yapılan araştırmalarda GA'dan oldukça iyi sonuçlar elde edilmiştir (Ak, 2012: 67).

Literatürde incelenen problemler ve getirilen çözüm önerileri gerçek hayat problemleri ile kıyaslandığında daha sade ve basit olabilmektedir. Gerçek hayatta karşılaşılan problemler daha karmaşık yapıda, karışık ve çözüm üretilebilmesi daha zordur. Bu tezde çözüm önerisi getirilecek problem gerçek hayattan seçilmiştir. Endüstriyel çelik halat imalatı yapmakta olan bir firmanın tel çekme bölümünde sıra bağımlı hazırlık süreleri de kullanılarak üretim çizelgeleme yapılmıştır.

3.1 Problemin Tanımı ve Formülasyonu

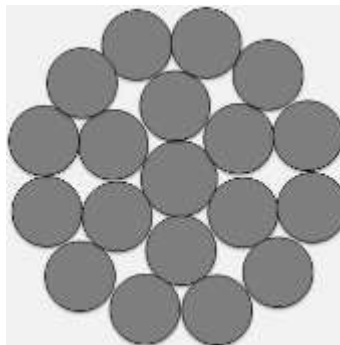
3.1.1 Firmadaki İşlerin Temel Yapısı

Firmada çok telli endüstriyel çelik halat, patentli tavlı teller, endüstriyel yaylık teller, alüminyum iletken çelik özler ve monotronların üretiminde kullanılan çelik tellerin üretimi tel çekme bölümünde yapılmaktadır. Üretim, kalın bir telin soğuk haddeleme yöntemi ile istenen çapa inceltilmesi şeklinde yapılmaktadır. Tel çekme bölümünde diğer tüm bölümlerin talepleri doğrultusunda oluşan işler belirlenen teslim tarihlerine göre planlanmakta ve fabrika içinde gerekli olan yarı mamullerin üretimi sağlanmaktadır. Diğer bölümler; son ürün üretimi yapmakta, müşteri talepleri doğrultusunda ve makine kapasiteleri göz önünde bulundurularak siparişler için son teslim zamanları verilmektedir. Bu işler, üretim esnasında alt dallara ayrılmaktadır ve her alt iş de kendi içinde teslim tarihine göre yeniden dallanmaktadır. Nihayetinde bu işler tel çekme bölümünün üretmesi gereken iş kompozisyonunu oluşturmaktadır. Talep edilen teller tel çekme bölümünde makaralara alınmaktadır. Makara kapasiteleri tellerin hangi uzunluklarda üretileceğini belirlemektedir.



Şekil 3.1 Çok telli çelik halat örneği

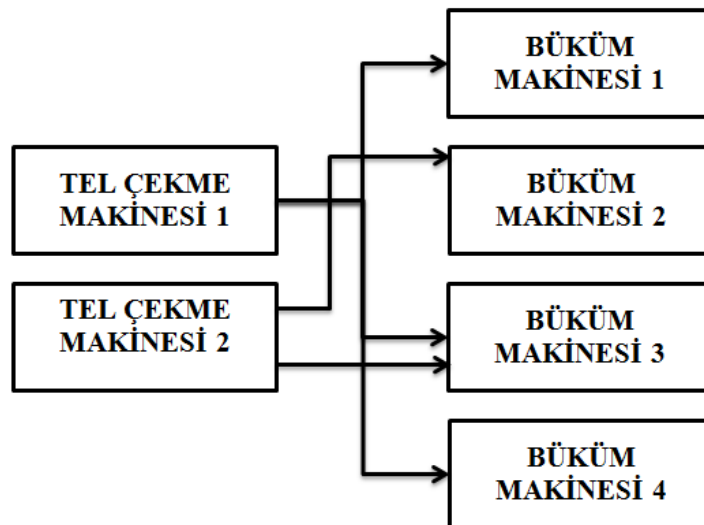
Şekil 3.1 de bir çok telli endüstriyel çelik halat örneği görülmektedir. Örnek olarak verilen bu halat toron olarak adlandırılan 7 ayrı dairesel kesitten oluşmaktadır ve bu dairesel kesitler çok sayıda tel içermektedir.



Şekil 3.2 Halat katmanlarını oluşturan toronun önden görünümü

Şekil 3.2 de örnek olarak ele alınan halatın toronunun önden görünümü verilmiştir. Buradan da görüldüğü toron birbiri üzerine sarılan 3 ayrı katmandan oluşmakta, bu katmanlar farklı sayılarda tel içermekte ve bu tellerin çapları birbirinden farklı olabilmektedir. Şekildeki en üst katmanda 12 adet tel bulunmakta, orta katta 6 adet tel bulunmakta ve merkezde de 1 adet tel bulunmaktadır. Her katman farklı bir iş grubunu temsil etmektedir ve örnek olarak verilen bu toronda 3 farklı iş grubu bulunmaktadır. Her katman için içerdiği tel adedi kadar makara gereksinimi olmaktadır ve istenen tellerin uzunluğu makara kapasitesini aşıyorsa iş bölünmekte ve yeni bir grup alt iş oluşmaktadır. Örneğin üst kat için 12 adet tel gerekmektedir ve bu teller için müşteri siparişine bağlı olarak oluşan tel uzunlukları hesaplanmaktadır. Eğer istenen uzunluklar makara kapasitesini aşıyorsa teller makaraların alabileceği şekilde alt boylara bölünmektedir ve bu bölünen gruplar yeni bir iş grubunu temsil etmektedir. Yeniden örnekleme gerekirse; makara kapasitesini aşan üst kat telleri 12+12 gibi yeniden gruplandırılmaktadır ve 12 makaralı her grup yeni bir işi temsil etmektedir.

Şekil 3.1 ve Şekil 3.2’de verilen örneklerde fabrikada üretilen bir halat referans alınmıştır. Fabrikada farklı konstrüksiyonlarda ve yapılarda çok sayıda halat çeşidi bulunmaktadır. Bu halatların ihtiva ettiği teller bir biri arasında benzerlikler gösterebilmekte ve bu benzerlik oranlarına göre hazırlık süreleri değişkenlik ihtiva etmektedir. Benzer işlerin arka arkaya sıralanması hazırlık sürelerini kısaltmakta, tel çekme makinalarındaki gereksiz işçilik ve makine zamanını azaltmaktadır.



Şekil 3.3 Farika içi iş akış şeması

Şekil 3.3'te fabrika içerisinde oluşan iş akışı şeması görülmektedir. Buradan da anlaşılacağı üzere tel çekme makinaları birden fazla büküm makinasının ihtiyaçlarını karşılayabilmektedir. Ayrıca iki farklı tel çekme makinası eş zamanlı olarak veya ayrı ayrı zamanlarda aynı büküm makinasının ihtiyaçlarını da karşılayabilmektedir. Bir toronu üretebilmek için farklı çaplarda tel çeşitleri istemekte ve bu tel çeşitleri farklı iş olarak tanımlanmaktadır. Fakat bu tellerin teslim zamanları aynı olmakta dolayısıyla farklı tel çekme makinalarında eş zamanlı olarak işlenmeleri gerekmektedir.

ÇOK TELLİ HALAT ÜRETİM İŞ EMRİ							
	İş Emri Çıkış Tarihi			10 Mayıs 2018 Perşembe			
Halat Tipi	6X19 ST			MBL KN	63		
Konstrüksiyonu	6*(1+6+12) 1*(1+6+12)			Toplam Kesit (mm ²)	83		
Halat Turu	NORMAL			Birim Ağırlığı (kg/m)	0,434		
Halat Çapı	10,00			Galvaniz Sınıfı:	GLV-B		
Kaplama Türü	GALVANİZ			Müşteri:			
Mukavemet Sınıfı	180			Birleştirilmiş Sipariş No			
	60006182						
Üretim iş emri no	2033370	2033371	2033372	2033374	2033375	2033376	
Malzeme	T1-1	T1-2	T1-3	T2-1	T2-2	T2-3	
Torondaki Tel Sayısı	1	6	12	1	6	12	
Tel Çapı (mm)	0,85	0,70	0,70	0,95	0,80	0,80	
Kullanılacak Sepet (Giriş Tel Çapı)	1,76/75/ TVG	1,76/75/ TVG	1,76/75/ TVG	1,89/75/ TVG	1,89/75/ TVG	1,89/75/ TVG	
Makara adedi	10	126	252	1	12	24	
Tel Boyu (metre)	34540	18334	18421	18240	12907	12968	
Tel Çekme Makinesi	T116	T115	T115	T116	T116	T116	
Makara Tipi	400-S DEMİR	315-S DEMİR	315-S DEMİR	400-S DEMİR	315-S DEMİR	315-S DEMİR	
Makara Kapasitesi (kg)	150	60	60	150	60	60	

Şekil 3.4 Örnek bir çok telli halat üretim iş emri

Şekil 3.4 örnek bir çok telli halat üretim iş emrini göstermektedir. Üretim iş emri no satırına bakıldığında 6 farklı iş emri numarası görülmektedir ve bunlar 6 farklı işi tanımlamaktadır. 2033371 numaralı iş emri sütununa ve makara adedi satırına

bakıldığında 126 sayısı, tel boyu satırına bakıldığında ise 18334 sayısı görülmektedir. Bu rakamlar bu işten 18334 metre boyunda 126 adet üretim yapılacağı manasına gelmektedir. Aynı iş emrinin torondaki tel sayısı satırına bakıldığında ise 6 rakamı görülmektedir. Bu rakam ise bir adet toron üretebilmek için bu telden aynı anda 6 adet boyun hazır bulunması gerektiğini göstermektedir. Bu durum bize bu işi altılı gruplara bölerek farklı işler gibi düşünebilme imkânı vermektedir ve her grubun teslim zamanı farklı olmaktadır. $126/6=21$ dediğimizde ise 2033371 numaralı işin altında 21 adet farklı iş barındırdığını görmekteyiz. Bu şekilde tüm iş emri için bir hesap yapıldığında 57 adet farklı teslim zamanlı iş ortaya çıkmaktadır.

Ayrıca 2033371 numaralı ve 2033372 numaralı işlerin tel çapı ve kullanılacak sepet satırlarına bakıldığında aynı değerler olduğunu görmekteyiz. Bu durum bize bu işlerin aslında aralarında hazırlık süresi olmayan benzer işler olduğunu göstermektedir. Bu işlerin diğer işlerle ilişkisi aralarındaki benzerlik durumuna göre hesaplanmaktadır. Tel çekme makinası satırına bakıldığında ise tüm işlerin iki farklı tel çekme makinasına atandığını görmekteyiz. Bu iş emrini üretebilmek için iki farklı tel çekme makinası gerektiği için bu makinalardan birinin üzerindeki işi erken üretmesi bir şey ifade etmemekte diğer makinadan çıkacak teller beklenmek durumunda kalmaktadır ve ara stoklar oluşmaktadır. Ayrıca makinalardan birinin üzerindeki işi geciktirmesi, halatın üretilmemesine sebep olmakta ve müşteri siparişini geciktirmektedir. Ayrıca tel çekme makinalarından birinin gecikmesi diğer makinadan çıkan tellerin de ara stokta kalmasına sebep olmaktadır.

İşletmenin tel çekme bölümünde hali hazırda merkezi planlama yapılmamaktadır. Diğer bölümlerin talepleri doğrultusunda üretim esnasında anlık planlamalar yapılmaktadır. Bu durum ise tel çekme bölümünün beslediği makinalarda gecikmelere sebep olmakta, erken üretimlerle gereksiz ara stoklar oluşmakta ve tel çekme makinelerinde uygun işler arka arkaya sıralanmayarak hazırlık sürelerinin artmasına sebep olunmaktadır. Artan hazırlık süreleri ise iş gücünde ve makine saatinde kayıplar oluşturmaktadır.

3.1.2 Problemin Analizi

Analizi yapılacak tel çekme bölümünde toplam 13 adet tel çekme makinesi bulunmaktadır. Bu makinaların her biri kendi içerisinde değerlendirilecektir ve tek makine üretim çizelgeleme yapılacaktır. Problemin çözümü için gerekli olan temel kabuller aşağıdaki gibi olacaktır.

- Problem deterministik bir model olarak düşünölmüştür.
- Çizelgelemeye başlamadan önce tüm işler teslim zamanları belli bir şekilde hazır bulunmaktadır.
- Erken üretim, yarı mamul ara stokuna sebep olacaktır ve bir stok maliyeti oluşturacaktır.
- Geç üretim bir sonraki iş istasyonun beklemesine sebep olacaktır. Bu durum müşteri siparişlerini de geciktirecektir.
- İşler arasında sıraya bağılı olarak değışken hazırlık süreleri oluşmaktadır.
- Hazırlık süreleri tel çekme makinalarının boş kalmasına sebep olmaktadır ve fırsat maliyetine sebep olmaktadır.
- İşler çizelgeleme yapmadan önce makineye atanacaktır ve makine tam kapasite doldurulacaktır.
- Tel çekme makinelerinde fazla yükleme veya boş zaman oluşturulmayacaktır.
- İş istasyonlarında arıza söz konusu olmayacak ve belirtilen kapasiteler doğrultusunda çalışılacaktır.
- Kalite problemleri oluşursa yeniden işleme ya da yeniden üretim yapılmayacağı farz edilmiştir.
- İlk sıraya gelecek işte başlangıç hazırlık süresinin olmayacağı kabul edilmiştir.

3.2 Problem İçin Matematiksel Model

Problemde tek bir makine fakat çok sayıda iş bulunmaktadır. Bu tarz problemlerin doğrusal programlama ile oluşturulacak matematiksel modeller ile çözölməsi imkânsıza yakınsamaktadır. Deneme yapmak amacıyla uygun bir matematiksel modelle problemin basit bir bölümüne çözüm üretilecektir.

Parametreler

n : İş sayısı

p_j : j işinin işlem süresi

d_j : j işinin teslim zamanı

h_j : ilk sıradaki işin sıra bağımlı hazırlık süresi (başlangıç hazırlık süresi)

s_{ij} : i işi j işinden önce sıralandığında j işinin sıra bağımlı hazırlık süresi

M : çok büyük bir pozitif tamsayı

Karar Değışkenleri

y_{jk} : j işi k . sıraya atanırsa 1; diğıer durumda 0

z_{ijk} : j işi k . sıraya atanırsa ve i işinden hemen sonra yapılacaksa 1; diğer durumda 0

C_j : j işinin tamamlanma zamanı

C_{max} : son işin tamamlanma zamanı

T_j : j işinin gecikmesi

Amaç Fonksiyonu

$$\text{Min } C_{max} \quad (1)$$

Kısıtlar

$$C_j + M(1 - y_{jk}) \geq h_j + p_j \quad \forall j, k=1, j, k \in N \quad (2)$$

$$C_j - C_i + M(1 - z_{ijk}) \geq s_{ij} + p_j \quad \forall i \neq j, k > 1, i, j, k \in N \quad (3)$$

$$1 + z_{ijk} \geq y_{ik-1} + y_{jk} \quad \forall i \neq j, k > 1, i, j, k \in N \quad (4)$$

$$\sum_{j=1}^n y_{jk} = 1 \quad \forall k \in N \quad (5)$$

$$\sum_{k=1}^n y_{jk} = 1 \quad \forall j \in N \quad (6)$$

$$T_j \geq C_j - d_j \quad \forall j \in N \quad (7)$$

$$C_{max} \geq C_j \quad \forall i \neq j, k=n \in N \quad (8)$$

$$y_{jk} \in \{0, 1\} \quad \forall j, k \in N \quad (9)$$

$$z_{ijk} \in \{0, 1\} \quad \forall i \neq j, k \in N \quad (10)$$

$$C_j \geq 0 \quad \forall j \in N \quad (11)$$

$$C_{max} \geq 0 \quad (12)$$

$$T_j \geq 0 \quad \forall j \in N \quad (13)$$

1 numaralı ifade amaç fonksiyonunu göstermektedir ve toplam süreyi minimize etmektedir. 2 numaralı kısıt birinci sıraya gelecek işin ve 3 numaralı kısıt k . ($k > 1$) sıradaki işin tamamlanma zamanını belirtmektedir. Kısıt 4, z_{ijk} karar değişkeninin 1 veya 0 değerini almasını sağlamaktadır. Kısıt 5 ve 6 sırasıyla her konuma yalnızca bir işin atanmasını ve bir işin yalnızca bir konuma atanmasını sağlamaktadır. 7 numaralı kısıt, j işinin gecikmesini ve 8 numaralı kısıt son işin tamamlanma zamanını belirlemektedir. Kısıt 9, 10, 11, 12, 13 karar değişkenlerine ait işaret kısıtlarıdır.

Matematiksel model de toplam üretim zamanı minimize edilecektir. Tüm giriş değerleri matris halinde programa tanıtılacak ve bu değerler göz önünde bulundurularak hesaplama yaptırılacaktır. Hesaplama algoritmaları MATLAB paket programının alt yapısı kullanılarak oluşturulmuştur.

3.2.1 Matematiksel Modelde Örnek Uygulamalar

Daha önce belirtildiği gibi matematiksel modelle basit bir iş modeli oluşturularak çözüm üretilecektir. Modelin amaç fonksiyonu toplam zamanın minimize edilmesi üzerine kurgulanmıştır. Elde edilmesi gereken sonucun önceden bilinebilmesi için, iş modeli yönlendirilmiş verilerden oluşacaktır. Bu şekilde eldeki modelle uygun çözümler üretilip üretilemeyeceği ve bu çözümlerin ne kadar zamanda üretilebildiği test edilecektir. İlk olarak 10 işten oluşacak bir veri seti denecektir.

