

**DİNAMİK ÇOK AMAÇLI ÇİZELGELEME  
PROBLEMLERİ İÇİN SEVK ETME  
KURALLARININ GELİŞTİRİLMESİ ve  
ANALİZİ**

**Yüksek Lisans Tezi**

**Ozan BAHADIR**

**Eskişehir 2017**

**DİNAMİK ÇOK AMAÇLI ÇİZELGELEME PROBLEMLERİ İÇİN SEVK  
ETME KURALLARININ GELİŞTİRİLMESİ ve ANALİZİ**

**Ozan BAHADIR**

**YÜKSEK LİSANS TEZİ**

**Endüstri Mühendisliği Anabilim Dalı  
Danışman: Doç. Dr. Gürkan ÖZTÜRK**

**Eskişehir  
Anadolu Üniversitesi  
Fen Bilimleri Enstitüsü  
Mayıs, 2017**

*Bu Tez Çalışması BAP Komisyonunca kabul edilen 1703F080 nolu proje kapsamında desteklenmiştir.*

## JÜRİ VE ENSTİTÜ ONAYI

Ozan BAHADIR 'ın “**Dinamik Çok Amaçlı Çizelgeleme Problemleri İçin Sevk Etme Kurallarının Geliştirilmesi ve Analizi**” başlıklı tezi 29/05/2016 tarihinde, aşağıdaki jüri tarafından “Anadolu Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği” nin ilgili maddeleri uyarınca Endüstri Mühendisliği Anabilim dalında Yüksek Lisans tezi olarak kabul edilmiştir.

	<u>Unvanı-Adı-Soyadı</u>	<u>İmza</u>
Üye (Tez Danışmanı) :	Doç. Dr. Gürkan ÖZTÜRK	.....
Üye :	Doç. Dr. Onur KAYA	.....
Üye :	Doç. Dr. Servet HASGÜL	.....

.....  
Enstitü Müdürü

## ÖZET

### DİNAMİK ÇOK AMAÇLI ÇİZELGELEME PROBLEMLERİ İÇİN SEVK ETME KURALLARININ GELİŞTİRİLMESİ ve ANALİZİ

Ozan BAHADIR

Endüstri Mühendisliği Anabilim Dalı

Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Mayıs, 2017

Danışman: Doç. Dr. Gürkan ÖZTÜRK

Bu çalışmada, çizelgeleme problemlerinde kullanılan bileşik sevk etme kuralları çıkarımı için Benzetim ve Gen İfade Programlamayı kullanan yeni bir yöntem önerilmiştir. Önerilen bu yöntem ile dinamik çizelgeleme problemleri için sistemin özelliklerini dikkate alan yeni sevk etme kuralları geliştirilmiştir. Bu kuralların çıkarım aşamasında Gen İfade Programlamadaki iki farklı yöntem uygulanmıştır. Bu yöntemlerin birinde üretim parametreleri temelli çok genli kromozomlar kullanılırken diğerinde ise literatürdeki kuralları doğrusal ve doğrusal olmayan şekilde birleştiren otomatik tanımlanmış fonksiyon yapıları kromozomlar kullanılmıştır. Bu çalışmadaki benzetim modeli, çok amaçlı dinamik esnek atölye tipi çizelgeleme problemleri için tasarlanmasına karşın diğer çizelgeleme problemlerine de kolayca uygulanabilir bir yapıya sahiptir. Çok amaçlı bakış açısına göre, önerilen yöntemlerden elde edilen sevk etme kuralları ile literatürden seçilen kurallar karşılaştırıldığında, türetilen kuralların performanslarının diğer tüm kuralların performanslarından daha iyi olduğu görülmektedir. Ayrıca, bu iki yöntem ile türetilen kurallar karşılaştırıldığında, ADF kurallarının çok genli kurallardan daha iyi olduğu gözlemlenmiştir. Sunulan yöntem ve türetilen sevk etme kuralları sağlam olduğu için optimizasyon yöntemleri çok zor olan gerçek, büyük ve dinamik tüm çizelgeleme problemlerine kolayca uygulanabilir.

**Anahtar Kelimeler:** Dinamik atölye tipi çizelgeleme, Sevk etme kuralları, Benzetim, Gen ifadesi programlama, Çok amaçlı optimizasyon.

## ABSTRACT

### DEVELOPMENT AND ANALYSIS OF DISPATCHING RULES FOR DYNAMIC MULTI-OBJECTIVE SCHEDULING PROBLEMS

Ozan BAHADIR

Department of Industrial Engineering Program

Anadolu University, Graduate School of Sciences, May, 2017

Supervisor: Assoc. Prof. Dr. Gürkan ÖZTÜRK

In this work, a new method that uses Simulation and Gene Expression Programming is proposed for composite dispatching rules extraction for scheduling problems. In the suggested method, new dispatching rules that take into account the features of the production system have been developed for dynamic scheduling problems. Two different methods of Gene Expression Programming have been applied in the development of these rules. One of these methods uses multi-genic chromosomes based on production parameters whereas the other uses chromosomes with automatical defined function (ADF) that combine rules in the literature in a way linear and nonlinear. Although the simulated model is designed for multi objective dynamic flexible job shop scheduling problems, it has a structure which can easily implement for the other scheduling problems. According to the multi-objective view point, the dispatching rules obtained from the proposed methods are compared with the selected rules from the literature, and it is seen that the performance of extracting rules is significantly better than all other rules in literature. Also, when it is compared two kind of extracting rules, it was observed that ADF rules are better than multi-genic rules. Because the presented method and the achieved dispatching rules are robust, they can be applied for all real, big and dynamic scheduling problems which their optimization models have high complexity.

**Keywords:** Dynamic job shop scheduling, Dispatching rules, Simulation, Gene expression programming, Multi-Objective optimization.

## TEŐEKKÜR

Lisans ve Yüksek Lisans hayatım boyunca benden desteęini hiçbir zaman esirgemeyen, akademisyen kimlięi ve insani yönleriyle kendime örnek aldığım hocam Sayın Doç. Dr. Gürkan ÖZTÜRK'e,

Yüksek Lisans eğitim sürecimde Ciol laboratuvarını beraber kullandığım ve bu çalışma kapsamında bana yardımcı olan değerli arkadaşım Aydın TEYMOURIFAR'a,

Eğitim-öğretim hayatım boyunca üzerimde emeęi olan hocalarıma, destekleri ile yanımda olan arkadaşlarıma,

Beni yetiştirip bu günlere getiren ve aldığım kararlar ne olursa olsun arkamda duran annem Şehrinaz BAHADIR ve babam Hüseyin BAHADIR' a,

Sonsuz teşekkürlerimi sunarım.

Ozan BAHADIR

MAYIS 2017

29/05/2017

## **ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ**

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilemeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Anadolu Üniversitesi tarafından kullanılan “bilimsel intihal tespit programı” ile tarandığını ve hiçbir şekilde “intihal içermediğini” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

.....

Ozan BAHADIR

## İÇİNDEKİLER

### Sayfa

BAŞLIK SAYFASI .....	i
JÜRİ VE ENSTİTÜ ONAYI.....	ii
ÖZET .....	iii
ABSTRACT.....	iv
TEŞEKKÜR .....	v
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ.....	vi
İÇİNDEKİLER .....	vii
TABLolar DİZİNİ .....	ix
ŞEKİLLER DİZİNİ.....	x
SİMGELER VE KISALTMALAR DİZİNİ.....	xi
1. GİRİŞ .....	1
2. ÇİZELGELEME VE SEVKETME KURALLARI .....	9
2.1. Çizelgeleme .....	9
2.1.1. Çizelgelemede kullanılan temel kavramlar .....	10
2.1.2. Çizelgelemenin sınıflandırılması .....	10
2.2. Sevk Etme Kuralları.....	14
2.2.1. Sevk etme kurallarının sınıflandırılması .....	14
2.3. EATÇ Problemlerinde Sevk Etme Kuralları .....	16
3. KULLANILAN YÖNTEM.....	20
3.1. Gen İfade Programlama (GEP).....	20
3.1.1. Kullanılan Kavramlar .....	20
3.2. Çalışmada Kullanılan GEP Tasarımı .....	24
3.2.1. Çok genli statik kurallar .....	25
3.2.2. ADF gen yapılı statik kurallar .....	26



<b>3.3. Deneysel Çalışma Ortamı Tasarımı .....</b>	<b>26</b>
<b>3.3.1. Benzetim ortamı tasarımı.....</b>	<b>26</b>
<b>3.3.2. GEP parametreleri .....</b>	<b>28</b>
<b>3.4. Deneysel Sonuçlar ve Analiz .....</b>	<b>32</b>
<b>3.4.1. Literatürden seçilen kurallar.....</b>	<b>32</b>
<b>3.4.2. Türetilen kurallar .....</b>	<b>32</b>
<b>3.4.3. Cmax için sonuçlar .....</b>	<b>33</b>
<b>3.4.4. Ortalama gecikme (Tort) için sonuçlar .....</b>	<b>35</b>
<b>3.4.5. Ortalama akış (Fort) için sonuçlar.....</b>	<b>37</b>
<b>3.4.6. Çok amaçlı fonksiyon için sonuçlar .....</b>	<b>39</b>
<b>4. SONUÇLAR ve TARTIŞMA .....</b>	<b>42</b>
<b>KAYNAKÇA.....</b>	<b>44</b>
<b>EK-1 Benzetim Algoritmaları</b>	
<b>EK-2 Least Waiting Time Algoritması</b>	
<b>EK-3 %50 ve %100 Esneklik Seviyelerinde Cmax İçin Analizler</b>	
<b>EK-4 %50 ve %100 Esneklik Seviyelerinde Tort İçin Analizler</b>	
<b>EK-5 %50 ve %100 Esneklik Seviyelerinde Fort İçin Analizler</b>	
<b>EK-6 %50 ve %100 Esneklik Seviyelerinde Çok Amaçlı İçin Analizler</b>	
<b>ÖZGEÇMİŞ</b>	