Tablo 3.1 Sıra bağımlı hazırlık sürelerinin gösterimi (dakika)

İş Numarası	1	2	3	4	5	6	7	8	9	10
1	--	0	100	100	100	100	100	100	100	100
2	100	--	0	100	100	100	100	100	100	100
3	100	100	--	0	100	100	100	100	100	100
4	100	100	100	--	0	100	100	100	100	100
5	100	100	100	100	--	0	100	100	100	100
6	100	100	100	100	100	--	0	100	100	100
7	100	100	100	100	100	100	--	0	100	100
8	100	100	100	100	100	100	100	--	0	100
9	100	100	100	100	100	100	100	100	--	0
10	100	100	100	100	100	100	100	100	100	--

Tablo 3.1'de bir işten başka bir işe geçildiğinde oluşacak hazırlık süreleri gösterilmektedir.

Tablo 3.2 Eldeki işler için üretim sürelerinin gösterimi (dakika)

İş Numarası	1	2	3	4	5	6	7	8	9	10
Üretim Süreleri	10	10	10	10	10	10	10	10	10	10

Tablo 3.2 de her iş için hesaplanmış olan üretim süreleri gösterilmektedir.

Tablo 3.3 Eldeki işler için başlangıç hazırlık sürelerinin gösterimi (dakika)

İş Numarası	1	2	3	4	5	6	7	8	9	10
Başlangıç Hazırlık Süresi	0	100	100	100	100	100	100	100	100	100

Tablo 3.3'de veri setindeki işlerden biri ilk sırada çizelgelenirse oluşacak hazırlık süreleri verilmektedir.

Tüm bu yönlendirilmiş giriş verileri değerlendirildiğinde optimum sıralamanın 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 şeklinde olacağı ve toplam üretim zamanının ise 100 dakika

olacağı görülmektedir. Oluşturulan matematiksel model, eldeki giriş verileri kullanılarak MATLAB paket programında çalıştırıldığında beklenen sonuç elde edilmiştir. İş sıralaması istediğimiz şekilde gerçekleşmiş, toplam tamamlanma zamanı 100 birim olacak şekilde optimum sonuç elde edilmiştir. Bu sonuca ulaşabilmek için program 142,4 saniye çalışmıştır.

Bu denemenin akabinde ise 15 işten oluşan bir veri seti denenecektir. Bu veri setinde yine yönlendirilmiş verilerden oluşacaktır ve optimum çözüm önceden bilinecektir.

Tablo 3.4 Sıra bağımlı hazırlık sürelerinin gösterimi (dakika)

İş Numarası	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	--	0	100	100	100	100	100	100	100	100	100	100	100	100	100
2	100	--	0	100	100	100	100	100	100	100	100	100	100	100	100
3	100	100	--	0	100	100	100	100	100	100	100	100	100	100	100
4	100	100	100	--	0	100	100	100	100	100	100	100	100	100	100
5	100	100	100	100	--	0	100	100	100	100	100	100	100	100	100
6	100	100	100	100	100	--	0	100	100	100	100	100	100	100	100
7	100	100	100	100	100	100	--	0	100	100	100	100	100	100	100
8	100	100	100	100	100	100	100	--	0	100	100	100	100	100	100
9	100	100	100	100	100	100	100	100	--	0	100	100	100	100	100
10	100	100	100	100	100	100	100	100	100	--	0	100	100	100	100
11	100	100	100	100	100	100	100	100	100	100	--	0	100	100	100
12	100	100	100	100	100	100	100	100	100	100	100	--	0	100	100
13	100	100	100	100	100	100	100	100	100	100	100	100	--	0	100
14	100	100	100	100	100	100	100	100	100	100	100	100	100	--	0
15	100	100	100	100	100	100	100	100	100	100	100	100	100	100	--

Tablo 3.4'te bir işten başka bir işe geçildiğinde oluşacak hazırlık süreleri gösterilmektedir.

Tablo 3.5 Eldeki işler için üretim sürelerinin gösterimi (dakika)

İş Numarası	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Üretim Süreleri	10	10	10	10	10	10	10	10	10	10	10	10	10	10	10

Tablo 3.5 de her iş için hesaplanmış olan üretim süreleri gösterilmektedir.

Tablo 3.6 Eldeki işler için başlangıç hazırlık sürelerinin gösterimi (dakika)

İş Numarası	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Başlangıç Hazırlık Süresi	0	100	100	100	100	100	100	100	100	100	100	100	100	100	100

Tablo 3.6’da veri setindeki işlerden biri ilk sırada çizelgelenirse oluşacak hazırlık süreleri verilmektedir.

Bir önceki denemede olduğu gibi bu yönlendirilmiş giriş verileri değerlendirildiğinde de optimum sıralamanın 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 şeklinde olacağı ve toplam üretim zamanının ise 150 dakika olacağı görülmektedir. Oluşturulan matematiksel model, eldeki giriş verileri kullanılarak MATLAB paket programında 7200 saniye çalıştırılmıştır. Bu sürenin sonunda beklenen en iyi çözüm elde edilememiştir ve toplam üretim zamanının 450 olduğu bir sıra oluşturulmuştur.

Çalışmalarda 10 işten oluşan ve 15 işten oluşan iki ayrı yönlendirilmiş veri seti denenmiştir. Denemeler 2,40 GHz işlemci hızına ve 6,00 GB RAM kapasitesine sahip bir bilgisayarda gerçekleştirilmiştir. Çizelgeleme için kullanılan 10 işe çözüm üretebilmek için 142,4 saniyelik bilgisayar zamanına ihtiyaç varken, iş miktarını sadece 5 birim artırarak 15 birime çıkarıldığında, bilgisayar 7200 saniye gibi yüksek bir süre çalışmış olmasına rağmen en iyi sonuca ulaşamamıştır. Bu durum bize iş sayısındaki küçük artışların bile doğrusal programlama ile çözüm üretebilmek için gerekli olan zamanın katlanarak arttığını göstermektedir. Bilgisayar zamanı ihtiyacının katlanarak artması, mevcut teknolojik gelişmeler ile doğrusal programlamada oluşturulan problemlerin çözümünü imkânsızlaştırmaktadır. Bu tezde, çözüm üretmede ortaya çıkan zorlukları aşmak için sezgisel yöntemlerden biri olan GA kullanılacaktır. Bu şekilde ele alınan karmaşık yapıdaki problemlere optimuma yakın veya kabul edilebilir çözümler üretmeye çalışılacaktır.

3.2.2 Mevcut Problem İçin Matematiksel Model Gösterimi

Deneme amacı ile kullanılan matematiksel modelden sonra ele alınan problem için kurgulanan amaç fonksiyonu ve bu problemin çözümünde kullanılacak matematiksel modelin gösterimi aşağıda yapılmıştır.

Parametreler

n : İş sayısı

p_j : j işinin işlem süresi

d_j : j işinin teslim zamanı

s_{ij} : i işi j işinden önce sıralandığında j işinin sıra bağımlı hazırlık süresi

α_j : j işi için erken bitirme maliyet değişkeni

β : geç bitirmelerin maliyet katsayısı

μ : hazırlık süreleri için maliyet katsayısı

M : çok büyük bir pozitif tamsayı

Karar Değişkenleri

y_{jk} : j işi k . sıraya atanırsa 1; diğer durumda 0

z_{ijk} : j işi k . sıraya atanırsa ve i işinden hemen sonra yapılacaksa 1; diğer durumda 0

C_j : j işinin tamamlanma zamanı

C_{max} : son işin tamamlanma zamanı

T_j : j işinin gecikme süresi

E_j : j işinin eken tamamlanma süresi

s_j : j işinin hazırlık süresi

Amaç Fonksiyonu

$$\text{Min} \sum_{j=1}^n (\alpha_j E_j^2 + \beta T_j + \mu s_j) \quad (1)$$

Kısıtlar

$$C_j + M(1 - y_{jk}) \geq p_j \quad \forall j, k=1, j, k \in N \quad (2)$$

$$C_j - C_i + M(1 - z_{ijk}) \geq s_{ij} + p_j \quad \forall i \neq j, k > 1, i, j, k \in N \quad (3)$$

$$1 + z_{ijk} \geq y_{ik-1} + y_{jk} \quad \forall i \neq j, k > 1, i, j, k \in N \quad (4)$$

$$\sum_{j=1}^n y_{jk} = 1 \quad \forall k \in N \quad (5)$$

$$\sum_{k=1}^n y_{jk} = 1 \quad \forall j \in N \quad (6)$$

$$T_j - E_j \geq C_j - d_j \quad \forall j \in N \quad (7)$$

$$s_j = \sum_{i=1}^n \sum_{k=2}^n (z_{ijk} \times s_{ij}) \quad \forall i \neq j, k > 1, i, j, k \in N \quad (8)$$

$$y_{jk} \in \{0, 1\} \quad \forall j, k \in N \quad (9)$$

$$z_{ijk} \in \{0, 1\} \quad \forall i \neq j, k \in N \quad (10)$$

$$C_j \geq 0 \quad \forall j \in N \quad (11)$$

$$T_j \geq 0 \quad \forall j \in N \quad (12)$$

$$E_j \geq 0 \quad \forall j \in N \quad (13)$$

$$s_j \geq 0 \quad \forall j \in N \quad (14)$$

1 numaralı ifade, amaç fonksiyonunu göstermektedir. Gecikmeleri, erken bitirmeleri ve hazırlık sürelerini minimize etmektedir. 2 numaralı kısıt, birinci sıraya gelecek işin ve 3 numaralı kısıt k . ($k > 1$) sıradaki işin tamamlana zamanını belirtmektedir. Kısıt 4, z_{ijk} karar değişkeninin 1 veya 0 değerini almasını sağlamaktadır. Kısıt 5 ve 6 sırasıyla her konuma yalnızca bir işin atanmasını ve bir işin yalnızca bir konuma atanmasını sağlamaktadır. 7 numaralı kısıt j işinin gecikmesini veya erken tamamlanmasını belirtmektedir. 8 numaralı kısıt ise j işinin hazırlık süresini hesaplamaktadır. Kısıt 9, 10, 11, 12, 13, 14 karar değişkenlerine ait işaret kısıtlarıdır.

3.3 Problemden Kullanılacak Temel Veriler ve Gösterimleri

Problemin çözümünde kullanılacak temel veriler tablolar halinde aşağıda verilmiştir.

Tablo 3.7’de işin temel karakteristiklerini belirten veriler bulunmaktadır.

Tablo 3.7 Bir numaralı temel veriler tablosu

İş Sırası	Halat Çapı (mm)	Kaplama İndisi	Halat Konstrüksiyon İndisi	Giriş Tel Çapı (mm)	Çıkış Tel Çapı (mm)
1	18	1	4	1,55	0,65
2	10	1	6	2,01	0,76
3	18	1	3	2,01	0,84
4	18	1	2	2,01	0,86
5	10	1	6	2,01	0,76
6	18	1	3	2,01	0,84
7	18	1	2	2,01	0,86
8	10	1	6	2,01	0,76
9	10	1	6	2,01	0,76
10	22	1	4	1,89	0,79
11	10	2	4	1,37	0,36
12	10	1	6	2,01	0,76
13	10	1	6	2,01	0,76

Tablo 3.8’de işin miktarı, kullanılacak makara adetleri, makine hızları, son teslim tarihleri ve zaman penceresi katsayıları verilmiştir.

Tablo 3.8 İki numaralı temel veriler tablosu

Tel Uzunluğu (metre)	Makara Adedi	Tel Çekme Hızı (m/s)	Son Teslim Zamanı (saat)	Artış Oranı
41260	7	12	72	1,72
17700	1	14	72	5,47
20460	7	12	72	1,84
20270	7	12	72	1,81
17700	1	14	87	5,47
20460	7	12	93	1,84
20270	7	12	93	1,81
17700	1	14	102	5,47
17700	1	14	117	5,47
41260	7	12	117	1,47
54700	7	14	132	2,31
17700	1	14	132	5,47
17700	1	14	147	5,47

Tablo 3.9’da işler arasında sıraya bağımlı olarak oluşmakta olan hazırlık sürelerini gösteren örnek bir tablo verilmiştir.

Tablo 3.9 Sıra bağımlı hazırlık süreleri (Saat)

İş Numarası	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	--	2,1	1,95	1,8	2,1	1,95	1,8	2,1	2,1	1,85	3,3	2,1	2,1	1,7	1,55
2	1,85	--	1,45	1,3	0	1,45	1,3	0	0	1,85	3,3	0	0	1,7	1,55
3	1,85	1,6	--	0,5	1,6	0	0,5	1,6	1,6	1,85	3,3	1,6	1,6	1,7	1,55
4	1,85	1,6	0,5	--	1,6	0,5	0	1,6	1,6	1,85	3,3	1,6	1,6	1,7	1,55
5	1,85	0	1,45	1,3	--	1,45	1,3	0	0	1,85	3,3	0	0	1,7	1,55
6	1,85	1,6	0	0,5	1,6	--	0,5	1,6	1,6	1,85	3,3	1,6	1,6	1,7	1,55
7	1,85	1,6	0,5	0	1,6	0,5	--	1,6	1,6	1,85	3,3	1,6	1,6	1,7	1,55
8	1,85	0	1,45	1,3	0	1,45	1,3	--	0	1,85	3,3	0	0	1,7	1,55
9	1,85	0	1,45	1,3	0	1,45	1,3	0	--	1,85	3,3	0	0	1,7	1,55
10	1,85	2,1	1,95	1,8	2,1	1,95	1,8	2,1	2,1	--	3,3	2,1	2,1	1,7	1,55
11	2,35	2,6	2,45	2,3	2,6	2,45	2,3	2,6	2,6	2,35	--	2,6	2,6	2,2	2,05
12	1,85	0	1,45	1,3	0	1,45	1,3	0	0	1,85	3,3	--	0	1,7	1,55
13	1,85	0	1,45	1,3	0	1,45	1,3	0	0	1,85	3,3	0	--	1,7	1,55
14	1,85	2,1	1,95	1,8	2,1	1,95	1,8	2,1	2,1	1,85	3,3	2,1	2,1	--	0,4
15	1,85	2,1	1,95	1,8	2,1	1,95	1,8	2,1	2,1	1,85	3,3	2,1	2,1	0,3	--

3.4 Problem İçin Genetik Algoritma

GA, doğada gözlemlenen çaprazlama ve mutasyon gibi genetik süreçleri taklit eden bir arama ve eniyileme yöntemidir. Çok boyutlu arama uzayında en iyilerin hayatta kalması sağlanarak en iyi çözüm aranır. Bunu yaparken de GA, birtakım yöntem ve operatörleri kullanır (Doğan 2013: 38).

Ele alınan probleme çözüm oluşturmak için ağırlıkları farklı olan üç ayrı amaç fonksiyonu kurgulanmıştır ve bu amaç fonksiyonlarının eş zamanlı minimize edilmesi planlanmıştır. Ağırlıkları farklı olmakla birlikte amaç fonksiyonları aşağıdaki gibidir.

- Sıra bağımlı hazırlık sürelerini minimize etmek.
- İşlerin erken bitirilme zamanlarını minimize etmek.
- İşlerin geç bitirilme zamanlarını minimize etmek.

$$f(s) = \sum_{j=1}^n (\alpha_j E_j^2 + \beta T_j + \mu s_j)$$

3.4.1 Genetik Algoritma ve Çözüm Metodolojisinin Adımları

Ele alınan problemin çözümünde kullanılacak GA'nın çizelge elde edebilmesi için izleyeceği adımlar şu şekilde özetlenebilir.

- İşlere ait temel verilerden olan o işte kullanılacak makara adetleri, makaralara alınacak tel uzunlukları, o işin tel çekme makinesindeki üretim hızı ve işlerin son teslim zamanları tablo olarak verilecektir. Ayrıca iş erken üretildiğinde elde bulundurma maliyetiyle ilişkili bir oran olan Artış Oranı verisi de tablo olarak verilecektir.
- Sıra bağımlı hazırlık sürelerini hesaplayabilmek için gerekli veriler olan halat çapı, giriş teli kaplama indisi, halat konstrüksiyon indisi, tel çekme makinasında inceltilecek olan telin çapı ve diğer bölümlerin talep ettiği inceltilmiş tel çapı tablo halinde verilecektir.
- İşler arasındaki ilişkiye göre değişen hazırlık süreleri programa entegre alt bir program aracılığıyla temel veriler kullanılarak hesaplanacak, matris haline getirilecek ve GA tarafından kullanılacaktır.

- İşler için gerekli olan üretim süreleri ve üretilecek ürünlerin kilogram cinsinden miktarları tel çekme makinesi üretim hızları ve istenen tel uzunlukları kullanılarak hesaplanacaktır ve GA tarafından kullanılacaktır.
- GA işini kolaylaştırmak için başlangıç çözümleri başka sezgisel algoritmalar ile hesaplatılacaktır ve GA ilk çözüm olarak bunları kullanacaktır.
- Popülasyon sayısı 20 mutasyon oranı 0,005 ve iterasyon sayısı da 100000 olarak belirlenmiştir.
- Üretilen kromozomlar içerisinde en iyi çözümü veren ilk dört adedi elde tutulacak, en iyi ilk dört adet kromozom ebeveyn olarak kullanılacak ve çocuk bireyler elde edilecektir. Elde tutulan ebeveynlerle sekiz adet çaprazlama yapılacak ve toplamda 16 adet çocuk birey elde edilecektir.
- Çaprazlanacak iş miktarı iterasyonun başında toplam işin %20 si kadar olacak, iterasyon sayısı arttıkça, çaprazlanacak iş miktarı azalacak ve programın sonunda %1,5 gibi bir değere gelecektir. Bu sayede, programın başlarında daha hızlı aramalar yapılırken sonlara doğru daha hassas aramalar amaçlanmaktadır.
- Kromozomlar üzerinde çaprazlanacak bölümler rassal olarak belirlenecek ve bu bölümler değiştirilecektir.
- Çaprazlama sonrasında kromozomlar üzerinde aynı işlerden iki kere yazılma durumu oluşacaktır. Bunu düzeltmek için çaprazlamaya tabi tutulmayan bölümler arasında kontrol yapılacak ve aynı işler değiştirilerek mükerrer işler engellenecektir.
- Yapılacak mutasyon miktarı; popülasyon sayısı, iş sayısı ve mutasyon oranlarının çarpımı ile elde edilecektir.
- Çaprazlama dışı tutulan işlerden en iyi olanı konumunda mutasyon dışı tutulacak, bu iş yirminci sıradaki işle değiştirilerek burada mutasyona tabi tutulacaktır.
- Mutasyon yapılacak birey rastgele seçilecek ve bu birey üzerinde yine rastgele seçilen iki genin yer değiştirmesi ile mutasyon gerçekleştirilecektir.
- Bir iterasyon içerisinde aynı bireyin birden fazla kez mutasyona tabi tutulması olasılıkları içerisindedir.

Tüm bu hesaplar MATLAB programında yazılan algoritmalar ile yapılacaktır. İşlemler bittikten sonra maliyet fonksiyonu çalıştırılacak ve sonuçlara göre en iyi sıralama yapılacak ve işlemler tekrarlanacaktır.

3.4.2 Çaprazlama Operatörü

Çaprazlama operatörünün mantığı Şekil 3.5'te gösterilmiştir. Çaprazlama rastgele seçilen iki bölge arasında yapılacaktır. Çaprazlanacak iş miktarı ise iterasyonlar ilerledikçe düşecektir. Programın başında toplam iş sayısının %20'si gibi bir iş miktarı çaprazlanacakken bu oran programın son iterasyonunda %1,5 gibi küçük bir orana düşecektir. Buradaki temel mantık, program ilerledikçe ortaya çıkan iyi bireylere daha az müdahale ederek bozulmaları engellemektir. Her iş tek tek yer değiştirecek ve her işin yer değiştirmesinden sonra kromozomlarda oluşan aynı işler de yer değiştirerek düzeltme yapılacaktır.

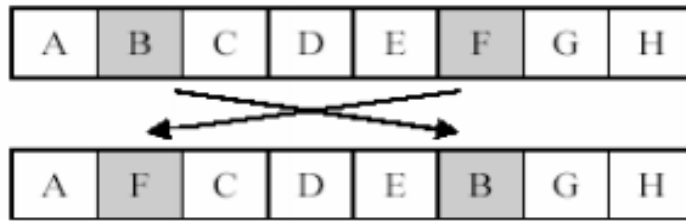
	Çaprazlanacak Bölgeler														
Ebeveyn 1	12	13	7	4	6	14	3	11	1	9	2	8	15	10	5
Ebeveyn 2	8	1	9	3	14	2	15	4	11	10	13	5	12	7	6
1. Çaprazlama	12	13	7	4	6	14	3	11	1	9	2	8	15	10	5
	8	1	9	3	14	2	15	4	11	10	13	5	12	7	6
1. Düzeltme	12	13	7	11	6	14	3	11	1	9	2	8	15	10	5
	8	1	9	3	14	2	15	4	4	10	13	5	12	7	6
2. Çaprazlama	12	13	7	11	6	14	3	4	1	9	2	8	15	10	5
	8	1	9	3	14	2	15	11	4	10	13	5	12	7	6
2. Düzeltme	12	13	7	11	10	14	3	4	1	9	2	8	15	10	5
	8	1	9	3	14	2	15	11	4	6	13	5	12	7	6
3. Çaprazlama	12	13	7	11	10	14	3	4	1	9	2	8	15	6	5
	8	1	9	3	14	2	15	11	4	6	13	5	12	7	10
3. Düzeltme	12	13	7	11	10	13	3	4	1	9	2	8	15	6	5
	8	1	9	3	14	2	15	11	4	6	14	5	12	7	10
Çocuk 1	12	14	7	11	10	13	3	4	1	9	2	8	15	6	5
Çocuk 2	8	1	9	3	13	2	15	11	4	6	14	5	12	7	10

Şekil 3.5 Çaprazlama operatörü gösterimi

Çaprazlama operatörünün temel kurgusu tek nokta çaprazlama ve iki nokta çaprazlama yöntemlerini temel olarak oluşturulmuştur. Çaprazlama yapmak için aralık belirlemesi bakımından iki noktalı çaprazlama yöntemine, bu belirlenen aralıktaki genleri tek tek yer değiştirmesi bakımından ise tek nokta çaprazlama yöntemine benzemektedir.

3.4.3 Mutasyon Operatörü

Mutasyon yöntemi olarak keyfi iki genin yer değiştirmesi yöntemi kullanılacaktır. Şekil 3.6'da gösterildiği gibi rastgele seçilen iki iş bir biri ile yer değiştirecektir. Kurulan modelde aynı işte birden fazla kez mutasyon yapılma durumu oluşabilmektedir.



Şekil 3.6 Keyfi iki genin yer değiştirmesi yöntemi

3.4.4 Başlangıç Popülasyonu İçin Sezgisel Yöntemler ve Adımları

GA'ya rassal olarak üretilen popülasyonlardan daha iyi başlangıç çözümleri üretmek ve programa hız katmak amacıyla dört adet sezgisel algoritma geliştirilmiştir. Bu bölümde sezgisel algoritmaların çalışma mantıkları ve işlem adımları anlatılacaktır.

Sezgisel algoritmalarda temel mantık olarak zaman penceresi oluşturma yöntemi kullanılacaktır. İşlerin teslim tarihleri sabit kalmak kaydıyla, başlama zamanları öne alınarak zaman pencereleri oluşturulacaktır. Zaman pencereleri her iterasyonda genişleyen bir yapıda olacaktır ve kullanıcının belirlediği miktara kadar genişleyecektir. Bir sezgiselde tüm işler için sabit zaman penceresi oluşturulacaktır. Diğer sezgiselerde ise Tablo 3.8'de verilen *artış oranları* kullanılarak her iş için kendi ağırlıklarına göre zaman pencereleri oluşturulacaktır. Artış oranları, işlerin kilogram cinsinden miktarlarına göre hesaplanan elde bulundurma maliyetiyle ters orantılı bir değerdir. İşlerin miktarları arttıkça artış oranları düşmektedir. Sezgiseller bu şekilde açılan zaman pencereleri içinde, son teslim tarihine göre sıralanarak sisteme verilmiş işler içerisinde benzer işleri ya da en düşük maliyetli işleri birbirinin arkasına ekleyerek yeni

popülasyonlar oluşturacaktır. Devam eden başlıklarda geliştirilen sezgisellerin işlem adımları anlatılacaktır.

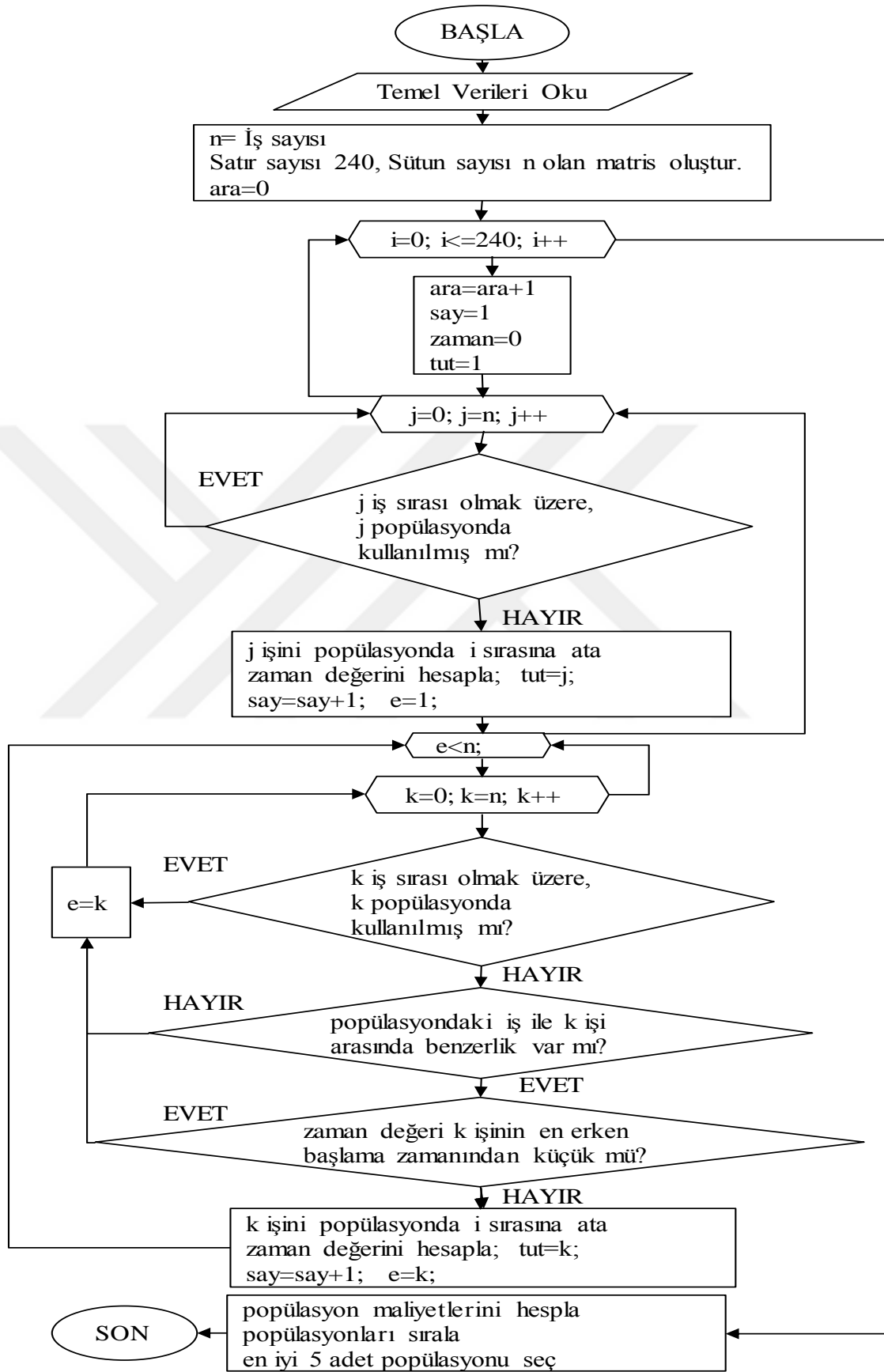
3.4.4.1 Bir Numaralı Sezgisel Algoritma (S1)

Bir numaralı sezgiselin işlem adımları aşağıdaki gibidir. İşler tabloda verilen sıra numaralarıyla takip edilecektir. Zaman çizelgesi oluşturmak için işlerin teslim tarihinden o an için üretilen değer çıkartılacak ve işlerin en erken başlama zamanı hesaplanacaktır.

- **Adım 1:** n iş sayısı olmak üzere 240 adet satırlı n adet sütunlu tüm elemanları 0 olan bir matris oluştur ve ikinci adıma geç.
- **Adım 2:** $i=1$ olmak üzere birinci satırdaki birey için işlemi başlat, 240 satır boyunca program buraya her geldiğinde i 'yi bir artır ve bir sonraki satırdaki bireyi kullan. $ara=1$ saat olmak üzere zaman penceresi sayacı oluştur, program buraya her geldiğinde sayacı bir saat artır ve üçüncü adıma geç. Eğer $i=241$ değerine ulaşırsa on birinci adıma geç.
- **Adım 3:** $j=1$ olmak üzere hazır olarak verilen tablo içerisinde birinci sıradaki işi seçerek iterasyona başla, program buraya alt adımlardan her geldiğinde j 'yi bir artır ve bir sonraki işe geç. Program buraya üst adımlardan geldiğinde ise yeniden $j=1$ olmak üzere birinci işten başla ve dördüncü adıma geç. Eğer $j=(n+1)$ değerine ulaşırsa ikinci adıma dön.
- **Adım 4:** Üçüncü adımda seçilen işin işlenmekte olan bireyde mevcut olup olmadığını kontrol et. Eğer bu iş daha önce kullanılmış ise ve birey içinde mevcut ise üçüncü adıma dön. Eğer bu iş daha önce kullanılmamış ise birey üzerindeki boş olan ilk pozisyona işi yerleştir, o ana kadar yerleştirilmiş işler için akış zamanını hesapla ve beşinci adıma geç.
- **Adım 5:** $e=1$ olmak üzere sayaç oluştur ve sayaç değeri n iş sayısına ulaştığında üçüncü adıma dön. Sayaç değeri iş sayısından düşük olduğu sürece altıncı adıma geç.
- **Adım 6:** $k=1$ olmak üzere hazır olarak verilen tablo içerisinde birinci sıradaki işi seçerek iterasyona başla, program buraya alt adımlardan her geldiğinde k 'yi bir artır ve bir sonraki işe geç. Program buraya üst adımlardan geldiğinde ise yeniden $k=1$ olmak üzere birinci işten başla ve yedinci adıma geç. Eğer $k=(n+1)$ değerine ulaşırsa beşinci adıma dön.

- **Adım 7:** Altıncı adımda seçilen işin işlenmekte olan bireyde mevcut olup olmadığını kontrol et. Eğer bu iş daha önce kullanılmış ise ve birey içinde mevcut ise e sayaç değerini o anki k değeri ile eşitle ve altıncı adıma dön. Eğer bu iş daha önce kullanılmamış ise sekizinci adıma geç.
- **Adım 8:** Altıncı adımda seçilen ve yedinci adımdaki kontrolden geçen iş ile bireye yerleştirilen son iş arasındaki benzerlik ilişkisini ölçmek üzere hazırlık süreleri matrisine git. Eğer bireye yerleştirilen son işten altıncı adımda seçilen işe geçmek için bir saatten az hazırlık süresi gerekiyor ise bu iki iş arasında benzerlik ilişkisi olduğunu kabul et ve dokuzuncu adıma geç. Eğer bireye yerleştirilen son işten altıncı adımda seçilen işe geçmek için bir saatten fazla hazırlık süresi gerekiyor ise bu iki iş arasında benzerlik ilişkisi olmadığını kabul et, e sayaç değerini o anki k değeri ile eşitle ve altıncı adıma dön.
- **Adım 9:** Zaman penceresi kontrol adımıdır. Altıncı adımda seçilen, yedinci ve sekizinci adımdaki kontrollerden geçen iş değerlendirilir. Bu iş için tabloda verilen son teslim tarihinden ikinci adımda verilen zaman penceresi sayacının değerini çıkar ve iş için en erken başlama zamanını bul. Eğer bu iş için üretilen en erken başlama zamanı, o zamana kadar bireye yerleştirilen işlerin akış zamanından küçük ise, işin işleme hazır olduğunu kabul et ve onuncu adıma geç. Aksi halde, işin işleme hazır olmadığını kabul et, e sayaç değerini o anki k değeri ile eşitle ve altıncı adıma dön.
- **Adım 10:** Altıncı adımda seçilen, yedinci, sekizinci ve dokuzuncu adımdaki kontrollerden geçen işi birey üzerindeki boş olan ilk pozisyona yerleştir, o ana kadar yerleştirilmiş işler için akış zamanını hesapla, e sayaç değerini o anki k değeri ile eşitle ve beşinci adıma dön.
- **Adım 11:** Yukarıdaki iterasyonlarla oluşan popülasyonu maliyet fonksiyonuna gönder. Bireylerin maliyetlerini hesaplatır, çıkan sonuçlara göre maliyeti küçük olandan büyük olana göre sırala ve en düşük maliyetli beş bireyi seçerek GA için oluşturulan popülasyona kaydet.

Burada vurgulanması gereken bir diğer nokta ise bireyleri oluştururken ilk sıraya daima iş havuzunda birinci sırada verilen işin yerleştirilmesidir.



Şekil 3.7 S1 için akış diyagramı

3.4.4.2 İki Numaralı Sezgisel Algoritma (S2)

Temel olarak bir numaralı sezgisel model ile aynı yapıdadır. Tek farkı zaman penceresi hesaplanırken Tablo 3.8’de verilen *artış oranları* kullanılarak her iş için kendi ağırlıklarına göre zaman pencereleri oluşturulmasıdır.