## TABLolar DİZİNİ

	<b><u>Sayfa</u></b>
<b>Tablo 1.1.</b> Makale karşılaştırması .....	8
<b>Tablo 2.1.</b> Sevk etme kuralları notasyonu.....	15
<b>Tablo 2.2.</b> Örnek problemdeki işlerin verileri.....	17
<b>Tablo 2.3.</b> Örnek problem operasyon verileri .....	17
<b>Tablo 2.4.</b> FIFO ve EDD sonuçlar .....	19
<b>Tablo 3.1.</b> Çok genli terminal kümesi .....	25
<b>Tablo 3.2.</b> Çok genli fonksiyon kümesi .....	25
<b>Tablo 3.3.</b> ADF gen yapılı terminal kümesi.....	26
<b>Tablo 3.4.</b> EATÇ problem parametreleri .....	27
<b>Tablo 3.5.</b> GEP parametreleri.....	28
<b>Tablo 3.6.</b> Türetilen kurallar .....	33
<b>Tablo 3.7.</b> Cmax sonuçlar .....	34
<b>Tablo 3.8.</b> Cmax Tukey test sonuçları .....	35
<b>Tablo 3.9.</b> Tort sonuçlar .....	36
<b>Tablo 3.10.</b> Tort Tukey test sonuçları .....	37
<b>Tablo 3.11.</b> Fort sonuçlar .....	38
<b>Tablo 3.12.</b> Fort Tukey test sonuçları .....	39
<b>Tablo 3.13.</b> Çok amaçlı sonuçlar.....	40
<b>Tablo 3.14.</b> Çok amaçlı test sonuçları.....	41

## ŞEKİLLER DİZİNİ

	<b><u>Sayfa</u></b>
Şekil 2.1. Üretim Sistemindeki Bilgi Akışı (Pinedo 2008) .....	9
Şekil 2.2. Esnek akış tipi (Pinedo, 2005).....	11
Şekil 2.3. Atölye tipi (Pinedo, 2005) .....	12
Şekil 2.4. FIFO için Gantt şeması.....	18
Şekil 2.5. EDD kuralı Gantt şeması.....	19
Şekil 3.1. İfade ağacı .....	21
Şekil 3.2. Örnek ağaç yapısı .....	22
Şekil 3.3. Statik bağlı kromozom ifade ağacı .....	23
Şekil 3.4. Homeotik gen ifade ağacı.....	23
Şekil 3.5. ADF 'li kromozom (Candida, 2006).....	24
Şekil 3.6. ADF 'li kromozom ifade ağacı (Candida, 2006) .....	24
Şekil 3.7. Çaprazlama sonrası ağaç derinliği değişimi (Candida, 2006) .....	30
Şekil 3.8. GEP benzetim ortamı akış şeması .....	31
Şekil 3.9. %95 güven seviyesinde EATÇ-20 için Cmax güven aralığı .....	34
Şekil 3.10. %95 güven seviyesinde EATÇ-20 için Tort güven aralığı.....	36
Şekil 3.11. %95 güven seviyesinde EATÇ-20 için Fort güven aralıkları.....	38
Şekil 3.12. %95 güven seviyesinde EATÇ-20 için çok amaçlı güven aralığı .....	40

## SİMGELER VE KISALTMALAR DİZİNİ

- $d_j$**  : Teslim zamanı (due date)
- $C_{\max}$**  : Toplam tamamlanma zamanı (makespan)
- $L_j$**  : İşlerin sapması (lateness)
- $p_{ij}$**  : İşlem süresi (processing time)
- $Re$**  : İşin kalan zamanı (remaining time)
- $r_j$**  : Serbest kalma zamanı (release data)
- $T_j$**  : İşlerin gecikmesi (tardiness)
- $w_j$**  : İş ağırlığı (works weight)
- $T_{ort}$**  : Geciken işlerin ortalaması
- $F_{ort}$**  : Ortalama akış
- $w_j$**  : İş ağırlığı (works weight)
- $CR$**  : Kritik oran (critical ratio)
- ATÇ** : Atölye tipi çizelgeleme (Job shop scheduling )
- EATÇ** : Esnek atölye tipi çizelgeleme (Flexible job shop scheduling )
- FSS** : Akış tipi çizelgeleme (Flow shop scheduling )
- OSS** : Açık atölye tipi çizelgeleme (Open shop scheduling )
- GA** : Genetik algoritma
- GP** : Genetik programlama
- GEP** : Gen ifade programlama
- ADF** : Otomatik tanımlanmış fonksiyon (Automatically Defined Functions)
- VNS** : Değişken komşuluk arama (Variable neighborhood search)
- SB** : Darboğaz değiştirme (Shifting bottleneck)
- EDD** : En erken teslim zamanı (Earliest due date )
- FIFO** : İlk giren ilk çıkar (first in first out)
- SPT** : En kısa işlem süresi (shortest processing time)
- SLK** : Boş zaman (slack time)
- TWK** : Toplam iş içeriği (total work)
- $\alpha$**  : Makinenin durumunu
- $\beta$**  : İşlemin detayı ve kısıtlar
- $\gamma$**  : Amaç fonksiyonunu göstergesidir
- SPR** : Basit öncelik kuralları (Simple priority rules)

**CPR** : Bileşik öncelik kuralları (Composite priority rules)

**LP** : Doğrusal Programlama

## 1. GİRİŞ

Teknolojinin gelişmesiyle globalleşen dünyada işletmeler arası rekabet artmıştır. İşletmeler bu rekabet ortamında ayakta kalabilmek için kısıtlı kaynaklarını en etkin şekilde kullanmak zorundadırlar. Kaynakların etkin kullanımı ancak iyi bir plan ve çizelgeleme ile gerçekleştirilebilmektedir. Çizelgeleme, hizmet ve imalat sektöründe belirli zaman aralıklarında düzenli olarak kullanılan karar verme sürecidir. İstenilen zaman aralıklarında kaynakların görevlere tahsis edilmesinde bir ya da daha fazla amaç optimize edilmeye çalışılmaktadır. Bir şirkette yapılması gereken etkinliklere sınırlı kaynaklar ayırmak için planlama ve zamanlama işlevleri, matematiksel teknikler ve buluşsal yöntemlerle yapılmalıdır. Bu kaynak tahsisi, şirketin hedeflerini optimize edecek ve bu hedeflere ulaşacak şekilde yapılmalıdır. Kaynaklara örnek olarak bir atölyedeki makineler, havalimanındaki pistler, bir şantiyedeki mürettebatlar veya bilgi işlem ortamındaki işleme birimleri verilebilir. Faaliyetlere ise bir atölye çalışması, havalimanında kalkış ve iniş, inşaat projelerinin aşamaları, bilgisayar programları örnek olarak verilebilir [1] [2]. Özellikle üretim sektöründe kaynakların etkin kullanımı işletmelere büyük avantaj sağlamaktadır. Çünkü kaynaklar kısıtlıdır ve bu kısıtlı kaynaklar ile üretilebilecek fazladan her ürün değerli ve anlamlıdır.

Üretim sistemleri, çeşitli faktörlerle karakterize edilebilir: kaynakların veya makinelerin sayısı, özellik ve konfigürasyonu, otomasyon seviyesi, malzeme taşıma sistemi türü vb. Tüm bu farklılıklar, çok sayıda farklı planlama ve çizelgeleme modeline sebep olmaktadır. Üretim sürecinde, bir iş, çeşitli makinelerde yapılması gereken tek bir operasyon ya da bir operasyon kümesi olabilir [2]. Bir işletmede yapılacak işler başlangıçta bilinebilir veya herhangi bir t anında sisteme yeni işler gelebilir. Birinci durumdaki üretim ortamı statik, ikinci durumdaki üretim ortamı ise dinamiktir. Gerçek hayatta yapısı gereği dinamik üretim ortamı daha yaygın görülmektedir. Çizelgeleme problemlerinde kullanılan üretim sistemleri farklılık göstermektedir. En çok bilinen ve kullanılan üretim sistemlerinden birisi Atölye Tipi Üretimdir. Bu sisteme gerçek hayatta oldukça sık rastlanılmaktadır. N iş ve m makineden oluşan bu sistemde, her bir iş belli bir sırada birçok makine tarafından işlenebilir. Bu işlem sırasında devir daim yapılmamaktadır [2]. Atölye Tipi Çizelgeleme Problemleri, problem boyutu arttıkça olası çizelge sayısı üstel olarak arttığından NP-Zor problem sınıfına girmektedir [3]. Bu çizelgeleme probleminde en çok kullanılan amaç, toplam tamamlanma süresini minimize etmektir. Bu problemin genel gösterimi  $J//C_{max}$  şeklindedir. Bu gösterimdeki

Cmax toplam tamamlanma süresini temsil eder [4]. Artan rekabet ortamının getirdiği değişiklikler, aynı anda birçok amacın optimize edilmesi gereksinimini ortaya çıkarmıştır. Atölye Tipi Çizelgeleme ortamlarında bir makinede oluşan darboğaz kısıtları nedeniyle ilgili makine sayısı arttırılmıştır. Böylece paralel makineli sistemler ve Esnek Atölye Tipi üretim ortamı ortaya çıkmıştır. Bu problem türü Atölye Tipi Çizelgeleme Problem 'inin genel halidir. En az bir operasyonu üretebilecek yetkinlikte birden fazla makine bulunan üretim sistemleridir. Ayrıca EATÇ, ATÇ 'den daha karmaşık olduğu için NP-Zor problem sınıfına girmektedir [5]. Bu yüzden kesin çözüm veren yöntemleri kullanmak yerine sezgisel algoritmalar ve benzetim yöntemi kullanılarak bu probleme çözüm aranmıştır.

ATÇ ve EATÇ sistemlerinin bulunduğu işletmelerde sevk etme kuralları yaygın olarak kullanılmıştır. Sevk etme kuralları, makinelerin mevcut işleri bittiğinde kuyruklarındaki işlerden birini seçmek için kullanılır [6]. Literatürde birçok klasik kural vardır ve bu kurallar bazı amaçlar için oldukça iyi sonuçlar vermektedirler. Örneğin FIFO kuralı toplam tamamlanma zamanının minimize edilmek istendiği yerlerde en iyi sonucu verir. Fakat literatürde çok amaçlı problemlerde her amaç için iyi çözüm veren klasik bir kural bulunmamaktadır. Bu yüzden daha karmaşık kuralların türetilmesi ihtiyacı doğmuştur. Holthaus ve Rajendran 1997 yılında literatürdeki klasik kuralların lineer birleşiminden oluşan yeni kurallar türetmişlerdir. Bunlardan bir tanesi PT+WINQ kuralıdır ve bu kural ortalama akış için oldukça iyi sonuçlar vermektedir [7]. Tay ve Ho üretimde bulunan parametrelerin (işlem zamanı, teslim tarihi vb.) doğrusal ve doğrusal olmayan birleşimlerini, Genetik Programlama kullanarak evrimleştirip çok amaçlı çizelgeleme problemlerinde tüm amaçlar için iyi çözüm sağlayan bileşik kurallar türetmişlerdir [5].

Bu tez çalışmasında dinamik çok amaçlı çizelgeleme problemlerinin çözümü için Gen İfadesi Programlama kullanılarak yeni kurallar türetilmiştir. Bunun için farklı çizelgeleme problemlerini içeren bir benzetim ortamı tasarımı yapılmıştır. Bu benzetim ortamında özellikle gerçek hayatta sıkça karşımıza çıkan EATÇ problemi üzerinde durulmuş ve analizler yapılmıştır. Fakat diğer çizelgeleme problemleri için de küçük değişiklikler ile aynı analizler yapılabilmesine elverişli bir benzetim ortamı mevcuttur. Bu tez çalışmasında, diğer çalışmalardan farklı olarak ilk defa GEP kullanılmıştır. Ayrıca klasik üretim parametrelerinin (işlem zamanı, teslim zamanı vb.) yanı sıra bazı özel üretim parametreleri (geriye kalan operasyon sayısı, geriye kalan operasyonun

kritik oranı vb.) kural türetilmesinde kullanılmıştır. Bir diğer farklılık ise literatürde bulunan ve iyi sonuçlar veren kuralları birer parametre olarak düşünüp bu kuralların doğrusal ve doğrusal olmayan birleşiminin GEP kullanılarak eğitilmesi ile elde edilen ve tüm amaçlar için literatürdeki kurallardan daha iyi sonuçlar veren kurallar türetilmiştir.

Bu tez çalışmasının geri kalan kısmı şu şekilde organize edilmiştir. Bu bölümün devamında literatürde ATÇ ve EATÇ ile ilgili yapılan çalışmalar hakkında bilgi verilmiştir. İkinci bölümde çizelgeleme ve sevk etme kuralları ile ilgili genel bilgiler verilmiştir. Üçüncü bölümde bu tez kapsamında tasarlanan sistem, türetilen kurallar ve sonuçlar anlatılmıştır. Son bölümde ise sonuçlar ile ilgili analizler ve gelecekteki çalışmalar hakkında bilgiler verilmiştir.

Geçmiş çalışmalar özetlenirken, literatürdeki makaleler iki ana başlık altında toplanmıştır. Birinci kısımda genel çizelgeleme problemleri ve EATÇ problemleri hakkında yapılan çalışmalar anlatılmıştır. İkinci kısımda bu çalışmanın da içerisine dâhil olduğu sevk etme kuralları ile çizelgeleme problemlerine çözüm arayan çalışmalara değinilmiştir.

Çizelgeleme alanında çalışmalar yapan ilk kişilerden olan Johnson (1954), iki makine üzerinde toplam tamamlanma zamanını minimum yapan bir algoritma geliştirmiştir [8].

Hanssmann ve W.Hess (1960) LP'yi (Doğrusal Programlama) kullanarak karı maksimize veya maliyeti minimize etmeyi amaçlayan problemleri çözdürmüşlerdir [9].

Held ve Karp (1962), Dinamik Programlamayı (DP) kullanarak çizelgeleme, montaj hattı dengeleme ve gezgin satıcı problemlerine cevap aramışlardır. Bu yaklaşımda büyük problem daha küçük alt problemlere bölünerek çözdürülmüştür [10].

Gupta (1982) makinelerin kurulum maliyetini en aza indirmek amacıyla n iş ve m makine içeren statik çizelgeleme problemlerini çözmek için dal sınır algoritma tabanlı bir matematiksel model geliştirmiştir. Bu çalışmada kurulum süreleri bir önceki işe bağlıdır. Tasarlanan problemde işler farklı rotalara sahip olabilir ve bir iş bir makineye birden fazla kez uğrayabilir. Bu yöntem sadece küçük boyutlu problemlerde kullanıma elverişlidir. Fakat öncelik kurallarının türetilmesinde kullanılabileceğinden büyük problemlerin çözümüne dolaylı olarak katkıda bulunmaktadır [11].

Brucker ve Schile (1990) EATÇ probleminin tanımını yapmışlardır ve grafik temelli bir algoritma geliştirmişlerdir. Bu algoritmayı toplam tamamlanma zamanını



minimize amaçlı iki işlik problemde denemiş ve iyi çözümler elde etmişlerdir. Fakat bu algoritma üç ve üzeri işlerden oluşan problemlerin çözümüne imkân vermemektedir [12].

Lin ve diğerleri (1997) Dinamik ATÇ probleminin çözümü için GA kullanmışlardır. Bu çalışmada hem kararlı hem de rassal dinamik ortamlar incelenmiştir. Çalışma çok amaçlı olarak ele alınmıştır. Yaptıkları çalışma sonucunda önerdikleri GA literatürdeki öncelik kurallarından daha iyi sonuçlar verdiği sonucuna ulaşmışlardır. Ayrıca önerdikleri GA'nın farklı teslim parametrelili, farklı makine iş yükleri ve farklı makine iş yükü dengesizlik parametrelerinin olduğu problemlerde de iyi sonuçlar verdiğini söylemişlerdir [13].

Demirkol ve diğerleri (1997) Darboğaz Değiştirme (SB) sezgisel yönteminin atölye tipi çizelgeleme problemlerinde toplam tamamlanma zamanı veya toplam gecikmeyi minimize amaçları için performansını analiz etmişlerdir. Çalışmalarında klasik test problemlerinin yanı sıra üç farklı rota yapısına sahip rasgele türetilmiş birçok problem üzerinde de deney yapmışlardır. Bu çalışmalar sonucunda SB yönteminin klasik sevk etme kurallarından daha iyi sonuçlar verdiğini söylemişlerdir. Fakat Yeniden Optimizasyon adımlarının kaldırılması ya da alt problemlerin çözümü için yaklaşık yöntemlerin kullanılması durumunda sevk etme kurallarının SB yöntemine baskın geldiğini söylemişlerdir. Ayrıca işlerin rotalarının yapısının çözüm performansı ve zamanı üzerinde etkili olduğunu ve artan problem boyutlarında (operasyon sayısı>5000) pratik olmadığını vurgulamışlardır [14].

Pezzella ve Merelli (2000) ATÇ probleminin çözümü için etkili bir yerel arama algoritması öne sürmüşlerdir. Önerilen algoritmada Darboğaz Değiştirme (SB) sezgiseli ile başlangıç çözüm oluşturduktan sonra Tabu Arama algoritması ile bu başlangıç çözüm geliştirilmiştir. Çözümleri hassaslaştırmak için kullanılan yöntem Kayan Tikanıklık Prosedürüdür. Bu yöntem ile deneme yapılan bazı problemlerde makul zamanlarda iyi sonuçlar elde etmişlerdir [15].

Zhang ve diğerleri (2008) EATÇ problemi çözümü için Değişken Komşuluk Arama ve GA birleşiminden oluşan bir yöntem kullanmışlardır. GA 'nın her bir bireyi topluma katılmadan önce VNS algoritması ile yerel aramaya tabii tutulup daha kaliteli hale getirilmiştir. VNS 'de iki farklı komşuluk tarif etmişlerdir. Sonuç olarak kullandıkları yöntem ile etkin ve etkili sonuçlar elde etmişlerdir [16].

Li ve Gao (2016) GA ve Tabu Arama algoritmalarını birleştirerek, Cmax minimize amaçlı EATÇ problemine uygulamışlardır. Literatürdeki en çok bilinen test örnekleri üzerinde bu melez algoritmayı denemişlerdir. Sonuç olarak önerdikleri bu melez algoritma hem hesaplama zamanı hem de amaç fonksiyonu açısından iyi sonuçlar vermiştir. Bazı test problemleri için yeni en iyi çözüm değerini bulmuşlardır [17].

Yazdani ve diğerleri (2017) ATÇ ortamında maksimum erken bitirme ve gecikmeleri minimize etmek amacıyla karma tam sayılı bir matematiksel model geliştirmişlerdir. Bu model küçük boyuttaki problemler için optimal sonuca ulaşsa da, orta ve büyük problemlere çözüm bulamamıştır. Bu yüzden etkili komşuluk arama yöntemi ile birleştirilmiş melez rekabetçi sezgisel bir algoritma geliştirmişlerdir. Algoritma test problemlerinde iyi sonuçlar vermiştir [18].