- **Adım 1:** n iş sayısı olmak üzere 240 adet satırlı n adet sütunlu tüm elemanları 0 olan bir matris oluştur ve ikinci adıma geç.
- **Adım 2:** $i=1$ olmak üzere birinci satırdaki birey için işlemi başlat, 240 satır boyunca program buraya her geldiğinde i 'yi bir artır ve bir sonraki satırdaki bireyi kullan. $ara=1$ saat olmak üzere zaman penceresi sayacı oluştur, program buraya her geldiğinde sayacı bir saat artır ve üçüncü adıma geç. Eğer $i=241$ değerine ulaşırsa on birinci adıma geç.
- **Adım 3:** $j=1$ olmak üzere hazır olarak verilen tablo içerisinde birinci sıradaki işi seçerek iterasyona başla, program buraya alt adımlardan her geldiğinde j 'yi bir artır ve bir sonraki işe geç. Program buraya üst adımlardan geldiğinde ise yeniden $j=1$ olmak üzere birinci işten başla ve dördüncü adıma geç. Eğer $j=(n+1)$ değerine ulaşırsa ikinci adıma dön.
- **Adım 4:** Üçüncü adımda seçilen işin işlenmekte olan bireyde mevcut olup olmadığını kontrol et. Eğer bu iş daha önce kullanılmış ise ve birey içinde mevcut ise üçüncü adıma dön. Eğer bu iş daha önce kullanılmamış ise birey üzerindeki boş olan ilk pozisyona işi yerleştir, o ana kadar yerleştirilmiş işler için akış zamanını hesapla ve beşinci adıma geç.
- **Adım 5:** $e=1$ olmak üzere sayaç oluştur ve sayaç değeri n iş sayısına ulaştığında üçüncü adıma dön. Sayaç değeri iş sayısından düşük olduğu sürece altıncı adıma geç.
- **Adım 6:** $k=1$ olmak üzere hazır olarak verilen tablo içerisinde birinci sıradaki işi seçerek iterasyona başla, program buraya alt adımlardan her geldiğinde k 'yi bir artır ve bir sonraki işe geç. Program buraya üst adımlardan geldiğinde ise yeniden $k=1$ olmak üzere birinci işten başla ve yedinci adıma geç. Eğer $k=(n+1)$ değerine ulaşırsa beşinci adıma dön.
- **Adım 7:** İkinci adımda verilen zaman çizelgesi sayacı değeri ile tabloda verilen ve her iş için kendine has ağırlığı olan artış oranı değerini çarp. Böylelikle kk ağırlıklandırılmış zaman penceresi sayacı değerini hesapla. Altıncı adımda

seçilen işin işlenmekte olan bireyde mevcut olup olmadığını kontrol et. Eğer bu iş daha önce kullanılmış ise ve birey içinde mevcut ise e sayaç değerini o anki k değeri ile eşitle ve altıncı adıma dön. Eğer bu iş daha önce kullanılmamış ise sekizinci adıma geç.

- **Adım 8:** Altıncı adımda seçilen ve yedinci adımdaki kontrolden geçen iş ile bireye yerleştirilen son iş arasındaki benzerlik ilişkisini ölçmek üzere hazırlık süreleri matrisine git. Eğer bireye yerleştirilen son işten altıncı adımda seçilen işe geçmek için bir saatten az hazırlık süresi gerekiyor ise bu iki iş arasında benzerlik ilişkisi olduğunu kabul et ve dokuzuncu adıma geç. Eğer bireye yerleştirilen son işten altıncı adımda seçilen işe geçmek için bir saatten fazla hazırlık süresi gerekiyor ise bu iki iş arasında benzerlik ilişkisi olmadığını kabul et, e sayaç değerini o anki k değeri ile eşitle ve altıncı adıma dön.
- **Adım 9:** Zaman penceresi kontrol adımdır. Altıncı adımda seçilen, yedinci ve sekizinci adımdaki kontrollerden geçen işi değerlendirir. Bu iş için tabloda verilen son teslim tarihinden yedinci adımda verilen ağırlıklandırılmış zaman penceresi sayacının değerini çıkar ve iş için en erken başlama zamanını bul. Eğer bu iş için üretilen en erken başlama zamanı, o zamana kadar bireye yerleştirilen işlerin akış zamanından küçük ise, işin işleme hazır olduğunu kabul et ve onuncu adıma geç. Eğer bu iş için üretilen en erken başlama zamanı, o zamana kadar bireye yerleştirilen işlerin akış zamanından büyük ise, işin işleme hazır olmadığını kabul et, e sayaç değerini o anki k değeri ile eşitle ve altıncı adıma dön.
- **Adım 10:** Altıncı adımda seçilen, yedinci, sekizinci ve dokuzuncu adımdaki kontrollerden geçen işi birey üzerindeki boş olan ilk pozisyona yerleştir, o ana kadar yerleştirilmiş işler için akış zamanını hesapla, e sayaç değerini o anki k değeri ile eşitle ve beşinci adıma dön.
- **Adım 11:** Yukarıdaki iterasyonlarla oluşan popülasyonu maliyet fonksiyonuna gönder. Bireylerin maliyetlerini hesaplatır, çıkan sonuçlara göre maliyeti küçük olandan büyük olana göre sırala ve en düşük maliyetli beş bireyi seçerek GA için oluşturulan popülasyona kaydet.

Burada vurgulanması gereken nokta, bireyleri oluştururken ilk sıraya daima iş havuzunda birinci sırada verilen işin yerleştirilmesidir.

3.4.4.3 Üç Numaralı Sezgisel Algoritma (S3)

İki numaralı sezgiselin geliştirilmiş versiyonudur. İki numaralı sezgiselde bireylerin ilk elemanları daima tabloda ilk sıraya gelen eleman olurken burada ilk sıraya gelen eleman değiştirilerek daha uygun sonuçlar yakalanmaya çalışılmaktadır.

- **Adım 1:** n iş sayısı olmak üzere 7200 adet satırlı n adet sütunlu tüm elemanları 0 olan bir matris oluştur ve ikinci adıma geç.
- **Adım 2:** Kullanılacak ilk işin belirlendiği yerdir. $ii=1$ olmak üzere veri tablosundan birinci işi seç, 30 iterasyon boyunca program buraya her geldiğinde ii 'yi bir artır ve veri tablosundaki hangi işin kullanılacağını belirle. Zaman sayacını (ara) sıfırla ve üçüncü adıma geç. $ii=31$ olduğunda on üçüncü adıma geç.
- **Adım 3:** Program buraya ikinci adımdan her geldiğinde $i=1$ olmak üzere işlemi başlat, 240 satır boyunca program buraya alt adımlardan geldikçe i 'yi bir artır ve $i=241$ değerine ulaşırsa ikinci adıma dön. $iii=1$ olmak üzere birinci satır için işlemi başlat, program buraya her geldiğinde iii değerini bir artır ve bir sonraki satır için işlemi başlat. $ara=1$ saat olmak üzere zaman penceresi sayacı oluştur, program buraya her geldiğinde sayacı bir saat artır. Birinci adımda seçilen işi mevcut satırın birinci sırasına yerleştir, mevcut satır için akış zamanını hesapla ve dördüncü satıra geç.
- **Adım 4:** $j=1$ olmak üzere hazır olarak verilen tablo içerisinde birinci sıradaki işi seçerek iterasyona başla, program buraya alt adımlardan her geldiğinde j 'yi bir artır ve bir sonraki işe geç. Program buraya üst adımlardan geldiğinde ise yeniden $j=1$ olmak üzere birinci işten başla ve dördüncü adıma geç. Eğer $j=(n+1)$ değerine ulaşırsa üçüncü adıma dön.
- **Adım 5:** Dördüncü adımda seçilen işin işlenmekte olan bireyde mevcut olup olmadığını kontrol et. Eğer bu iş daha önce kullanılmış ise ve birey içinde mevcut ise dördüncü adıma dön. Eğer bu iş daha önce kullanılmamış ise altıncı adıma geç.
- **Adım 6:** $e=1$ olmak üzere sayaç oluştur ve sayaç değeri n iş sayısına ulaştığında on ikinci adıma geç. Sayaç değeri iş sayısından düşük olduğu sürece yedinci adıma geç.

- **Adım 7:** $k=1$ olmak üzere hazır olarak verilen tablo içerisinde birinci sıradaki işi seçerek iterasyona başla, program buraya alt adımlardan her geldiğinde k 'yi bir artır ve bir sonraki işe geç. Program buraya üst adımlardan geldiğinde ise yeniden $k=1$ olmak üzere birinci işten başla ve sekizinci adıma geç. Eğer $k=(n+1)$ değerine ulaşırsa altıncı adıma dön.
- **Adım 8:** Üçüncü adımda verilen zaman çizelgesi sayacı değeri ile tabloda verilen ve her iş için kendine has ağırlığı olan artış oranı değerini çarp. Böylelikle kk ağırlıklandırılmış zaman penceresi sayacı değerini hesapla. Yedinci adımda seçilen işin işlenmekte olan bireyde mevcut olup olmadığını kontrol et. Eğer bu iş daha önce kullanılmış ise ve birey içinde mevcut ise e sayaç değerini o anki k değeri ile eşitle ve yedinci adıma dön. Eğer bu iş daha önce kullanılmamış ise dokuzuncu adıma geç.
- **Adım 9:** Yedinci adımda seçilen ve sekizinci adımdaki kontrolden geçen iş ile bireye yerleştirilen son iş arasındaki benzerlik ilişkisini ölçmek üzere hazırlık süreleri matrisine git. Eğer bireye yerleştirilen son işten yedinci adımda seçilen işe geçmek için bir saatten az hazırlık süresi gerekiyor ise bu iki iş arasında benzerlik ilişkisi olduğunu kabul et ve onuncu adıma geç. Eğer bireye yerleştirilen son işten altıncı adımda seçilen işe geçmek için bir saatten fazla hazırlık süresi gerekiyor ise bu iki iş arasında benzerlik ilişkisi olmadığını kabul et, e sayaç değerini o anki k değeri ile eşitle ve yedinci adıma dön.
- **Adım 10:** Zaman penceresi kontrol adımıdır. Yedinci adımda seçilen, sekizinci ve dokuzuncu adımdaki kontrollerden geçen işi değerlendireceğiz. Bu iş için tabloda verilen son teslim tarihinden sekizinci adımda verilen ağırlıklandırılmış zaman penceresi sayacının değerini çıkar ve iş için en erken başlama zamanını bul. Eğer bu iş için üretilen en erken başlama zamanı, o zamana kadar bireye yerleştirilen işlerin akış zamanından küçük ise, işin işleme hazır olduğunu kabul et ve on birinci adıma geç. Eğer bu iş için üretilen en erken başlama zamanı, o zamana kadar bireye yerleştirilen işlerin akış zamanından büyük ise, işin işleme hazır olmadığını kabul et, e sayaç değerini o anki k değeri ile eşitle ve yedinci adıma dön.
- **Adım 11:** Yedinci adımda seçilen, sekizinci, dokuzuncu ve onuncu adımdaki kontrollerden geçen işi birey üzerindeki boş olan ilk pozisyona yerleştir, o ana

kadar yerleştirilmiş işler için akış zamanını hesapla, e sayaç değerini o anki k değeri ile eşitle ve altıncı adıma dön.

- **Adım 12:** Program buraya altıncı adımdan geldiğinde dördüncü adımda seçilen işi birey üzerindeki boş olan ilk pozisyona yerleştir, o ana kadar yerleştirilmiş işler için akış zamanını hesapla ve dördüncü adıma dön.
- **Adım 13:** Yukarıdaki iterasyonlarla oluşan popülasyonu maliyet fonksiyonuna gönder. Bireylerin maliyetlerini hesaplatır, çıkan sonuçlara göre maliyeti küçük olandan büyük olana göre sırala ve en düşük maliyetli beş bireyi seçerek GA için oluşturulan popülasyona kaydet.

Burada vurgulanması gereken nokta, bireyleri oluştururken birinci sıraya iş havuzunda ilk otuz içerisinde olan işlerden seçim yapılarak iyi bir sonuç elde edilmeye çalışılmıştır.

3.4.4.4 Dört Numaralı Sezgisel Algoritma (S4)

Zaman penceresine giren işler arasından en düşük maliyetli olanı sıraya yerleştiren bir yöntemdir. Üçüncü sezgiselle benzerlikler taşır.

- **Adım 1:** n iş sayısı olmak üzere 7200 adet satırlı n adet sütunlu tüm elemanları 0 olan bir matris oluştur ve ikinci adıma geç.
- **Adım 2:** Kullanılacak ilk işin belirlendiği yerdir. $ii=1$ olmak üzere veri tablosundan birinci işi seç, 30 iterasyon boyunca program buraya her geldiğinde ii 'yi bir artır ve veri tablosundaki hangi işin kullanılacağını belirle. Zaman sayacını (ara) sıfırla ve üçüncü adıma geç. $ii=31$ olduğunda on dördüncü adıma geç.
- **Adım 3:** Program buraya ikinci adımdan her geldiğinde $i=1$ olmak üzere işlemi başlat, 240 satır boyunca program buraya alt adımlardan geldikçe i 'yi bir artır ve $i=241$ değerine ulaşırsa ikinci adıma dön. $iii=1$ olmak üzere birinci satır için işlemi başlat, program buraya her geldiğinde iii değerini bir artır ve bir sonraki satır için işlemi başlat. $ara=1$ saat olmak üzere zaman penceresi sayacı oluştur, program buraya her geldiğinde sayacı bir saat artır. Birinci adımda seçilen iş mevcut satırın birinci sırasına yerleştir, mevcut satır için akış zamanını hesapla ve dördüncü satıra geç.
- **Adım 4:** $j=1$ olmak üzere hazır olarak verilen tablo içerisinde birinci sıradaki işi seçerek iterasyona başla, program buraya alt adımlardan her geldiğinde j 'yi bir artır ve bir sonraki işe geç. Program buraya üst adımlardan geldiğinde ise yeniden $j=1$ olmak üzere birinci işten başla ve dördüncü adıma geç. Eğer $j=(n+1)$ değerine ulaşırsa üçüncü adıma dön.
- **Adım 5:** Dördüncü adımda seçilen işin işlenmekte olan bireyde mevcut olup olmadığını kontrol et. Eğer bu iş daha önce kullanılmış ise ve birey içinde mevcut ise dördüncü adıma dön. Eğer bu iş daha önce kullanılmamış ise altıncı adıma geç.
- **Adım 6:** $e=1$ olmak üzere sayaç oluştur ve sayaç değeri n iş sayısına ulaştığında on üçüncü adıma geç. Sayaç değeri iş sayısından düşük olduğu sürece $maliyet=5000000$ olmak üzere çok büyük bir değer belirle ve yedinci adıma geç.
- **Adım 7:** $k=1$ olmak üzere hazır olarak verilen tablo içerisinde birinci sıradaki işi seçerek iterasyona başla, program buraya alt adımlardan her geldiğinde k 'yi

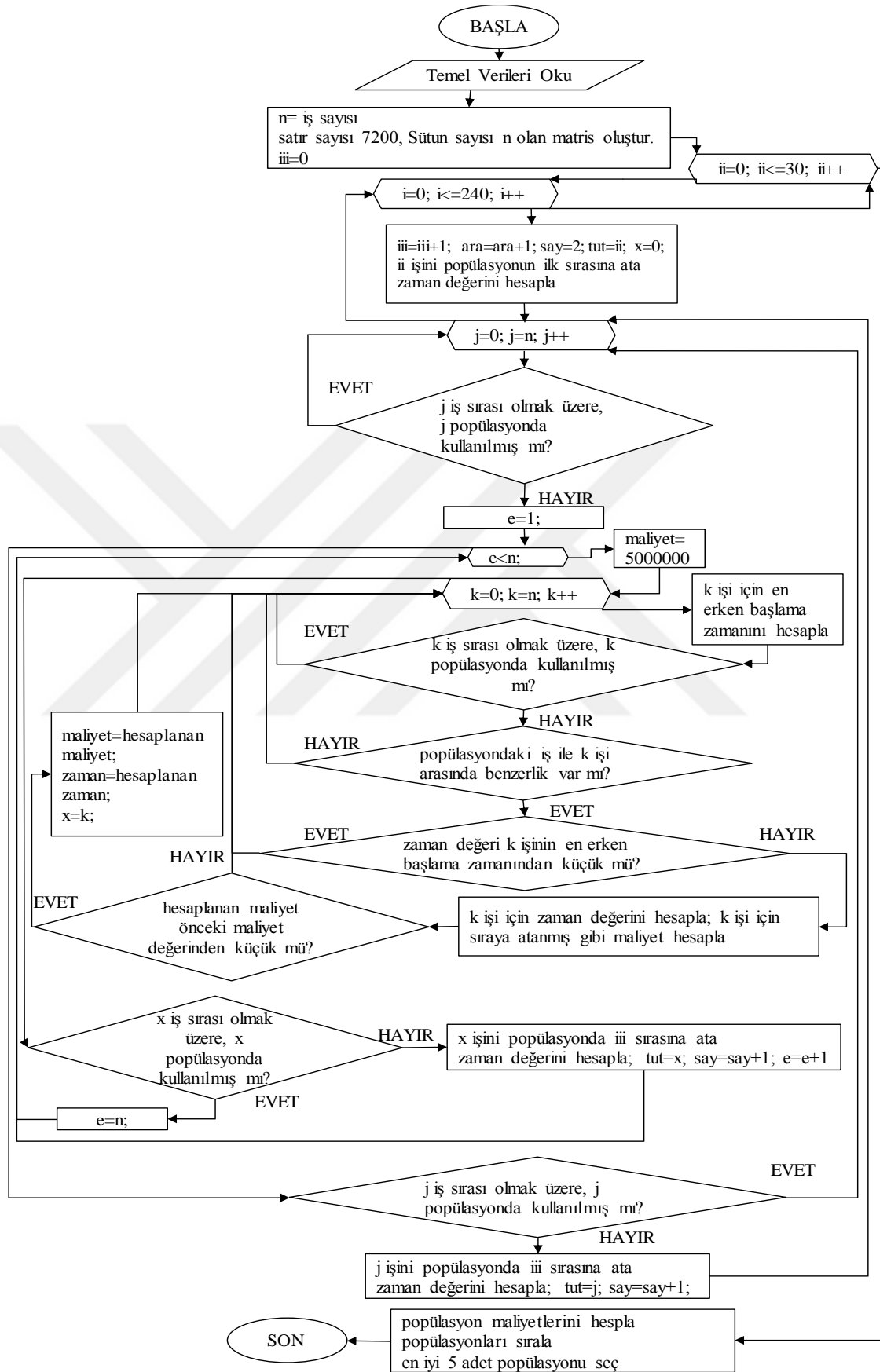
bir artır ve bir sonraki işe geç. Program buraya üst adımlardan geldiğinde ise yeniden $k=1$ olmak üzere birinci işten başla ve sekizinci adıma geç. Eğer $k=(n+1)$ değerine ulaşırsa on ikinci adıma geç.

- **Adım 8:** Üçüncü adımda verilen zaman çizelgesi sayacı değeri ile tabloda verilen ve her iş için kendine has ağırlığı olan artış oranı değerini çarp. Böylelikle kk ağırlıklandırılmış zaman penceresi sayacı değerini hesapla. Yedinci adımda seçilen işin işlenmekte olan bireyde mevcut olup olmadığını kontrol et. Eğer bu iş daha önce kullanılmış ise ve birey içinde mevcut ise e sayaç değerini o anki k değeri ile eşitle ve yedinci adıma dön. Eğer bu iş daha önce kullanılmamış ise dokuzuncu adıma geç.
- **Adım 9:** Yedinci adımda seçilen ve sekizinci adımdaki kontrolden geçen iş ile bireye yerleştirilen son iş arasındaki benzerlik ilişkisini ölçmek üzere hazırlık süreleri matrisine git. Eğer bireye yerleştirilen son işten yedinci adımda seçilen işe geçmek için bir saatten az hazırlık süresi gerekiyor ise bu iki iş arasında benzerlik ilişkisi olduğunu kabul et ve onuncu adıma geç. Eğer bireye yerleştirilen son işten altıncı adımda seçilen işe geçmek için bir saatten fazla hazırlık süresi gerekiyor ise bu iki iş arasında benzerlik ilişkisi olmadığını kabul et, e sayaç değerini o anki k değeri ile eşitle ve yedinci adıma dön.
- **Adım 10:** Zaman penceresi kontrol adımıdır. Yedinci adımda seçilen, sekizinci ve dokuzuncu adımdaki kontrollerden geçen işi değerlendirir. Bu iş için tabloda verilen son teslim tarihinden sekizinci adımda verilen ağırlıklandırılmış zaman penceresi sayacının değerini çıkar ve iş için en erken başlama zamanını bul. Eğer bu iş için üretilen en erken başlama zamanı, o zamana kadar bireye yerleştirilen işlerin akış zamanından küçük ise, işin işleme hazır olduğunu kabul et ve on birinci adıma geç. Eğer bu iş için üretilen en erken başlama zamanı, o zamana kadar bireye yerleştirilen işlerin akış zamanından büyük ise, işin işleme hazır olmadığını kabul et ve yedinci adıma dön.
- **Adım 11:** Yedinci adımda seçilen, sekizinci, dokuzuncu ve onuncu adımdaki kontrollerden geçen iş için sıradaki boş pozisyona atanmış gibi akış zamanını hesapla. Bu iş için sıradaki pozisyona atanmış gibi maliyet hesabı yap ve bu maliyeti altıncı adımda verilen maliyet değeri ile kıyasla. Eğer yeni maliyet eski maliyetten küçük ise *maliyet* değerinin yeni değeri olarak düşük değeri ata. O iş

için akış zamanı değerini, maliyet değerini, işin sırasının verisini tut ve yedinci adıma dön.

- **Adım 12:** On birinci adımda seçilen işin işlenmekte olan bireyde mevcut olup olmadığını kontrol et. Eğer iş kullanılmış ise e sayaç değerini n iş sayısı değerine eşitle ve altıncı adıma dön. Eğer iş daha önce kullanılmamış ise boş olan ilk sıraya işi ata, o zamana kadar kullanılan işler için akış zamanını hesapla, e sayaç değerini bir artır ve altıncı adıma dön.
- **Adım 13:** Program buraya altıncı adımdan geldiğinde dördüncü adımda seçilen işi birey üzerindeki boş olan ilk pozisyona yerleştir, o ana kadar yerleştirilmiş işler için akış zamanını hesapla ve dördüncü adıma dön.
- **Adım 14:** Yukarıdaki iterasyonlarla oluşan popülasyonu maliyet fonksiyonuna gönder. Bireylerin maliyetlerini hesaplatır, çıkan sonuçlara göre maliyeti küçük olandan büyük olana göre sırala ve en düşük maliyetli beş bireyi seçerek GA için oluşturulan popülasyona kaydet.

Bu algorithmada vurgulanması gereken nokta işler seçilirken zaman penceresi içinde kalan tüm işler arasından en uygun maliyetli olan işin sıraya yazılmasıdır.



Şekil 3.10 S4 için akış diyagramı

3.5 GA ve Sezgisel Algoritmaların Analizi

Geliştirilen sezgisel algoritmaları test edip performanslarını değerlendirmek için çeşitli yöntemler ve dağılımlar kullanılarak veriler türetilmiştir. Bu veriler kullanılarak sezgisel algoritmaların ürettiği popülasyonların maliyetleri kıyaslanmıştır ve sezgisel modellerin başarısı hakkında yorumlar yapılmıştır. Daha sonra ise sezgisel algoritmaların ürettiği bu başlangıç popülasyonları ile GA çalıştırılmış ve çıkan sonuçlar yine birbiri arasında kıyaslanmıştır.

3.5.1 Sentetik Deneyler İçin Senaryolar

Veriler, firmada daha önce yapılmış olan bir üretim çizelgeleme probleminde kullanılan işler referans alınarak türetilenlerdir. Bu çizelge, firmadaki gerçek zamanlı işler kullanılarak bir planlama uzmanı tarafından hazırlanmıştır ve 137 adet iş içermektedir. Hazırlanan gerçek çizelgenin sırası, akış zamanı, hazırlık süreleri toplamı ve maliyetleri mevcuttur.

Örnek olarak alınan çizelgenin ve firmadaki işlerin teslim tarihleri hafif dalgalanmalar içerse de genel olarak düzgün dağılıma uymaktadır. Burada teslim tarihleri için farklı dağılımlar kullanılarak veriler üretilenlerdir ve bu veriler kullanılarak sezgisel algoritmaların performansları değerlendirilecektir. Teslim tarihleri dışında kalan değişkenler ise orijinal verideki dağılımları kullanılarak türetilenlerdir. Ayrıca üretilen veriler, gerekli olan toplam makine zamanları göz önüne alınarak farklılaştırılacak ve makine kapasitesini %90 - %100 - %110 dolduracak şekilde çoğaltılacaktır. Çizelge başlangıç zamanı sıfır olarak alınacaktır ve teslim tarihleri sıfır zamanından başlatılarak türetilenlerdir. Tablo 3.10'da veri türetme mantığı gösterilmiştir.

Tablo 3.10 Veri türetme tablosu

	İş Miktarı	Ortalama (ORT- Saat)	Standart Sapma (STD)	%90 Dolu	%100 Dolu	%110 Dolu
Düzensiz Dağılım (DD)	100	270	X	10 Tekrarlı	10 Tekrarlı	10 Tekrarlı
	300	810	X	3 Tekrarlı	3 Tekrarlı	3 Tekrarlı
Normal Dağılım 1 (ND1)	100	270	305	10 Tekrarlı	10 Tekrarlı	10 Tekrarlı
	300	810	305	3 Tekrarlı	3 Tekrarlı	3 Tekrarlı
Normal Dağılım 2 (ND2)	100	270	420	10 Tekrarlı	10 Tekrarlı	10 Tekrarlı
	300	810	420	3 Tekrarlı	3 Tekrarlı	3 Tekrarlı

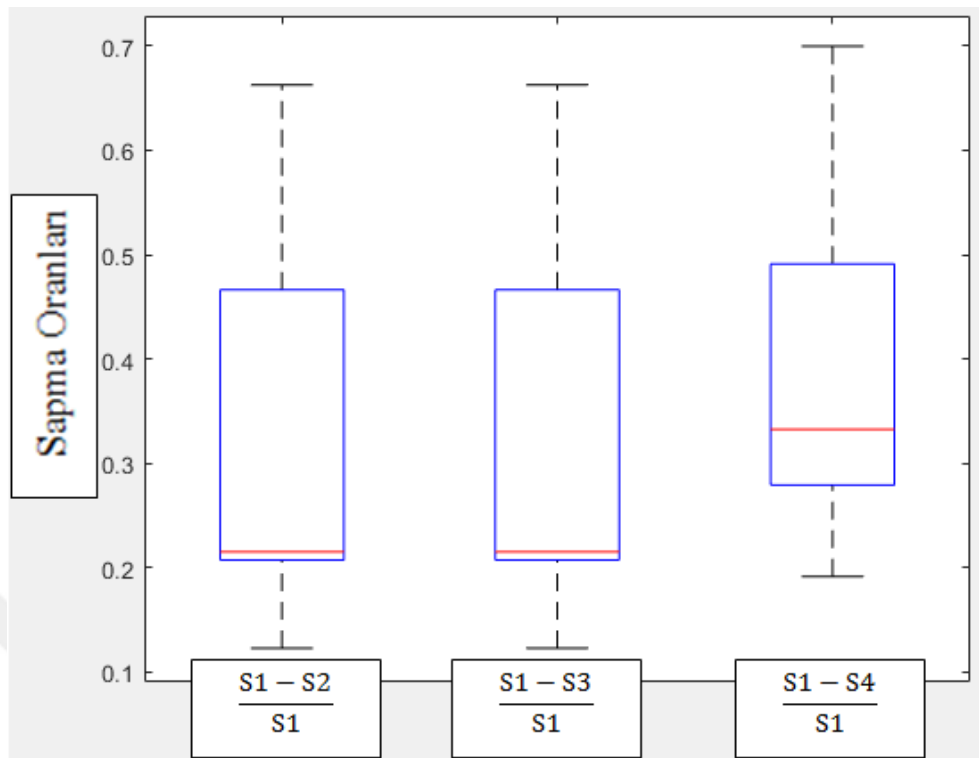
Tablo 3.10’da görüldüğü gibi üç farklı dağılımda veri türetilmiştir. Düzensiz dağılımda türetilen veriler orijinal verinin bir benzeridir ve sadece iş sayıları farklılaştırılmıştır. Normal dağılım 1 verileri (ND1) standart sapması düşük olan ve işlerin teslim tarihlerinin ortalama etrafında ciddi şekilde yoğunlaştığı bir veri setidir. Normal dağılım 2 verileri (ND2) ise standart sapmaların yüksek olduğu ve işlerin teslim tarihlerinin nispeten ortalamadan uzaklaştığı daha basık normal dağılım karakteristiği gösteren bir veri setidir. Bundan sonraki analizlerde bu veri setleri kullanılacaktır.

3.5.2 Veri Setlerinin Sezgisel Algoritmalar ile Analizi

Bu bölümde türetilen veriler sezgisel algoritmalarla denenmiş ve aynı veri setleri için sezgisel algoritmaların birbirleri ile olan performans ilişkisi incelenmiştir. Tablo 3.11’de bu inceleme sonucu elde edilen sonuçlar paylaşılmıştır. Tablo 3.12’de ise sezgisel algoritmaların ürettiği sonuçlar eşleştirilmiş örneklemeler *t* testi yardımıyla birbirleri ile kıyaslanmıştır. Böylelikle sezgisel algoritmaların sonuçları arasında benzerlik olup olmadığı incelenmiştir.

Tablo 3.11 Sezgisel algoritmalar ile veri analizi

				Sezgiseller için İş Başına Ortalama Birim Maliyet				1 Numaralı Sezgisel Algoritma İle Diğerlerinin Kıyaslanması			
		İş Sayısı	Doluluk Oranı		S1	S2	S3	S4	(S1-S2)/S1	(S1-S3)/S1	(S1-S4)/S1
DD	100	%100 Dolu	ORT	408	323	291	272	%20,8	%28,5	%33,2	
			STD	253	171	160	153	%32,4	%36,5	%39,5	
		%90 Dolu	ORT	175	139	124	113	%20,8	%29,1	%35,3	
			STD	93	56	48	40	%40,0	%47,8	%56,6	
		%110 Dolu	ORT	1426	1214	1159	1105	%14,9	%18,7	%22,5	
			STD	1033	1010	955	928	%2,3	%7,6	%10,2	
	300	%100 Dolu	ORT	335	266	227	211	%20,7	%32,3	%37,0	
			STD	173	81	67	56	%53,2	%61,6	%67,6	
		%90 Dolu	ORT	179	154	140	144	%13,6	%21,8	%19,2	
			STD	44	38	18	22	%12,4	%58,7	%49,5	
		%110 Dolu	ORT	1115	978	916	804	%12,3	%17,8	%27,9	
			STD	383	317	373	315	%17,2	%2,6	%17,7	
ND1	100	%100 Dolu	ORT	477	372	349	318	%22,0	%26,9	%33,4	
			STD	233	195	187	179	%16,1	%19,5	%22,9	
		%90 Dolu	ORT	212	167	158	153	%21,2	%25,5	%27,9	
			STD	135	84	82	76	%38,0	%39,1	%44,1	
		%110 Dolu	ORT	1244	938	904	871	%24,6	%27,3	%30,0	
			STD	575	436	423	413	%24,2	%26,4	%28,2	
	300	%100 Dolu	ORT	4799	2178	2149	1956	%54,6	%55,2	%59,2	
			STD	932	690	658	556	%26,0	%29,3	%40,3	
		%90 Dolu	ORT	3194	1078	1029	959	%66,3	%67,8	%70,0	
			STD	785	367	341	330	%53,2	%56,5	%57,9	
		%110 Dolu	ORT	6760	3607	3585	3439	%46,6	%47,0	%49,1	
			STD	1187	992	980	1086	%16,4	%17,4	%8,5	
ND2	100	%100 Dolu	ORT	499	392	365	334	%21,4	%26,8	%33,0	
			STD	249	193	178	153	%22,4	%28,5	%38,4	
		%90 Dolu	ORT	203	162	151	138	%20,5	%25,5	%32,3	
			STD	91	53	46	34	%42,0	%49,3	%62,4	
		%110 Dolu	ORT	1282	1004	960	928	%21,7	%25,1	%27,6	
			STD	685	581	562	556	%15,2	%17,9	%18,8	
	300	%100 Dolu	ORT	2215	1087	1032	972	%50,9	%53,4	%56,1	
			STD	796	779	714	701	%2,1	%10,3	%11,9	
		%90 Dolu	ORT	1040	434	413	396	%58,3	%60,3	%61,9	
			STD	652	345	309	286	%47,1	%52,6	%56,2	
		%110 Dolu	ORT	3611	2362	2256	2117	%34,6	%37,5	%41,4	
			STD	1193	1113	1102	1053	%6,7	%7,6	%11,7	



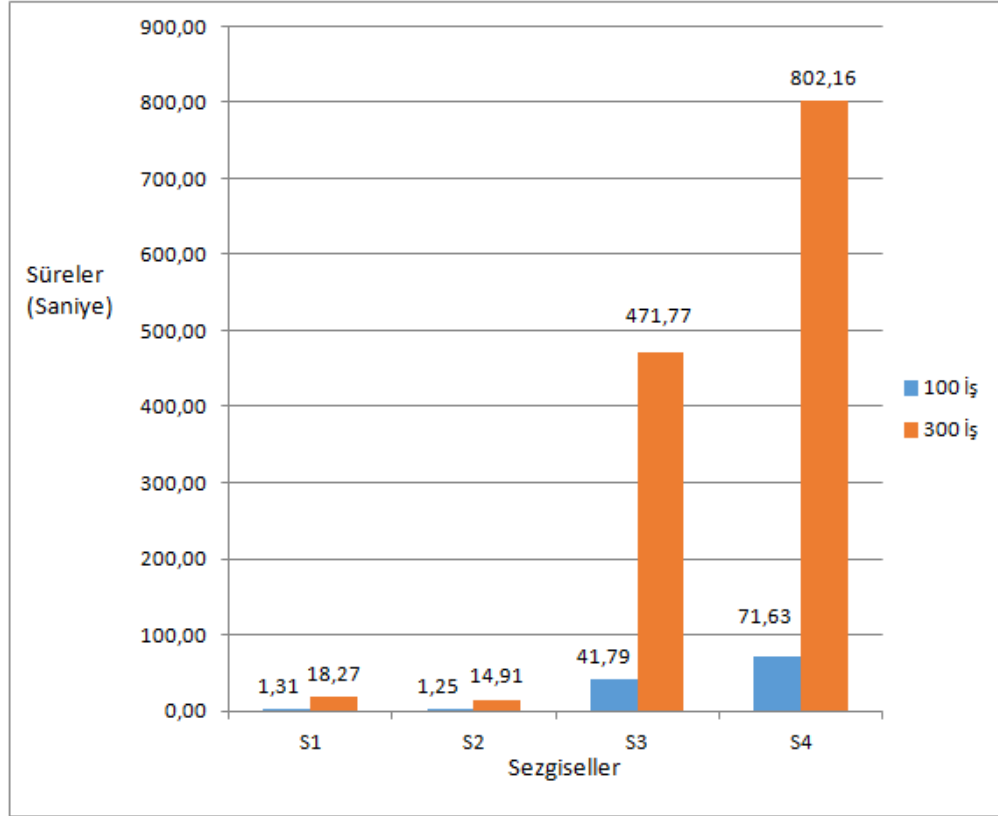
Şekil 3.11 Sezgisel algoritmaların S1 algoritması ile oranlarının kutu grafiği

Tablo 3.12 Sezgisel algoritmaların birbirleri ile ilişkisi

Sezgiseller	S1	S2	S3	S4
S1	-	1	1	1
S2	<i>p<0,009</i>	-	1	1
S3	<i>p<0,006</i>	<i>p<0,000</i>	-	1
S4	<i>p<0,005</i>	<i>p<0,000</i>	<i>p<0,001</i>	-

Tablo 3.12'den görüldüğü gibi %95 güven düzeyinde yapılan teste göre sezgisel algoritmaların sonuçları arasında benzerlik ilişkisi bulunamamıştır. Tablo 3.11'de ise diğer sezgisellerin S1'den sapmalarına bakıldığında en büyük sapmaların S4'te olduğunu görmekteyiz. Şekil 3.7 ile Tablo 3.11 birlikte karşılaştırıldığında da S4 algoritmasının ürettiği sonuçların daha istikrarlı olduğu ve orta noktasının daha yüksek olduğu görülmektedir. Bu durum S4'ün ürettiği popülasyonların maliyetlerinin daha düşük olduğunu ve diğer sezgisellere göre daha başarılı olduğunu söylemektedir. S3 algoritması ise S4 algoritmasına yakın değerler üreterek ikinci iyi sonuçları vermiştir. Şekil 3.8'de geliştirilen sezgisellerin çözüm üretmek için harcadıkları bilgisayar zamanları gösterilmiştir. S4 algoritması en iyi çözümü üretebilmekle beraber en fazla

zamanı harcamaktadır ve yine arkasından S3 algoritması gelmektedir. İş sayısı arttıkça zaman ihtiyacı da katlanarak artmaktadır.