Çizelgeleme problemlerinde birçok amaç minimize edilmeye çalışılmaktadır, günümüzde artan enerji ve çevre kirliliği maliyetlerinden dolayı düşük karbonlu çizelgeleme problemleri ortaya çıkmıştır. Yin ve diğerleri (2017) enerji tüketimi ve çevre kirliliğini azaltmayı amaçlayan EATÇ problemi için karma-tam sayılı bir matematiksel model geliştirmişlerdir. Bu çalışmada esnek makine hızları, kullanılan enerji ve çevreye yayılan gürültüyü etkilemektedir. Farklı ölçeklerde üç problem örneği üzerinde bu yöntemin performansını analiz etmişlerdir. Sonuçlar, düşük karbonlu EATÇ problemi için önerilen modelin ve yöntemin etkin olduğunu göstermiştir [19].

Mansouri ve diğerleri (2016) iki makineli sıra bağımlı permütasyon akış türü üretim için çelişen toplam tamamlanma zamanı ve toplam enerji tüketimi amaçları minimize edilmeye çalışılmaktadır. Makinelerin hızları arttıkça tüketilen enerji miktarı da artmaktadır. Bu problem için matematiksel model kurulmuştur. Problem NP- ZOR olduğu için çözümü için sezgisel algoritma geliştirilmiştir. Bu sezgiselde Jhonson algoritması ile başlangıç çözümü oluşturup yerel arama ile iyileştirme yapmışlardır [20].

Wang ve diğerleri (2017) yapmış oldukları çalışmada, farklı makine(paralel) hızları kullanımı ile değişen enerji tüketim miktarını ve toplam tamamlanma zamanını minimize etmeye çalışmışlardır. Ayrıca enerji tüketimi için bir sınır değeri mevcuttur. 3 makine, 20 iş ve 25 kw (aynı anda 3 makinenin tükettiği toplam enerji) enerji kısıtı olan problemin çizelgelemesini yapmışlardır. Çözüm için matematiksel model geliştirilmiştir. Bu modelle iyi sonuçlar elde edilemeyince iki aşamalı sezgisel algoritma geliştirip probleme çözüm aramışlardır [21].

Çizelgeleme problemlerinde sevk etme kuralı çıkarımı, uygulaması ve karşılaştırması ile ilgili literatürde birçok çalışma yer almaktadır. Blackstone ve diğerleri (1982) üretimdeki sevk etme kuralları uygulamalarına ilişkin geniş bir literatür taraması yapmışlardır [22].

Rochette ve Sadowski (1976), deneysel tasarımlara dayanan ATÇ problemlerinde basit sevk etme kurallarının performansını karşılaştırmışlardır [23].

Panwalkar ve diğerleri (1977) yılında literatürde daha önceden önerilen yüzden fazla sevk etme kuralı hakkında kısa bilgiler vermişlerdir. Bu çalışmada kuralları basit, ağırlıklı ve sezgisel olmak üzere üç ana başlıkta toplamışlardır [24].

Anderson ve Nyirenda (1990), ATÇ probleminde gecikmeyi en aza indirmek için uygulanması kolay olan ve uygulanmaları için herhangi bir parametre tahmini gerektirmeyen iki sevk etme kuralı önermişlerdir [25].

Pierreval ve Mebarki (1997) sevk etme kurallarının dinamik seçimine dayanan bir zamanlama yöntemi öne sürmüşlerdir. Bu yöntemde her bir işyeri için geçerli tek bir sevk etme kuralı uygulamak yerine, dinamik bir seçim yöntemi ile birincil bir hedefi gerçekleştirmek ve ikincil bir üretim hedefinde iyi sonuçlar elde etmek amacıyla sistemin durumunda meydana gelen değişiklikler için en uygun kurallar seçilmektedir. Bu seçim, sistemin çalışma koşullarına ve sistem durumunun analiz sonuçlarına dayanmaktadır [26].

Chiang ve Fu (2007), ATÇP'leri için teslim zamanı temelli bir sevk etme kuralı tanımlamışlar ve bu kuralla literatürdeki 18 diğer kuralı, gecikme yüzdesi, ortalama gecikme ve maksimum gecikme amaçları için karşılaştırmışlardır. Bu karşılaştırma sonucunda gecikme yüzdesinde iyi sonuçlar elde etmelerine karşın maksimum gecikme ve ortalama gecikmede önerdikleri kural başarılı olmamıştır [27].

Sels ve diğerleri (2012) literatürdeki 30 basit sevk etme kuralını beş farklı amaç için ATÇ problemleri üzerinde test etmişlerdir. Bu beş amacı akış ve gecikme olarak iki ana başlıkta toplamışlardır. Buradan elde ettikleri sonuçlarla farklı amaçlar için iyi sonuç veren kuralları yeni kural türetmek için kullanmışlardır. Bu kuralların birleşiminden oluşan hibrid kurallar önceki kurallardan daha iyi sonuçlar vermiştir. Daha sonra yaptıkları çalışmayı EATÇ ve Dinamik problemler üzerinde uygulamışlardır. Bu üç ortamdan elde ettikleri sonuçlara göre hibrid kuralların basit kurallardan daha iyi sonuçlar verdiğini, kuralların ortamın esnek oluşuyla değişmediğini yani sağlam

olduğunu ve statik ortamdan dinamik ortama geçerken bu sağlamlığın bozulduğunu gözlemlemişlerdir [28].

Doh ve diğerleri (2013) EATÇ problemlerini iki aşamalı olarak ele almışlardır. Birinci aşama operasyonların hangi makinelerde yapılacağına karar verme, ikinci aşama ise makinelerin kuyruklarındaki operasyonların sıralanmasıdır. Çalışmada amaç fonksiyonu olarak toplam akış süresi, geciken işlerin yüzdesi, gecikmiş işlerin sayısı ve maksimum gecikmeyi ele almışlardır. Makineye operasyon seçimi için 3 farklı kuralın ve işlerin sıralanması için 12 farklı kuralın analizini yapmışlardır [29].

Jing ve Süer (2015) Çok amaçlı üretim çizelgeleme problemlerinin çözümü için sevk etme kuralına dayalı GA ile bulanık memnuniyet seviyeleri temelli bir yaklaşım öne sürmüşlerdir. Çalışmada  $C_{max}$ , ortalama akış, maksimum gecikme ve toplam gecikmeyi minimize amaçlı ATÇ problemine odaklanılmıştır. Bir çok test probleminde yapılan denemeler sonucunda optimal ve optmale yakın sonuçlar elde etmişlerdir [30].

Lu ve diğerleri (2011) sevk etme kuralları ile sipariş serbest bırakmayı (ORR) birleştirip ATÇ problemlerine benzetim modeli ile çözüm aramışlardır. Teslim tarihi ve akış temelli amaçları göz önünde bulundurmışlardır [31].

Rajendran ve Holthaus, dinamik ATÇ için sevk etme kurallarını araştırmış ve kendileri tarafından önerilen üç sevk etme kuralını diğer 13 sevk etme kuralı ile karşılaştırmışlardır [6].

Jayamohan ve Rajendran (2000) akış süresi ve gecikmesinin ortalama, maksimum ve varyansı gibi çeşitli amaç fonksiyonları için sevk etme kuralları tasarlamışlar ve bu kuralları, dinamik çizelgeleme problemleri için klasik kurallarla karşılaştırmışlardır [32].

Şekil 1.1'de bu tez çalışmasının temelini oluşturan altı makale; kullandıkları deney tasarımı, amaç fonksiyonu ve sevk etme kuralları açısından karşılaştırılmıştır [7] [6] [5] [32] [33] [34].