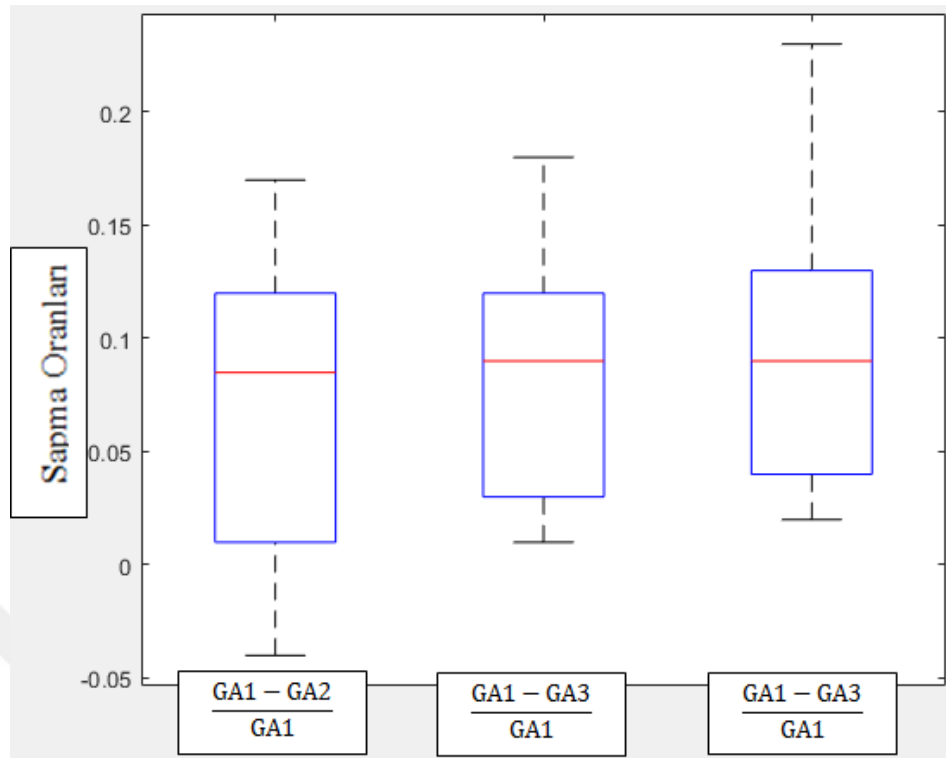
Şekil 3.12 Sezgisel modeller için çalışma zamanları

3.5.3 Veri Setlerinin GA ile Analizi

Bu bölümde sezgisel algoritmalar ile üretilen başlangıç popülasyonları GA ile iyileştirilecektir ve başlangıç popülasyonuna göre üretilen sonuçlar birbirleri ile kıyaslanarak performansları değerlendirilecektir. Tablo 3.13’de bu inceleme sonu elde edilmiş sonuçlar paylaşılmıştır. Tablo 3.14’de ise sezisel algoritmaların ürettiği başlangıç popülasyonları GA’da çalıştırılmıştır ve sonuçlar eşleştirilmiş örneklemeler t testi yardımıyla birbirleri ile kıyaslanmıştır. Böylelikle GA’ların sonuçları arasında benzerlik olup olmadığı incelenmiştir.

Tablo 3.13 GA ile veri analizi

	İş Sayısı	Doluluk oranı		GA için İş Başına Ortalama Birim Maliyet				1 Numaralı GA ile Diğerlerinin Kıyaslanması		
				GA1	GA2	GA3	GA4	(GA1-GA2) /GA1	(GA1-GA3) /GA1	(GA1-GA4) /GA1
DD	100	% 100 Dolu	ORT	165	148	150	144	% 10,3	% 8,6	% 12,6
			STD	115	103	104	102	% 11,0	% 10,1	% 11,5
		% 90 Dolu	ORT	69	68	66	68	% 1,1	% 4,9	% 1,8
			STD	6	8	5	8	% -20,5	% 12,9	% -23,2
		% 110 Dolu	ORT	679	670	672	662	% 1,3	% 0,9	% 2,5
			STD	552	563	563	570	% -2,0	% -2,0	% -3,2
	300	% 100 Dolu	ORT	122	127	119	118	% -4,3	% 2,4	% 3,7
			STD	28	28	30	19	% -0,4	% -5,6	% 34,0
		% 90 Dolu	ORT	96	95	93	93	% 1,3	% 2,9	% 3,0
			STD	9	7	7	7	% 23,7	% 22,6	% 22,0
		% 110 Dolu	ORT	514	517	507	470	% -0,5	% 1,4	% 8,6
			STD	190	213	210	213	% -12,0	% -10,5	% -11,7
ND1	100	% 100 Dolu	ORT	208	194	186	189	% 6,8	% 10,6	% 9,0
			STD	150	141	141	138	% 6,0	% 5,7	% 7,4
		% 90 Dolu	ORT	93	89	88	86	% 3,7	% 5,4	% 7,7
			STD	49	44	50	42	% 9,6	% -2,1	% 14,5
		% 110 Dolu	ORT	502	500	491	494	% 0,4	% 2,3	% 1,6
			STD	245	250	237	243	% -2,0	% 3,3	% 0,9
	300	% 100 Dolu	ORT	1194	1054	1055	1038	% 11,7	% 11,6	% 13,0
			STD	429	405	349	374	% 5,5	% 18,7	% 12,8
		% 90 Dolu	ORT	537	449	457	416	% 16,4	% 15,0	% 22,6
			STD	180	168	187	133	% 6,4	% -4,1	% 26,0
		% 110 Dolu	ORT	2238	1946	1925	1949	% 13,1	% 14,0	% 12,9
			STD	516	532	569	585	% -3,1	% -10,2	% -13,4
ND2	100	% 100 Dolu	ORT	219	194	198	199	% 11,5	% 9,3	% 9,2
			STD	115	91	95	98	% 20,4	% 17,6	% 14,6
		% 90 Dolu	ORT	86	76	77	76	% 11,3	% 9,7	% 11,4
			STD	32	14	17	14	% 57,4	% 47,4	% 56,4
		% 110 Dolu	ORT	533	517	509	511	% 3,1	% 4,5	% 4,2
			STD	289	305	308	308	% -5,8	% -6,5	% -6,7
	300	% 100 Dolu	ORT	592	506	510	497	% 14,5	% 13,8	% 15,9
			STD	429	406	439	414	% 5,4	% -2,3	% 3,6
		% 90 Dolu	ORT	206	171	168	171	% 16,8	% 18,4	% 16,9
			STD	91	45	32	36	% 50,5	% 64,4	% 60,6
		% 110 Dolu	ORT	1364	1228	1210	1212	% 10,0	% 11,3	% 11,1
			STD	681	782	685	804	% -14,9	% -0,7	% -18,1



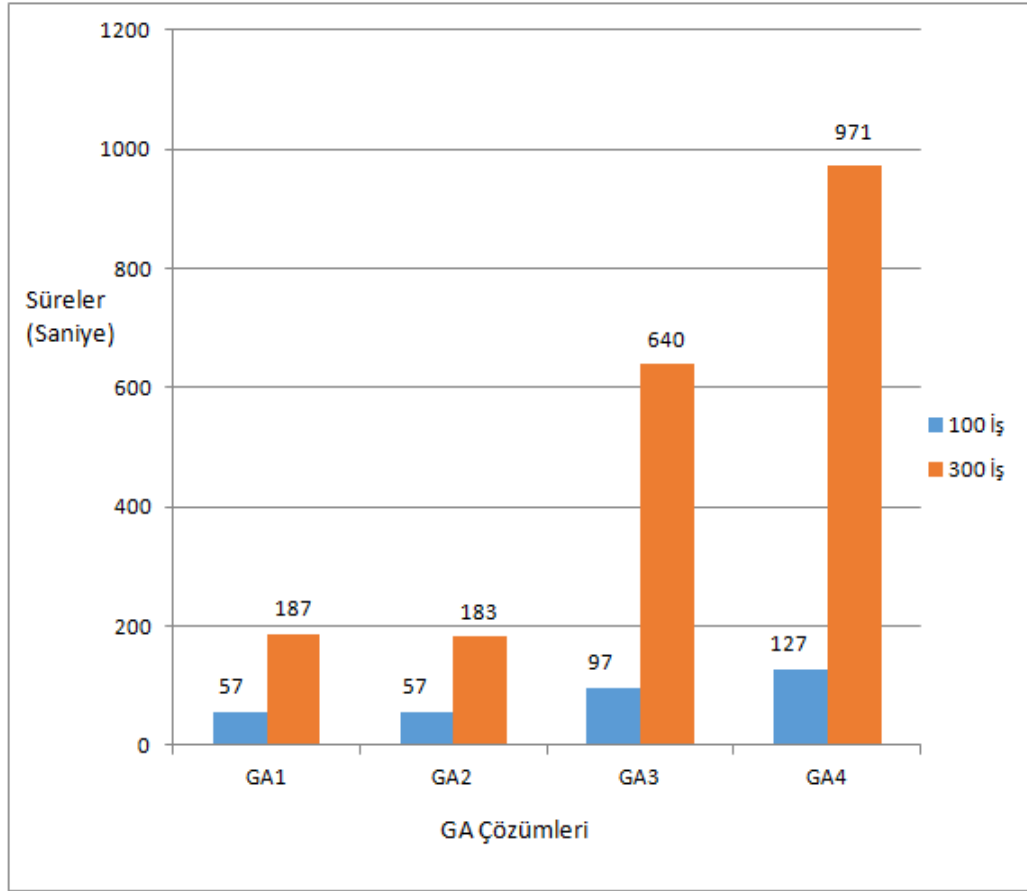
Şekil 3.13 GA'ların GA1 algoritması ile oranlarının kutu grafiği

Tablo 3.14 GA sonuçlarının birbirleri ile ilişkisi

GA'lar	GA1	GA2	GA3	GA4
GA1	-	1	1	1
GA2	$p < 0,016$	-	0	1
GA3	$p < 0,014$	$p < 0,053$	-	0
GA4	$p < 0,006$	$p < 0,011$	$p < 0,185$	-

Bu analiz yapılırken GA her iş grubu ve veri seti için 100000 iterasyon çalıştırılmıştır. Tablo 3.13'de S1 sezgiselinin ürettiği popülasyonun GA'da kullanılması ile üretilen sonuçlar diğer başlangıç popülasyonlarının performansı ile kıyaslanmıştır. S4'ün ürettiği başlangıç popülasyonları iyi performans göstermekle birlikte S3 algoritması da ona yakın değerler üretmiştir. Tablo 3.14'e bakıldığında ise %95 güven düzeyinde GA1 ile GA2, GA3 ve GA4 arasında bir ilişki olmadığı görülmüştür. Tablodan bakıldığında GA2 ile GA3 arasında bir ilişki varken GA4 ile aralarında bir ilişki olmadığı görülmektedir. Şekil 3.9'da genetik algoritmaların kıyaslandığı kutu grafiklerine bakıldığında da GA3 ve GA4 sonuçlarının daha istikrarlı ve birbirine yakın değerler

olduğu görülmektedir. Şekil 3.10'da ise her bir sezgisel ile GA'nın kombinasyonu ile çözüm üretmek için gerekli olan bilgisayar zamanları gösterilmiştir.



Şekil 3.14 Genetik algoritmalar için çalışma zamanları

3.6 Gerçek Veri Seti ile Çalışmalar

Burada firmada yapılmış bir çizelge ve bu çizelgede kullanılan gerçek veriler ile incelemeler sonucunda oluşturulan GA algoritmasının bir kıyaslaması yapılacaktır. Öncelikle amaç fonksiyonu katsayılarını değiştirmenin sonuçlar üzerine etkisini inceleyen bir deneysel tasarım yapılacaktır.

3.6.1 Deneysel Tasarımı

$f(s) = \sum_{j=1}^n (\alpha_j E_j^2 + \beta T_j + \mu s_j)$ amaç fonksiyonunda üç ayrı katsayı kullanılmıştır.

α katsayısı altında iki ayrı alt katsayı barındırmaktadır, bunlar tellerin sarılı olduğu bir makaranın bir saat dolu kalmasının maliyeti ve bir kilo telin bir saat stokta bulundurulma maliyetidir. β katsayısı ise bir işin gecikmesinin maliyetidir ve işin

miktarından bağımsız sabit bir değerdir. μ katsayısı ise hazırlık yapılan makinenin boşta kalma maliyetidir ve yine işin miktarından bağımsız sabit bir değerdir. Bu katsayıların sayısal karşılıkları firma gizlilik kuralları gereği verilmemiştir. Amaç fonksiyonu katsayıları üzerinde değişiklikler yapıp programlar bu şekilde çalıştırılacak ve performansları değerlendirilecektir. Tablo 3.15'te katsayılar için kullanılacak düzeylerin gösterimi yapılmıştır.

Tablo 3.15 Deney tasarımı düzeneği

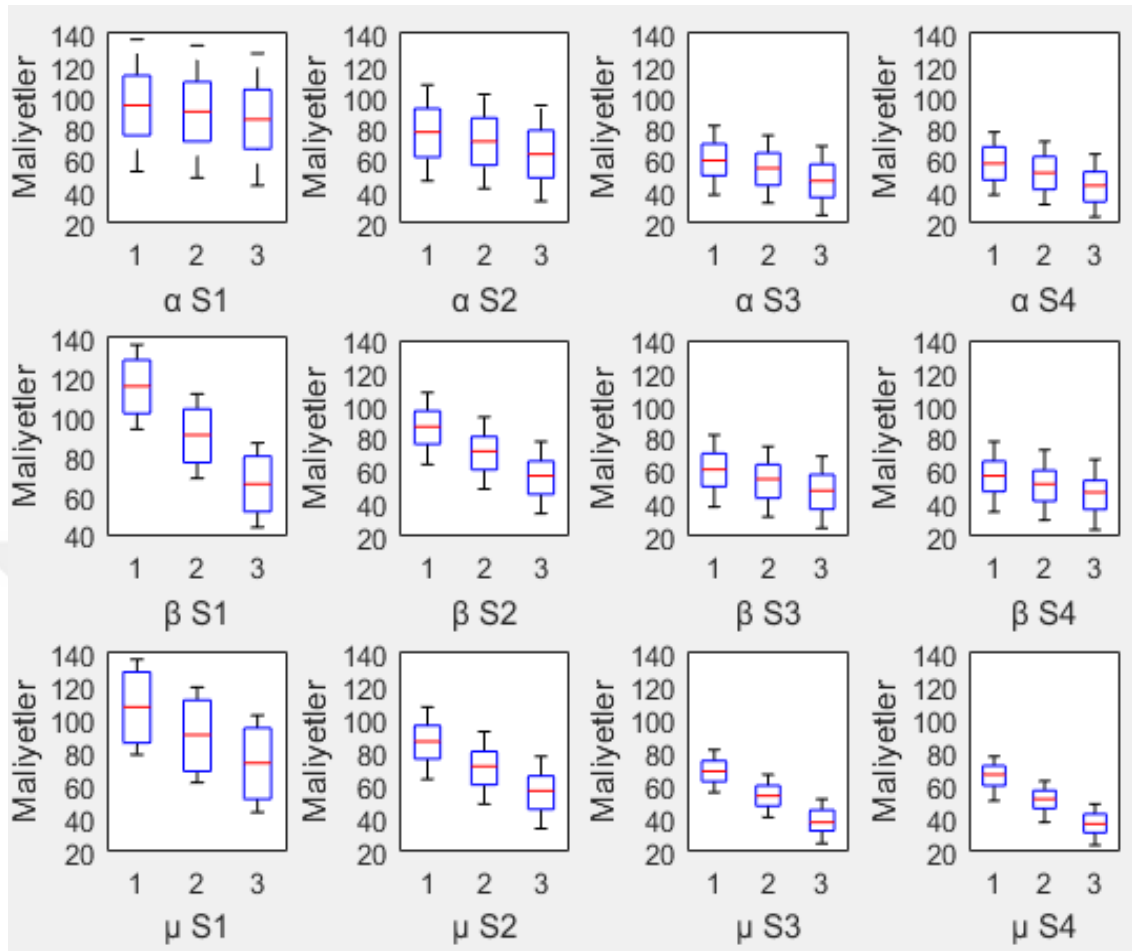
		1. Düzey	2. Düzey	3. Düzey
Katsayılar	α	a1	a2	a3
	β	b1	b2	b3
	μ	m1	m2	m3

Tablo 3.15'te verilen katsayılar için 1. Düzey firmada kullanılan orijinal katsayıların %50 fazlasını temsil etmektedir. 2. Düzeyde firmada kullanılan orijinal katsayılar kullanılmıştır, 3. Düzeyde ise orijinal katsayıların %50 eksikliği tasarımda denemeye tabi tutulacaktır.

Bundan sonraki kısımda düzeyler için geçerli olabilecek tüm kombinasyonlar geliştirilen programlarda 50000 iterasyon boyunca çalıştırılacaktır ve çıkan sonuçlar analiz edilecektir. Öncelikle sezgisel algoritmalar ayrı ayrı başlangıç popülasyonları üretecektir ve bu popülasyonlar GA'da çözülerek sonuçlar kendi içerisinde değerlendirilecektir. Tablo 3.16'te her bir düzey için sezgisel algoritmaların ürettiği sonuçlar gösterilmiştir. Buradan da görüldüğü sezgisellerin maliyetlerine bakıldığında S1 algoritmasının en kötü sonucu verdiği görülmüştür ve diğer sezgiseller bu algoritma ile kıyaslanmıştır. Yapılan oranlamalarda S4 sezgisel algoritmasının en iyi sonucu verdiği görülmüştür.

Tablo 3.16 Deney tasarımı sezgisel algoritmalar sonuçları

Denemeler	Katsayılar ve Düzeyleri			Sezgiseller için İş Başına Ortalama Birim Maliyet				1 Numaralı Sezgisel Algoritma ile Diğerlerinin Kıyaslanması		
	α	β	μ	S1	S2	S3	S4	(S1-S2) /S1	(S1-S3) /S1	(S1-S4) /S1
Deneme 1	a1	b1	m1	137	108	82	78	%21,1	%40,2	%43,0
Deneme 2	a1	b1	m2	120	93	67	63	%22,5	%44,3	%47,1
Deneme 3	a1	b1	m3	103	78	52	49	%24,4	%49,7	%52,6
Deneme 4	a1	b2	m1	112	93	75	73	%17,2	%32,9	%35,3
Deneme 5	a1	b2	m2	95	78	60	58	%18,2	%36,7	%39,0
Deneme 6	a1	b2	m3	78	62	45	43	%20,1	%42,1	%44,5
Deneme 7	a1	b3	m1	87	78	69	67	%11,1	%21,3	%23,1
Deneme 8	a1	b3	m2	70	62	54	52	%11,0	%23,6	%25,3
Deneme 9	a1	b3	m3	53	47	38	38	%11,9	%27,4	%29,0
Deneme 10	a2	b1	m1	133	102	76	72	%23,2	%42,8	%45,8
Deneme 11	a2	b1	m2	116	87	61	57	%24,9	%47,4	%50,5
Deneme 12	a2	b1	m3	99	72	46	43	%27,2	%53,5	%56,7
Deneme 13	a2	b2	m1	108	87	70	67	%19,7	%35,8	%38,5
Deneme 14	a2	b2	m2	91	72	55	52	%21,1	%40,2	%43,0
Deneme 15	a2	b2	m3	74	57	39	37	%23,3	%46,8	%49,6
Deneme 16	a2	b3	m1	84	72	63	61	%14,0	%24,6	%27,5
Deneme 17	a2	b3	m2	66	57	48	47	%14,6	%27,8	%30,0
Deneme 18	a2	b3	m3	49	42	33	32	%15,8	%33,3	%35,4
Deneme 19	a3	b1	m1	128	95	69	64	%26,2	%46,4	%49,8
Deneme 20	a3	b1	m2	111	79	54	50	%28,4	%51,7	%55,3
Deneme 21	a3	b1	m3	94	64	38	35	%31,4	%59,0	%62,8
Deneme 22	a3	b2	m1	103	79	62	58	%23,2	%39,9	%43,6
Deneme 23	a3	b2	m2	86	64	47	44	%25,5	%45,5	%48,8
Deneme 24	a3	b2	m3	69	49	32	30	%28,8	%53,8	%57,3
Deneme 25	a3	b3	m1	79	64	56	51	%18,3	%29,3	%35,5
Deneme 26	a3	b3	m2	62	49	41	38	%20,1	%34,2	%38,2
Deneme 27	a3	b3	m3	44	34	25	24	%23,2	%42,8	%45,8



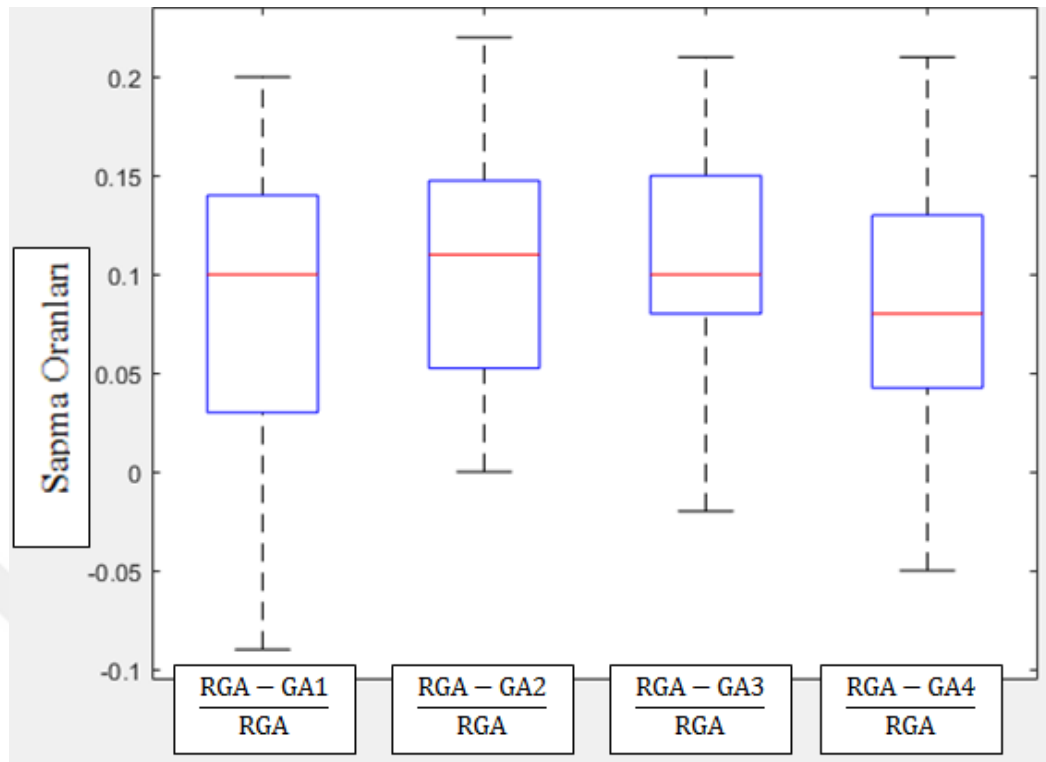
Şekil 3.15 Sezgisel algoritmaların deney düzeylerinde analizi

Şekil 3.11’de her deney düzeyi için sezgisel algoritmalar çalıştırılmıştır ve sezgisel algoritmaların deney düzeyleri için yoğun bir şekilde tepki verdiği gözlemlenmiştir. Ayrıca dört numaralı sezgiselde üretilen popülasyon maliyetlerinin orta nokta etrafında yoğunlaştığı gözlemlenmiştir.

Deneyin ikinci aşamasında ise sezgisel algoritmalar ve rastgele sayılar ile üretilen başlangıç popülasyonları ile GA çalıştırılmıştır ve sonuçlar Tablo 3.17’de gösterilmiştir. Bu tabloda, GA’ya başlangıç popülasyonu olarak rastgele sayılar gönderildiğinde üretilen sonuçlar ile sezgisel algoritmaların ürettiği başlangıç popülasyonlarını gönderildiğinde üretilen sonuçlar karşılaştırılmıştır. Tüm sezgisel algoritmaların ürettiği başlangıç popülasyonları GA’da 50000 iterasyon çalıştırıldığında rastgele sayılarla çalışan GA’dan daha başarılı sonuçlar vermiştir.

Tablo 3.17 Deney tasarımı GA sonuçları

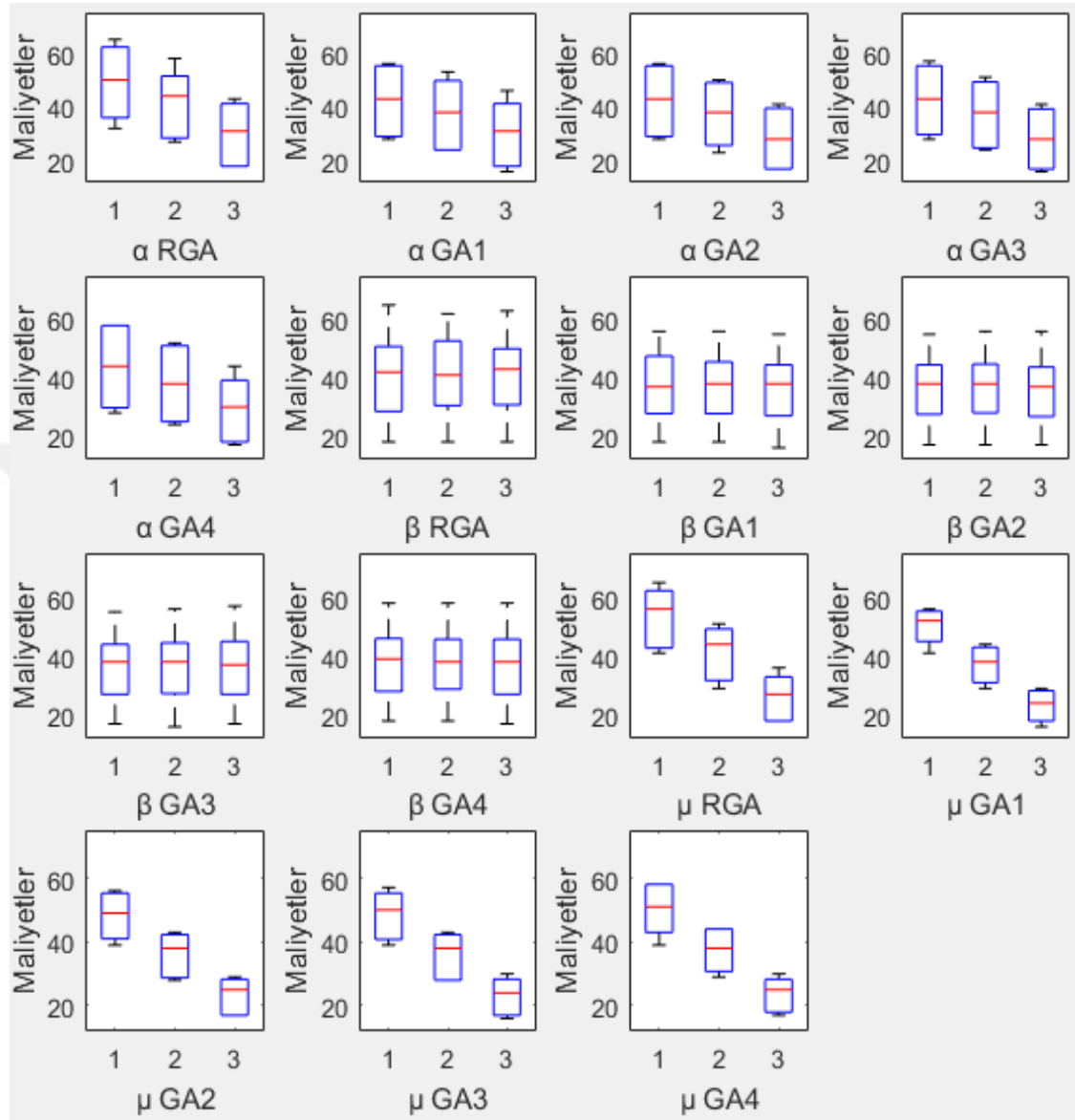
Denemeler	Katsayılar ve Düzeyleri			GA için İş Başına Ortalama Birim Maliyet					RGA ile Diğerlerinin Kıyaslanması			
	α	β	μ	RGA	GA1	GA2	GA3	GA4	(RGA-GA1)/RGA	(RGA-GA2)/RGA	(RGA-GA3)/RGA	(RGA-GA4)/RGA
Deneme 1	a1	b1	m1	65	56	55	55	58	%14,5	%15,0	%15,3	%10,2
Deneme 2	a1	b1	m2	49	44	43	42	44	%9,0	%10,8	%12,8	%9,5
Deneme 3	a1	b1	m3	36	29	29	30	29	%17,9	%19,3	%15,3	%18,5
Deneme 4	a1	b2	m1	62	56	56	56	58	%9,7	%9,8	%8,3	%6,1
Deneme 5	a1	b2	m2	51	43	43	43	44	%15,3	%15,5	%14,7	%13,7
Deneme 6	a1	b2	m3	32	29	29	29	30	%9,8	%8,8	%9,7	%5,5
Deneme 7	a1	b3	m1	63	55	56	57	58	%13,0	%10,8	%9,5	%7,9
Deneme 8	a1	b3	m2	50	43	42	43	44	%14,8	%16,7	%15,3	%12,8
Deneme 9	a1	b3	m3	36	28	28	28	28	%20,5	%22,3	%20,9	%20,9
Deneme 10	a2	b1	m1	56	52	49	50	52	%7,6	%12,1	%10,7	%7,2
Deneme 11	a2	b1	m2	43	37	38	38	39	%13,9	%12,8	%12,5	%9,5
Deneme 12	a2	b1	m3	27	24	26	24	25	%11,7	%5,2	%10,5	%8,0
Deneme 13	a2	b2	m1	58	53	50	49	51	%8,3	%13,4	%16,0	%12,5
Deneme 14	a2	b2	m2	44	38	38	38	38	%14,5	%14,0	%14,9	%13,5
Deneme 15	a2	b2	m3	29	24	25	25	25	%16,6	%14,7	%15,7	%15,0
Deneme 16	a2	b3	m1	50	49	49	51	51	%2,6	%1,4	%-1,6	%-0,9
Deneme 17	a2	b3	m2	44	38	37	37	38	%13,4	%15,3	%15,4	%12,7
Deneme 18	a2	b3	m3	27	24	23	24	24	%10,9	%13,1	%7,7	%9,6
Deneme 19	a3	b1	m1	42	46	39	40	44	%-8,6	%7,7	%5,5	%-4,6
Deneme 20	a3	b1	m2	29	31	28	28	31	%-4,5	%5,3	%4,6	%-5,5
Deneme 21	a3	b1	m3	18	18	17	17	18	%2,3	%4,5	%8,0	%3,1
Deneme 22	a3	b2	m1	41	42	41	39	39	%-3,7	%0,3	%5,5	%4,3
Deneme 23	a3	b2	m2	31	31	29	28	30	%-0,3	%5,9	%8,8	%1,7
Deneme 24	a3	b2	m3	18	18	17	16	18	%0,1	%4,4	%8,8	%1,3
Deneme 25	a3	b3	m1	43	41	41	41	40	%3,2	%4,4	%5,3	%6,6
Deneme 26	a3	b3	m2	32	29	28	28	29	%8,3	%10,4	%10,7	%7,4
Deneme 27	a3	b3	m3	18	16	17	17	17	%11,4	%7,2	%6,9	%6,7



Şekil 3.16 GA sonuçlarının RGA sonucu ile karşılaştırılması

Şekil 3.12’de rastgele üretilen başlangıç çözümlerinin GA sonuçları ile sezgisellerle oluşturulan başlangıç çözümlerinin GA sonuçları karşılaştırılmıştır. Oluşturulan kutu grafiklerinden görüldüğü gibi tüm sonuçlar rastgele oluşturulan başlangıç çözümlerinden ciddi manada ayrışma göstermiştir ve daha iyi sonuçlar vermiştir.

Şekil 3.13’de her deney düzeyi için sezgisellerin ürettiği başlangıç popülasyonları GA’da çalıştırılmıştır ve sonuçlar analiz edilmiştir. Şekillerden de görüldüğü gibi GA en büyük tepkiyi hazırlık sürelerinin maliyet katsayısındaki değişime göstermiştir. Bu da hazırlık sürelerinin çizelge maliyetleri üzerine etkisini göstermektedir.