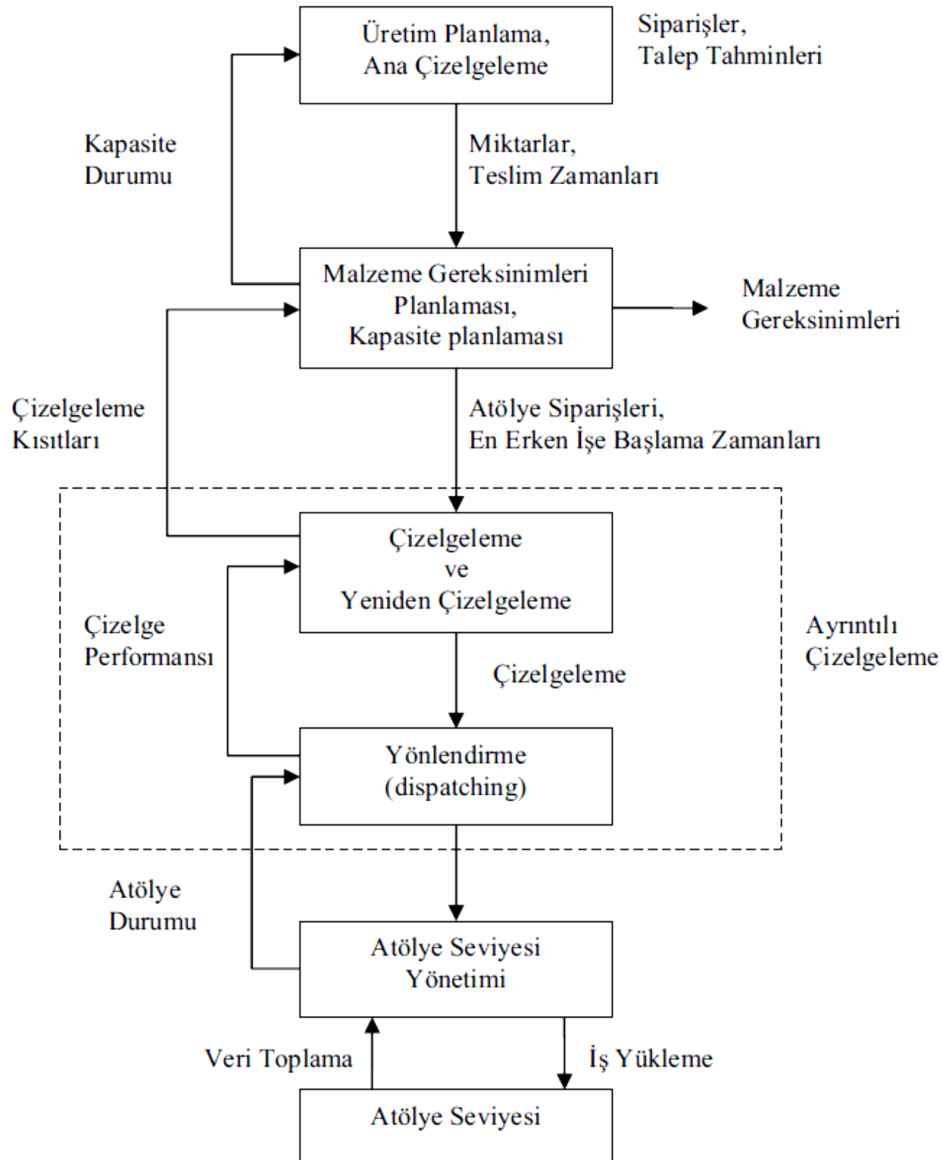
**Tablo 1.1.** Makale karşılaştırması

	Önerilen Kurallar	Karşılaştırılan Kurallar	Statik/Dinamik	Amaç	Problem Türü	Teslim Tarihi Parametresi	Makine Yüzdesi
Holthaus ve Rajendran (1997)	PT+WINQ, PT+WINQ+AT PT+WINQ+SL, MCOVERT	SPT, FIFO, COVERT, WINQ, RAN-SPT/WINQ	Dinamik	Fort, Fmax, Fvar, T%, Tmax, Tvar	ATÇ	3, 4, 5, 6	%85 %95
Rajendran and Holthaus(1999)	(PT+WINQ)/TIS, PT/TIS, AT-RPT	SPT, FIFO, COVERT, EDD, RR, AT, S/OPN, PT+WINQ, PT+WINQ+AT, PT+WINQ+SL	Dinamik	Fort, Fmax, Fvar, T%, Tmax, Tvar	ATÇ,FS,MFS	4, 6, 8	%80 %85 %90 %95
Chen ve Matis (2013)	WBMR	FIFO, SRPT, WSPT, RR, PT+WINQ, WATC	Dinamik	WTort	ATÇ	2, 5	%85 %95
Tay ve Ho (2008)	$RD + 2PT + 2aTPT + nOps,$ $RD + PT + 2aTPT,$ $7aTPT + 11PT + 12(nOps+RD),$ $(RD + DD) + 2(RD + aTPT) +$ $PT - nOps,$ $(RD + DD) + (aTPT + PT) -$ $2(RD/nOps)$	SPT FIFO EDD MDD SL	Statik	Fort Tort Cmax T%	EATÇ	1.2, 1.5, 2	100%
Xiong, Hegen, et al.(2017)	SOP, MSOP, RR+SOP, RR+MSOP	SPT, EDD, MDD, ODD, WINQ SL/OPN, RR, PT+WINQ+SL, PT+PW, PT+PW+ODD, LBF+LBT PT+PW+FDD, PT+RR+PW, RR+PT+PW+ODD, RR- (LBF+LBT)	Dinamik	T T%	ATÇ	10, 14, 18, 22	90%
Jayamohan ve Rajendran (2000)	FDD, PT+PW, PT+PW+ODD, (OPSLK/PT;FDD), (OPSLK/PT;ODD), AVPRO	SPT, MOD, EDD, RR, PT+WINQ, ATC, AT-RPT, ODD, PT+WINQ+SL	Dinamik	Fort, Fmax, Fvar, T%, Tmax, Tvar	ATÇ, MFS	3, 5, 7	%85 %95

## 2. ÇİZELGELEME VE SEVKETME KURALLARI

### 2.1. Çizelgeleme

Çizelgeleme, hizmet ve imalat sektöründe belirli zaman aralıklarında düzenli olarak kullanılan karar verme sürecidir [2]. Üretim sistemi ya da hizmet organizasyonundaki çizelgeleme işlevi, diğer pek çok işlevle etkileşim içerisindedir. Bu etkileşimler sisteme bağlıdır ve bir durumdan diğerine büyük ölçüde farklılık gösterebilir [1]. Şekil 2.1 'de üretim sektöründe çizelgelemenin yeri gösterilmiştir.



Şekil 2.1. Üretim Sistemindeki Bilgi Akışı (Pinedo 2008)

### **2.1.1. Çizelgelemede kullanılan temel kavramlar**

#### ***İşlem zamanı ( $p_{ij}$ )***

Bu gösterim i. makinede j. işin işlem zamanını temsil etmektedir. Bu zaman makinelere göre değişiklik gösterebilmektedir [2].

#### ***Serbest kalma zamanı ( $r_j$ )***

Bu gösterim j. işin sistemde işlenmeye hazır hale geldiği zamanı ifade eder [2].

#### ***Teslim tarihi ( $d_j$ )***

Bu gösterim j. işin müşteriye teslim edileceği zamanı ifade eder [2].

#### ***Ağırlık ( $w_j$ )***

J. işin sistemdeki diğer işlere göre önemini gösteren öncelik faktörüdür [2].

#### ***İşleme başlama zamanı ( $S_{ij}$ )***

Bu gösterim i. makinede j. işin işleme başlama zamanını temsil etmektedir [2].

#### ***Tamamlanma zamanı ( $C_{ij}$ )***

Bu gösterim i. makinede j. işin tamamlanma zamanını temsil etmektedir [2].

### **2.1.2. Çizelgelemenin sınıflandırılması**

Üretimde çizelgeleme problemleri çeşitli bakış açılarına göre sınıflandırılabilir. Bu başlığın altında kısaca çizelgeleme problemleri hakkında bilgi verilecektir.

#### **2.1.2.1. Makine Ortamı**

Makine sayısı bakımından çizelgeleme problemleri bir ve birden fazla makine diye ikiye ayrılır.

##### **2.1.2.1.1. Tezgâh sayısı göre çizelgeleme**

Makine sayısı bakımından çizelgeleme problemleri bir ve birden fazla makine diye ikiye ayrılır.

#### ***Tek makineli***

Tüm makine ortamlarının en basit halidir, sadece tek makine bulunan üretim ortamıdır. Tek makine modelleri için elde edilebilen sonuçlar, yalnızca tek makine ortamına ilişkin fikir sunmakla kalmayıp aynı zamanda daha karmaşık makine ortamları için geçerli buluşsal yöntemler için bir temel oluşturmaktadır [1].

#### ***Çok makineli***

Birden fazla makinenin seri veya paralel olarak yerleştirildiği üretim ortamlardır. Seri bağlı makineler farklı işleri yaparken, paralel makineler aynı işlemleri yapmaktadır. Seri makinelerin olduğu sistemlerde bir makinede sıkışıklık olduğunda aynı makineden

bir tane daha alınıp sistemin rahatlatılması günlük hayatta oldukça rastladığımız bir durumdur. Bu yüzden paralel makineli ortamlarda üretimde sıkça kullanılmaktadır.

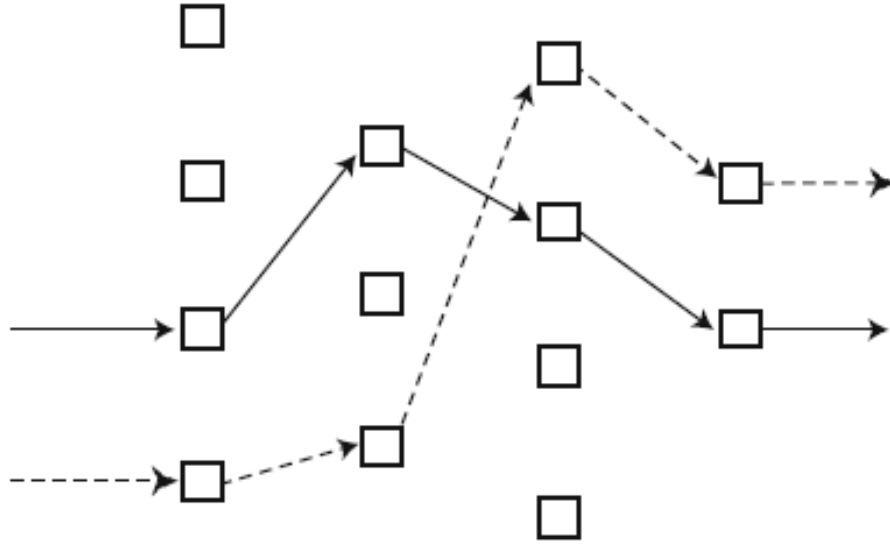
#### 2.1.2.1.2. Akış şekline göre çizelgeleme

##### *Akış tipi*

M adet seri makinenin bulunduğu üretim ortamında her işin aynı rotayı kullanarak tüm makinelerde işlem gördüğü üretim sistemidir. Bir makinedeki operasyon biten iş sıradaki makinenin kuyruğuna girmektedir [1].

##### *Esnek akış tipi*

Akış tipi üretim sisteminin genel halidir. Özdeş paralel makinelerin bulunduğu c adet istasyondan oluşan üretim ortamıdır. İşlerin operasyonların her istasyonda yalnızca bir makine tarafından işleme tabii tutulur. İşler 1. istasyondan başlayarak sırayla tüm istasyonlarda işlem görürler [1].

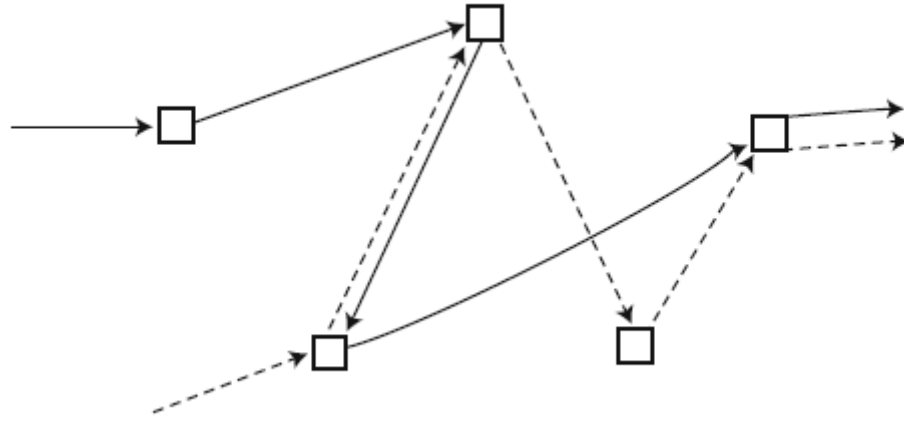


Şekil 2.2. Esnek akış tipi (Pinedo, 2005)

##### *Atölye tipi*

N iş ve m makineden oluşan sistemde her bir iş belirli bir sırada birçok makine tarafından işlenebilir ve devridaim yapılmamaktadır [2].





**Şekil 2.3.** *Atölye tipi (Pinedo, 2005)*

### ***Esnek atölye tipi***

Atölye tipi ve paralel makineli ortamların genelleştirilmiş halidir. İşler farklı rotalar ile, paralel makinelerin bulunduğu c adet istasyonda, her istasyonda yalnızca bir makine tarafından işlem görmektedir [1].

### ***Açık atölye tipi***

İşlerin hepsi tüm makinelerde işlem gördüğü üretim sistemidir. Bazı makinelerde işlem süresi sıfırdır ve herhangi bir rota kısıtı bulunmamaktadır [1].

### ***2.1.2.2. Parametrelerin durumuna göre çizelgeme***

Çizelgeme problemleri sistem durumuna göre deterministik ve stokastik olmak üzere ikiye ayrılmaktadır.

#### ***Deterministik modeller***

Bu modellerde olaylar ve zamanlar önceden bilinmektedir. Herhangi bir rassallık söz konusu değildir.

#### ***Stokastik modeller***

Bu modellerde olaylar ve zamanlar önceden bilinmemektedir. Bu modellerde makine bozulmaları, sonradan gelen siparişler, belirsiz işlem süreleri vb. etkenler dikkate alınır.

### ***2.1.2.3. Sistemin durumuna göre çizelgeme***

Çizelgeme problemleri sistem durumuna göre statik ve dinamik olmak üzere ikiye ayrılmaktadır.

### ***Statik***

Bütün bilgiler önceden bilindiği ve zaman içerisinde değişmediği durumlardır. Bu tip problemlerden elde edilen sonuçlar daha karmaşık olan dinamik ortamlardaki problemlerin çözümü için oldukça önemli bilgiler sunar.

### ***Dinamik***

Bu sistemlerde işler sürekli gelmektedir. Üretim esnasında gelen işler dikkate alınarak çizelgeleme yapılır. Bu yüzden statik çizelgelemeye oranla daha zor bir problemidir.

#### ***2.1.2.4. Performans ölçütüne göre çizelgeleme***

Amaç fonksiyonlarına göre çizelgeleme problemleri, tek amaçlı ve çok amaçlı olarak iki ana başlık altında toplanabilir. Çizelgeleme problemlerinde en çok kullanılan performans ölçütleri aşağıda açıklanmıştır.

#### ***Tamamlanma Süresi ( $C_j$ )***

Sistemdeki son işin sistemi terk ettiği zamandır. Bu sürenin minimum olması makinelerden maksimum yararlanıldığı anlamına gelir [1].

#### ***Akış süresi ( $F_j$ )***

İşlerin sisteme girdikleri zaman ile çıktıkları zaman arasındaki farktır.

#### ***Sapma süresi ( $L_j$ )***

Sistemdeki geç kalan işler arasında maksimum olanını ifade eden amaç fonksiyonudur. Diğer bir ifadeyle işlerin teslim zamanı ile tamamlanma zamanı arasındaki farktır.

$$L_j = C_j - d_j \quad (2.1)$$

#### ***Gecikme süresi ( $T_j$ )***

İşlerin teslim tarihi ile tamamlanma zamanı arasındaki pozitif sapma zamanıdır.

$$T_j = \text{Enb}\{0, L_j\} \quad (2.2)$$

#### ***Geciken iş sayısı ( $N$ )***

Zamanında teslim edilemeyen işlerin sayısıdır. Aşağıda kullanılan literatürde kullanılmış performans ölçütlerin bazıları sunulmuştur [35].

- Siparişin tamamlanma zamanının en küçüklenmesi
- Ortalama tamamlanma zamanının minimize edilmesi
- Ortalama ağırlıklı tamamlanma zamanının minimize edilmesi
- Ortalama bekleme süresinin minimize edilmesi

- Ortalama gecikme veya teslim gecikmesi suresinin minimize edilmesi
- Ortalama ağırlıklı gecikme veya teslim gecikmesi suresinin minimize edilmesi
- En büyük gecikme veya teslim gecikmesi suresinin minimize edilmesi
- Ara stok düzeyinin minimize edilmesi
- Erken teslim maliyetinin minimize edilmesi
- Geciken iş sayısının minimize edilmesi
- Ağırlıklı geciken iş sayısı toplamının minimize edilmesi
- Tezgâh ve işgücü kullanım oranlarının minimize edilmesi
- Kritik makine yükünün minimize edilmesi
- Toplam makine yükünün minimize edilmesi
- Siparişin akış suresinin minimize edilmesi
- Ortalama akış suresinin minimize edilmesi

#### **2.1.2.5. Çizelgeleme problemlerinin gösterim şekli**

Literatürde çizelgeleme problemlerinin gösterimi için kullanılan en yaygın yöntem 3 gösterge ile temsil edilir [1].

- $\alpha$  makinelerin durumunu gösteren simgedir.
- $\beta$  işlerin detayı ve kısıtları gösteren simgedir.
- $\gamma$  amaç fonksiyonunu gösteren simgedir.

## **2.2. Sevk Etme Kuralları**

Çizelgeleme problemleri NP-Zor olduğundan kesin çözüm veren yöntemlerin kullanması artan problem boyutunda imkânsız hale gelmiştir. Bu yüzden çizelgeleme problemlerinde benzetim temelli yaklaşımların kullanılması son zamanlarda oldukça popülerleşmiştir. Benzetim ortamında, makineler belirli kurallar dâhilinde önündeki işlerden birini seçmektedir. Bu kurallar öncelik kuralları ya da sevk etme kuralları olarak literatürde yer almaktadır. Bu çalışmada sevk etme kuralları olarak kullanılacaktır. Sevk etme kuralları, makinelerin mevcut işleri bittiğinde kuyruklarındaki işlerden birini seçmek için kullanılır [6].

### **2.2.1. Sevk etme kurallarının sınıflandırılması**

Sevk etme kuralları farklı şekillerde sınıflandırılabilir. Blackstone ve diğerleri 1982 yılında sevk etme kurallarını aşağıdaki gibi sınıflandırmıştır [22]:

- İşlem süresi temelli kurallar,

- Teslim tarihi temelli kurallar,
- Hem işlem süresi hem de teslim tarihi temelli kurallar,
- Ne işlem süresi nede teslim tarihi temelli olmayan kurallar.

En kısa işlem süresi (SPT) kuralı işlem zamanı temelli kurallar örnektir. Bu kural ortalama akışın minimize edilmek istendiği problemlerde oldukça iyi sonuçlar verir. Başka bir sınıflandırma yönteminde ise Panwalkar ve Iskander [24]:

- Basit öncelikli kurallar (SPR)
- Basit kuralların birleşimi (CDR)
- Ağırlıklı öncelik indisleri
- Sezgisel kurallar

şeklinde tanımlamışlardır. Sevk etme kuralları tanımları yapılırken kullanılan notasyon tablo 2.1 'de özetlenmiştir.

**Tablo 2.1.** *Sevk etme kuralları notasyonu*

Sembol	Tanım
t	Sevk etme kuralları ile karar verilecek zaman
$p_{ij}$	i. işin j. operasyonun işlem süresi
$d_i$	i. işin teslim tarihi
c	Teslim tarihi sıklık parametresi
$Z_i$	i. işin öncelik indeksi
$r_i$	i. işin serbest kalma zamanı
$s_i$	i. işin boş zamanı
$W_i$	i. işin geriye kalan işlem zamanı

Literatürdeki yaygın olarak kullanılan bazı sevk etme kuralları aşağıda açıklanmıştır.

1. FIFO, ilk giren ilk çıkar (2.3)

$$Z_i = r_i$$

2. SPT, en kısa işlem süresi

$$Z_i = \sum p_{ij} \quad (2.4)$$

3. EDD, en erken teslim tarihi (işler için)

$$Z_i = d_i \quad (2.5)$$

4. ODD, en erken operasyon teslim tarihi

$$Z_i = r_i + \frac{(d_i - r_i)}{P_i} \quad (2.6)$$

5. MDD, en küçük düzeltilmiş teslim tarihi (işler için)

$$Z_i = Enb \left\{ d_i, t + \sum_j p_{ij} \right\} \quad (2.7)$$

6. MODD, düzeltilmiş operasyon teslim zamanı

$$Z_i = Enb \left\{ r_i + \frac{(d_i - r_i)}{P_i}, t + p_{ij} \right\} \quad (2.8)$$

7. CR, en küçük kritik oran

$$Z_i = (d_i) \quad (2.9)$$

8. COVERT

$$c_i = \begin{cases} \frac{Wr_i - s_i}{Wr_i} & \text{eğer } 0 \leq s_i < Wr_i \\ 0 & \text{eğer } s_i \geq Wr_i \\ 1 & \text{eğer } s_i < Wr_i \end{cases}$$

$$Z_i = c_i/p_{ij} \quad (2.10)$$

### 2.3. EATÇ Problemlerinde Sevk Etme Kuralları

EATÇ şu şekilde ifade edilebilir.  $J = \{J_1, J_2, \dots, J_n\}$  bağımsız işler kümesidir. Bir iş ( $J_i$ ), verilen sıraya göre birbiri ardına yapılmak üzere operasyon ( $O_{i1}, O_{i2}, \dots, O_{ini}$ ) dizisinden oluşur.  $U = \{M_1, M_2, \dots, M_m\}$  kümesidir ve her bir operasyon ( $O_{ij}$ ), uyumlu makinelerin bir alt seti ( $U_{ij} \subseteq U$ ) içindeki herhangi bir makinede işlenebilir. Her operasyon için  $U_{ij} = U$  olduğu durumda tam esneklik, en az bir operasyon için  $U$ 'nun uygun bir altkümesi  $U_{ij}$  varsa kısmi esneklik söz konusudur. Her operasyonun işlem süresi makineye bağlıdır.  $t_{ijk}$  ile makine  $M_k$ 'de çalıştırıldığında  $O_{ij}$  operasyonun işleme süresini göstermektedir. Her operasyonun başladıktan sonra kesintisiz devam ettiği ve makineler bir seferde tek bir operasyon gerçekleştirdiği varsayılmaktadır [36].