Şekil 3.17 Sezgisel algoritmalar ve deney düzeylerinin GA’da analizi

3.6.2 Örnek Veri Setinin Çeşitli İterasyon Sayılarında Çalıştırılması

Firmada tecrübeli bir planlama uzmanı tarafından 137 işten oluşan bir çizelgeleme yapılmıştır. Tüm bilgileri hazır olarak verilen işlerden yapılan bu çizelge için planlama uzmanı yaklaşık beş saat gibi bir zaman harcamıştır. Çizelgeleme sonucunda bir iş başına maliyetin 61,3 olduğu bir sonuç ortaya çıkmıştır. Bu bölümde farklı iterasyon düzeyleri için geliştirilen modellerin tamamı entegre bir şekilde çalıştırılacaktır. Sonrasında çıkan sonuçlar planlama uzmanının oluşturduğu çizelge maliyeti ile

karşılaştırılarak geliştirilen modellerin performansı değerlendirilecektir. Ayrıca GA'nın farklı iterasyon düzeyleri için çözüm üretme zamanları da gösterilecektir.

Tablo 3.18 Gerçek verilerin GA'da çalıştırılması

		Maliyet (Birim İş İçin)	Bilgisayar Zamanı (Toplam Saniye)
50000 İterasyon	1. Çevrim	57,9	921
	2. Çevrim	57,8	943
	3. Çevrim	57,1	934
100000 İterasyon	1. Çevrim	57,5	1280
	2. Çevrim	56,9	1329
	3. Çevrim	55,2	1333
500000 İterasyon	1. Çevrim	54,5	4331
	2. Çevrim	55,8	4250
	3. Çevrim	53,7	4400

Tablo 3.18'de farklı iterasyon sayıları ile GA çalıştırıldığında ortaya çıkan sonuçlar görülmektedir. İterasyon sayıları arttıkça gerekli olan bilgisayar zamanı miktarı artsa da ortaya çıkan sonuçlarda ciddi maliyet iyileşmeleri görülmektedir.

Tablo 3.19 Gerçek verilerin mevcut çizelge performansı ile karşılaştırılması

	ORT Maliyet (Birim İş İçin)	STD	Mevcut Çizelge ile GA'nın Karşılaştırılması
50000 İterasyon	57,6	0,43	%6,0
100000 İterasyon	56,5	1,16	%7,8
500000 İterasyon	54,7	1,05	%10,8

Tablo 3.19'da ise çeşitli iterasyon düzeylerinde çalışan GA'nın ürettiği değerlerin ortalaması, standart sapması ve bu değerlerin gerçek çizelge maliyeti ile kıyaslaması görülmektedir. Her iterasyon düzeyi için de ortalama maliyetler ve standart sapmalar beklenen şekilde düşük çıkmıştır. Ayrıca bu sonuçlar gerçek çizelge maliyeti ile kıyaslandığında da ciddi iyileşmeler olduğu ve kısa süreler içerisinde uygun çizelgelerin yapılabileceği görülmüştür. Sonuç olarak yapılan bu çalışma, planlama uzmanının tecrübelerine bağımlı olmaksızın kısa süreler içerisinde uygunluk değeri gayet iyi olan çizelgeler oluşturabilmektedir.

4. BÖLÜM

TARTIŞMA - SONUÇLAR VE ÖNERİLER

Hazırlık süreleri üretim sistemlerinin verimliliğini düşüren, fazladan makine ve işçilik saati harcanmasına sebep olan, fire ve hurda gibi makine hazırlıkları kayıplarına sebep olan ama aynı zamanda tamamen yok edilmesi de mümkün olmayan üretim içi süreçlerdir. Yapılan bu çalışmada aralarında sıra bağımlı hazırlık süreleri ilişkisi olan işler tek makinede dengelenmiş ve çizelgelenmiştir. Kurgulanan problemin temel yapısı sıra bağımlı hazırlık sürelerini de göz önünde bulundurarak toplam ağırlıklı erken bitirmeyi, toplam ağırlıklı gecikmeyi ve ağırlıklandırılmış hazırlık sürelerini minimize etmektir. Bilindiği gibi tek makine çizelgeleme problemleri, çizelgelemenin temel mantığını oluşturmada hala önemli yer tutmaktadır ve literatürde birçok çalışmaya konu olmaya devam etmektedir.

Bu tezde ele alınan problem bir firmada yürütülmekte olan gerçek bir sistemi ve o firmada üretilen gerçek işleri modellemek üzere kurgulanmıştır. Gerçek hayat problemlerinin çok fazla değişkeni bir arada içermesi, çok sayıda duruma çözüm getirme isteği problemi NP-zor sınıfına sokmaktadır ve yapılan çalışmanın önemini göstermektedir.

İlgili problem için geliştirilen matematiksel programlama modeli küçük ölçekli problemlerle denenmiştir. Model, 10 iş için hızlı bir şekilde çözüm üretebilirken, iş sayısı 15'e çıkarıldığında etkin sürede çözüm üretememiştir ve bu durum sezgisel yöntemlerin kurgulanmasında etkili olmuştur. Dolayısıyla probleme çözüm üretebilmek için GA kullanmaya karar verilmiştir ve GA için başlangıç çözümleri üretecek dört adet sezgisel yöntem geliştirilmiştir. Geliştirilen sezgisel algoritmalar GA'nın rastgele üretilmiş başlangıç popülasyonları ile vakit kaybetmesini engellemiştir ve programa çok ciddi hız katkısı sunmuştur.

Geliştirilen modeller firmadan alınan ve tecrübeli bir planlamacı tarafından yapılan ve 137 adet gerçek iş içeren çizelge kullanılarak test edilmiştir. Yapılan denemelerde 50000 iterasyon sonucunda ve çok kısa süreler içerisinde geliştirilen modeller tecrübeli planlamacının yaptığı çizelgeden daha iyi sonuçlar vermeyi başarmıştır. İterasyon sayıları artırıldıkça çok daha güzel sonuçlar elde edilmiş ve başarıya ulaşılmıştır.

Bu tezde problem tek makine için ifade edilmiştir. İlerde yapılacak çözüm önerilerinde paralel makine, akış tipi ve atölye tipi çizelgeleme gibi çok makineli ortamları içeren uygulamalar yapılabilir. Ayrıca bu tezde işlerin deterministik olduğu kabul edilmiştir. Fakat gerçek hayat problemlerinde işler stokastik karakterler taşıyabilmektedir ve çizelgelerin hızla bozulmasına sebep olabilmektedir. İleriki çalışmalarda makine arızaları, acil işler ve talebi geri çevrilemeyecek müşteri gibi değişken durumlar göz önünde bulundurularak yeni modeller geliştirilebilir.

KAYNAKLAR

1. Abdul-Razaq, T., S., Potts, C., N., Van Wassenhove, L., N., 1990. A Survey of algorithms for the single machine total weighted tardiness scheduling problem, **Discrete Applied Mathematics**, **26**: 235-253.
2. Allahverdi, A., Gupta, J. N. D., Aldowaisan, T., 1999. A review of scheduling research involving setup consideration, **OMEGA**, **27**: 219-239.
3. Allahverdi, A., Ng, C. T., Cheng, T. C. E., Kovalyov, M. Y., 2008. A survey of scheduling problems with setup times or costs, **European Journal of Operational Research**, **187**: 985-1032.
4. Allahverdi, A., 2015. The third comprehensive survey on scheduling problems with setup times/costs, **European Journal of Operational Research**, **246**: 345-378.
5. Ak, B., 2012. Seçilmiş Bir Endüstriyel Tesiste Üretim Çizelgeleme Problemi ve Genetik Algoritma ile Optimizasyonu, Yüksek Lisans Tezi, Samsun, 140 s.
6. Baker, K., R., Trietsch, D., 2009. Principles of Sequencing and Scheduling. Wiley, New Jersey, 488 pp.
7. Baker, K., R., Schrage, L., 1978. Finding an optimal sequence by dynamic programming: An extension to precedence-related tasks, **Operations Research**, **26**: 111-120.
8. Bigras L.P., Gamache M., Savard G., 2008. The time-dependent traveling salesman problem and single machine scheduling problems with sequence dependent setup times, **Discrete Optimization**, **5**: 685-699.
9. Bolat, B., Erol, K. O., İmrak, C. E., 2004. Mühendislik uygulamalarında genetik algoritmalar ve operatörlerin işlevleri. **Mühendislik ve Fen Bilimleri Dergisi**, 264-271.
10. Boysen, N., Fliedner, M., Scholl, A., 2009. Sequencing mixed-model assembly lines: Survey, classification and model critique. **European Journal of Operation Research**, **192**: 349-373.
11. Ceran, G., 2006. Esnek Akış Tipi Çizelgeleme Problemlerinin Veri Madenciliği ve Genetik Algoritma Kullanılarak Çözülmesi, Yüksek Lisans Tezi, Konya, 134 s.
12. Chen, Z. L., 1997. Scheduling with batch setup times and earliness-tardiness penalties, **European Journal of Operational Research**, **96**: 518-537.
13. Cheun, C., Bulfin, R., L., 1990. Scheduling unit processing time jobs on a single machine with multiple criteria, **Computers and Operations Research**, **17**: 1-7.

14. Coleman, B. J., 1992. A simple model for optimizing the single machine early/tardy problem with sequence-dependent setups, **Production and Operation Management, 1**: 225-228.
15. Dağ, S., 2012. Akış Tipi Çizelgeleme Problemlerinin Sezgisel Yöntemlerle Optimizasyonu, Doktora Tezi, İstanbul, 162 s.
16. Doğan, N., 2013. Sıralamaya Bağımlı Hazırlık Sürelerinin Olduğu Permütasyonlu Akış Tipi Üretim Çizelgeleme Problemi İçin Bir NSGA-II Algoritması, Yüksek Lisans Tezi, Eskişehir, 67 s.
17. Elmaghraby, S., E., 1968. The one machine scheduling problem with delay cost, **Journal of Industrial Engineering, 19**: 105-108.
18. Emmons, H., 1969. One machine sequencing to minimize certain functions of job tardiness, **Operations Research, 17**: 701-715.
19. Eren, T., Guner E., 2006. A bicriteria scheduling with sequence-dependent setup times, **Applied Mathematics and Computation, 179**: 378-385.
20. Eren, T., Güner, E., 2002. Tek ve paralel makineli problemlerde çok ölçütlü çizelgeleme problemleri için bir literatür taraması. **Gazi Üniv. Müh. Mim. Fak. Der, 17**: 37-69.
21. Graham, R., Lawler, E., Lenstra, J., Rinnoy, K., A., 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. **Annals of Discrete Mathematics, 5**: 287-326.
22. Gupta, S. R., Smith, J. S., 2006. Algorithms for single machine total tardiness scheduling with sequence dependent setups, **European Journal of Operational Research, 175**: 722-739.
23. Hallah, R. M., 2007. Minimizing total earliness and tardiness on a single machine using a hybrid heuristic. **Computers And Operations Research, 34**: 3126–3142.
24. Haupt, R. L., Haupt, S. E., 2004. Practical Genetic Algorithms. John Wiley & Sons, New Jersey, 253 pp
25. Held, M., Karp, R., M., 1962. A dynamic programming approach to sequencing problems, **SIAM Journal, 10**: 196-210.
26. Holland, J. H., 1975. Adaptation in natural and artificial systems, **University of Michigan Press**, Ann Arbor.

27. Kaya, S., Saraç, T., 2013. Plastik enjeksiyon makinelerinin vardiya bazında çizelgelenmesi problemi için bir hedef programlama modeli. **Endüstri Mühendisliği Dergisi**, **24 (1-2)**: 12-26.
28. Khorshidian, H., Javadian, N., Zandieh, M., Rezaeian, J., Rahmani, K., 2011. A genetic algorithm for JIT single machine scheduling with preemption and machine idle time. **Expert Systems With Applications**, **38**: 7911–7918.
29. Koçanlı, M., M., Aydınbeyli, Y., E., Saraç, T., 2012. Eti şirketler grubunda üretim çizelgeleme problemi için hedef programlama modeli ve genetik algoritma. **Endüstri Mühendisliği Dergisi**, **23**: 4-21.
30. Kolahan, F., Liang, M., 1998. An adaptive TS approach to JIT sequencing with variable processing times and sequence-dependent setups, **European Journal of Operational Research**, **109**: 142-159.
31. Koulamas, C., 1994. The total tardiness problem: Review and extensions, **Institute for Operations Research and the Management Sciences**, **42**: 1025-1041.
32. Koulamas, C., 1996. Single-machine scheduling with time windows and earliness/tardiness penalties, **European Journal of Operational Research**, **91**: 190-202
33. Küçük, B., 2010. Özdeş Paralel Yapılı Bir Üretim Sisteminin Karınca Koloni Algoritması ile Çizelgelenmesi, Doktora Tezi, İstanbul, 97s.
34. Lee, S. M., Asllani, A. A., 2004. Job scheduling with dual criteria and sequence-dependent setups: Mathematical versus genetic programming, **Omega**, **32**: 145-153.
35. Li, C., L., Cheng, T., C., E., Chen, Z., L., 1995. Single-Machine scheduling to minimize the weighted number of early and tardy agreeable jobs, **Computers and Operations Research**, **22**: 205-219.
36. McMullen, P., R., Tarasewich, P., 2003. A beam search heuristic method for mixed-model scheduling with set-up, **Int. J. Production Economics**, **96**: 273-283.
37. Muştu, S., 2013. Öğrenme Etkili ve Farklı Geliş Zamanlı Tek Makineli Çizelgelemede Toplam Ağırlıklı Gecikme Probleminin Çözümü, Yüksek Lisans Tezi, Kırıkkale, 149 s.

38. Özdemir, M., H., 2017. Kısıtlandırılmış Teslim Tarihli ve Sıraya Bağımlı Hazırlık Süreli Tek Makine Çizelgeleme Problemlerinde Erkenlik ve Geçlik Toplamının Enküçüklenmesi, Yüksek Lisans Tezi, Ankara, 109s.
39. Pinedo, L., M., 2008. Scheduling Theory, Algorithms and Systems. Springer Science+Business Media, New York, 663 pp.
40. Potts, C., N., Van Wassenhove, L., N., 1991. Single machine tardiness sequencing heuristics, **IIE Transactions**, **23**: 346-354.
41. Rabadi, Mollaghasemi, M., Anagnostopoulos, G. C., 2004. A branch-and-bound algorithm for the early/tardy machine scheduling problem with a common due-date and sequence-dependent setup time, **Computers and Operations Research**, **31**: 1727-1751.
42. Rachamadugu, R., M., V., 1987. A note on the weighted tardiness problem, **Operations Research**, **35**: 450-451.
43. Rinnooy Kan, A., H., G., Lageweg, B., J., Lenstra, J., K., 1975. Minimizing total costs in one machine scheduling, **Operations Research**, **23**: 908-927.
44. Schaller, J., Gupta, J. N. D., Vakharia A. J., 2000. Scheduling a flowline manufacturing cell with sequence dependent family setup times, **European Journal of Operational Research**, **125**: 324-339.
45. Schild, L., Fredman, K., R., 1961. Scheduling tasks with linear loss functions, **Management Science**, **7**: 280-285.
46. Sen, T., Sulek, J., M., Dileepan, P., 2003. Static scheduling research to minimize weighted and unweighted tardiness: A state-of-the-art survey, **International Journal of Production Economics**, **83**: 1-12.
47. Sen, T., Austin, L., M., Ghandforoush, P., 1983. An algorithm for the single machine sequencing problem to minimize total tardiness, **IIE Transactions**, **15**: 363-366.
48. Sevaux, M., Peres S. D., 2003. Genetic algorithms to minimize the weighted number of late jobs on a single machine. **European Journal of Operational Research**, **151**: 296–30
49. Shwimer, J., 1972. On the n-job one machine sequence-independent scheduling problem with tardiness penalties: A branch-and-bound solution, **Management Science**, **18**: 301-313.

50. Smith, W., E., 1956. Various optimizers for single-stage production. **Naval Research Logistics Quarterly**, **3**: 59-66.
51. Tan, K. C., Narasimhan, R., 1997. Minimizing tardiness on a single processor with sequence-dependent setup times: A simulated annealing approach, **Omega**, **25**: 619-634.
52. Tansel, B., C., Sabuncuoğlu, I., 1997. New insights on the single machine total tardiness problem, **Journal of the Operational Research Society**, **48**: 82-89.
53. Valente, J.M.S., Alves, R.A.F.S., 2005. Filtered and recovering beam search algorithms for early/tardy scheduling problem with no idle time, **Computers and Industrial Engineering**, **48**: 363-375.
54. Van Wassenhove, L., N., Gelders, F., 1980. Solving a bicriterion scheduling problem, **European Journal of Operational Research**, **4**: 42-48.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı: Orhun KARACAN

Uyruğu: Türkiye (TC)

Doğum Tarihi ve Yeri: 20 Ocak 1987, Nevşehir

Medeni Durumu: Evli

Tel: +90 543 325 99 23

Fax:

email: orhunkaracan@gmail.com

Yazışma Adresi: Bahçelievler Mahallesi. Zeybek Sitesi No:10 TALAS /KAYSERİ

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Lisans	Uludağ Üniversitesi M.M.F. Endüstri Mühendisliği	2009
Lise	Nuh Mehmet Küçükçalık Anadolu Lisesi, Kayseri	2004

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2011- Halen	Erciyes Çelik Halat A.Ş.	Üretim Şefi

YABANCI DİL

İngilizce

YAYINLAR