EATÇ problemleri, operasyonların uygun makinelerden birine atanması ve bu atama işlemlerinden sonra makinelerin kuyruklarındaki işlerin sıralanması olarak iki alt problemden oluşmaktadır. Bu alt problemlerden ilki çözüldüğünde problem artık ATÇ haline gelir. EATÇ problemlerinin çözüm yöntemlerini hiyerarşik ve bütünleşik olarak ikiye ayırabiliriz. Hiyerarşik yaklaşımda EATÇ problemin iki alt problemi birbirinde bağımsız olarak düşünülür. Önce problem operasyonların makinelere atanması ile ATÇ problemi haline getirilir. Daha sonra sıralama işlemi uygulanarak problem çözülür. Bütünleşik yaklaşımda ise atama işlemi ve sıralama işlemi birbiriyle ilişkili şekilde

yapılır. Sıralama işlemi aynı zamanda atama işlemine de etkiler. Bu çalışma kapsamında bütünlük yaklaşımını kullanarak EATÇ problemlerine çözüm aranmıştır.

EATÇ problemlerinin çözümü için bir çok yöntem kullanılmaktadır. Örneğin matematiksel model, sezgisel algoritmalar ve meta sezgisel algoritmalar ile bu probleme cevap aranmıştır. Bu çalışma kapsamında benzetim temelli sevk etme kuralları ile bu probleme çözüm aranmıştır. Sevk etme kuralları makinelerin kuyruklarındaki işlerin sıralamasında kullanılmıştır. Operasyonların makinelere atanmasında farklı bir algoritma kullanılmıştır. Fakat bu algoritma sıralama işlemine bağlıdır. İleriki çalışmaların daha iyi anlaşılabilmesi için bir örnek üzerinde literatürde kullanılan iki sevk etme kuralları analiz edilmiştir. Örnek %0 esnekliğe sahip yani ATÇ problemidir. Örnekte 3 makine, 5 iş ve her işin 3 'er operasyonu vardır. İşlerin sisteme geliş zamanları ve teslim tarihleri tablo 2.2 'deki gibidir.

**Tablo 2.2.** Örnek problemdeki işlerin verileri

İş	Serbest Kalma Zamanı (rj)	Teslim Zamanı (dj)
1	5,92	26,95
2	2,24	13,99
3	10,63	29,85
4	1,17	11,6
5	18,68	34,69

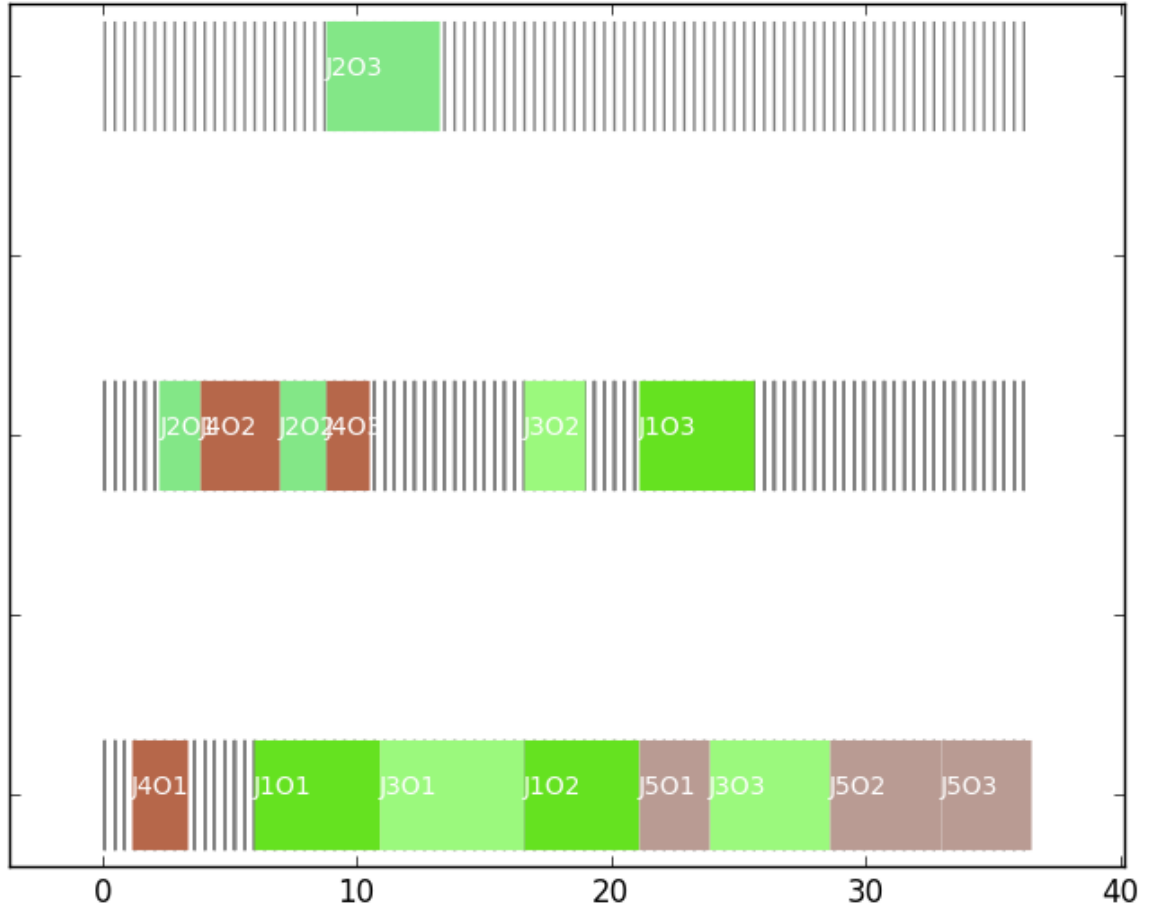
Tablo 2.3' de ise bu işlerin operasyonlarının zamanları verilmiştir.

**Tablo 2.3.** Örnek problem operasyon verileri

İş	Operasyon	Makine	Süre
1	O <sub>1</sub>	1	4,97
	O <sub>2</sub>	1	4,55
	O <sub>3</sub>	2	4,49
2	O <sub>1</sub>	2	1,58
	O <sub>2</sub>	2	1,84
	O <sub>3</sub>	3	4,42
3	O <sub>1</sub>	1	5,65
	O <sub>2</sub>	2	2,43
	O <sub>3</sub>	1	4,74
4	O <sub>1</sub>	1	2,15
	O <sub>2</sub>	2	3,14
	O <sub>3</sub>	2	1,66
5	O <sub>1</sub>	1	2,74
	O <sub>2</sub>	1	4,37
	O <sub>3</sub>	1	3,56

### *FIFO kuralına göre çizelgeleme*

Bu kural önceden de ifade edildiği üzere makinenin sırasında ilk gelen işin ilk olarak işlenmesi prensibine dayanır. Burada operasyonların makinelerin sırasında girebilmesi için önceki operasyonlarının tamamlanması gerekmektedir. Bir diğer deyişle her operasyonun serbest kalma anda makinenin kuyruğuna girdiği varsayılmaktadır. Verilen örnek FIFO kuralına göre çizelgelendiğinde Gantt grafiği şekil 2.4 'deki gibi oluyor. Bu şekil incelendiğinde işlerden en erken sisteme gelen 4. işin ilk operasyonu ilk olarak işlenmeye başlandığı görülmektedir. Yine grafikten hareketle işlerin önceki operasyonları bitmeden diğer operasyonlarının başlamadığı gözlemlenmiştir.

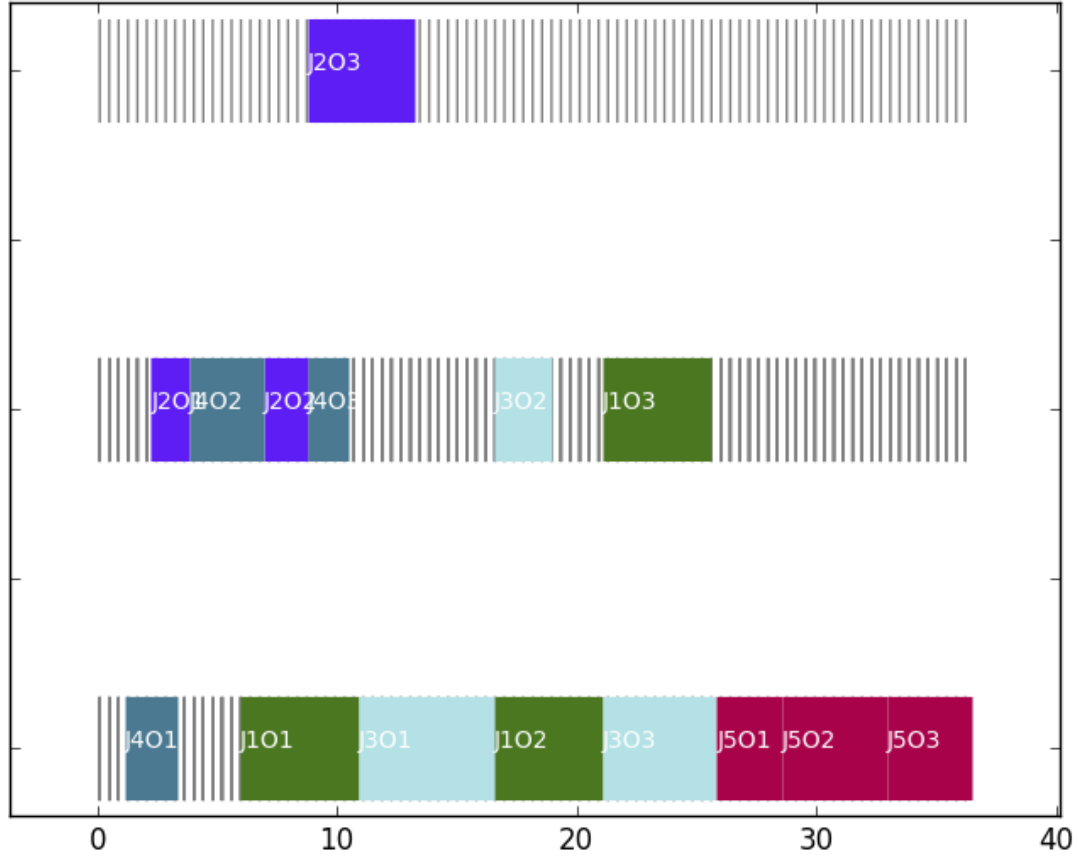


**Şekil 2.4.** FIFO için Gantt şeması

### *EDD kuralına göre çizelgeleme*

Bu kural önceden de ifade edildiği üzere makinenin sırasındaki operasyonlardan teslim zamanı (işin teslim zamanı) en erken olana öncelik verilir. FIFO kuralı için

geçerli olan kuyruğa girebilme durumları burada da geçerlidir. Verilen örnek EDD kuralına göre çizelgelendiğinde Gantt grafiği şekil 2.5 'deki gibi oluyor.



Şekil 2.5. EDD kuralı Gantt şeması

Tablo 2.4 'de FIFO ve EDD kuralları ile çizelgelenmiş bu problem için Cmax, Tort ve Fort değerleri gösterilmiştir. Bu iki şekli incelediğimizde makine 2 ve makine 3 'deki sıralamada herhangi bir farklılık söz konusu değildir. Fakat makine 1 'de 5. işin ilk operasyonu ile 3. işin üçüncü operasyonu yer değiştiği görülmektedir. Bu değişim toplam tamamlanma zamanında bir değişikliğe neden olmasa da ortalama kış ve ortalama gecikmede değişimlere sebep olmuştur. Problem boyutu arttıkça bu değişimler daha büyük farklılıklara yol açacaktır.

Tablo 2.4. FIFO ve EDD sonuçlar

Kural	Cmax	Tort	Fort
FIFO	36,51151987	1,27427	14,60333205
EDD	36,51151987	1,82307	13,10667255



### 3. KULLANILAN YÖNTEM

Bu bölümde EATÇ probleminin çözümü için kullanılan sevk etme kurallarının nasıl türetildiği ayrıntılı olarak anlatılmıştır. Kuralları türetirken benzetim ve Gen İfade Programlama ‘dan yararlanılmıştır. Yapılan çalışmanın daha iyi anlaşılabilmesi için ilk olarak sevk etme kurallarının türetilmesinde kullanılan GEP hakkında genel bilgiler verilmiştir. Daha sonra bu iki yöntem kullanılarak yapılan çalışma ayrıntılı olarak anlatılmıştır.

#### 3.1. Gen İfade Programlama (GEP)

Gen ifadesi programlama, 1999'da Candida Ferreira tarafından geliştirilmiştir. Bu programlamada Candida, GA ‘da kullanılan sabit uzunluktaki lineer yapıyı ve GP ‘da kullanılan farklı boyut ve şekillere sahip dallanmış yapıyı birleştirmiştir. Ayrıca, GP ‘deki farklı şekil ve büyüklükteki ağaç yapılarının tanımlamasını doğrusal olmayan yapıdan çıkararak sabit uzunlukta doğrusal bir yapı olarak tasarlamıştır. GEP ‘in en önemli özelliği kromozomların, basit, lineer kompakt, nispeten küçük ve kolayca işlenebilir olmasından kaynaklanmaktadır. GEP modelinin diğer bir artısı da açık formül üretimine olanak sağlamasıdır [37].

##### 3.1.1. Kullanılan Kavramlar

Bu kısımda GEP ‘de kullanılan kavramlar hakkında kısa bilgiler verilmiştir.

###### *Fonksiyon kümesi*

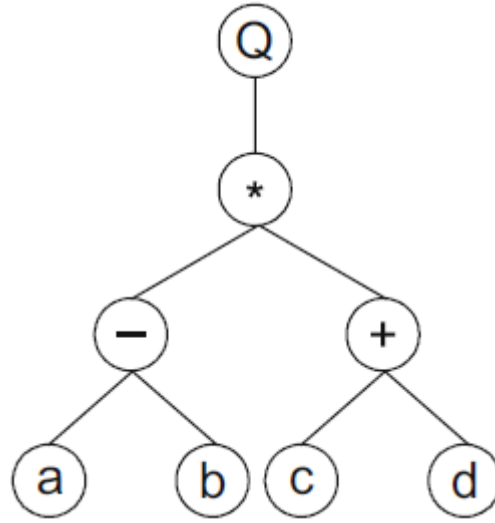
İçerinde matematiksel operatörlerin bulunduğu kümedir ve “**F**” ile gösterilir.  $F = \{Q, +, -, /, *\}$  gibi matematiksel operatörler bulunabilir.

###### *Terminal kümesi*

İçerinde sabit değerlerin bulunduğu kümedir ve “**T**” ile gösterilir.  $T = \{a, b, c\}$  gibi bağımsız değişkenler örnek olarak gösterilebilir.

###### *Açık okuma çerçevesi (ORF)*

İçerdiği kodlama yapısı sayesinde GEP, kolaylıkla kendini değiştirebilme özelliğine sahiptir. Bu kodlama Karva dili olarak adlandırılır ve K-dili olarak gösterilir. Bu dil ile oluşturulan ağaç yapısına ifade ağacı denir.  $\sqrt{(a - b) * (c + d)}$  İfadesinin kromozom gösterimi “**Q, x, -, +, a, b, c, d**” şeklindedir. İfade ağacı Şekil 3.1 ‘de gösterilmiştir.



Şekil 3.1. İfade ağacı

İfade ağacı, bir kuralı bilgisayarda temsil etmek için mükemmel bir yoldur, çünkü ağaç keyfi olarak karmaşık olabilir. İçerdiği yapı sayesinde ifade ağaçları hızlı değerlendirilebilir.

ET ‘yi K-dili ‘ne dönüştürmek için yukarıdan aşağıya doğru ve soldan sağa doğru tarama yapılır. Bu tarama yapılırken bir sembole gelindiğinde soldan başlayarak dal eklenir. Dal sayısı sembole göre değişiklik gösterir. Örneğin “Q” karekök işlemini temsil etmektedir ve bir tane dal gerektirirken “+” için en az iki dal gereklidir. Birbirine kolayca dönüştürülebildiği için yapılan değişiklikler kısa sürede analiz edilebilir [38].

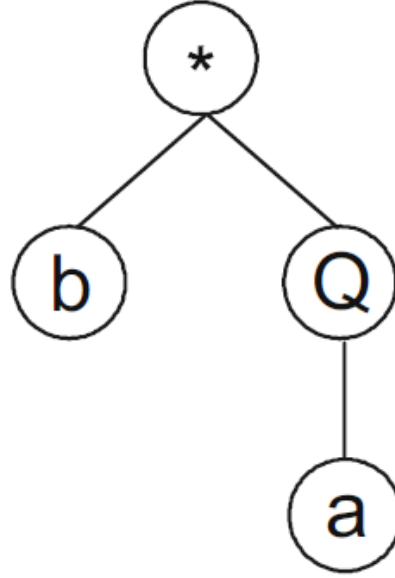
### Gen

Bir gen, K-dilinde kodlanmış sabit sayıdaki sembollerden meydana gelir. Genler baş ve kuyruk kısmından oluşur. Baş kısmında ifadenin kodlanması için kullanılan operatörler ve sabitler bulunurken kuyruk kısmında ise sadece sabitler bulunur. Böylece baş kısımdaki operatörlerin sonlanması için yeterli sayıda terminalin olması sağlanır. Baş kısmın uzunluğu bilinen bir GEP ‘te kuyruğun uzunluğu denklem (3.1) ‘deki gibi hesaplanır.

$$t = h * (EnbArg - 1) + 1 \quad (3.1)$$

Bu denklemde t kuyruğun uzunluğunu, h baş kısmın uzunluğunu ve EnbArg ise herhangi bir operatör tarafından kullanılmasına izin verilen maksimum bağımsız değişken sayısını temsil etmektedir. Örneğin  $F=\{Q,+,-,/,*\}$  ve  $T=\{a, b\}$  olan veri kümesine sahip problemde  $h=5$  ise  $t=5*(2-1)+1=6$  olur. Böylece genin uzunluğu

5+6=11 olur. Bu genin dizilimi “\*,b,Q,a,+, b, b, b, a, b, b” şeklinde olursa ağaç yapısı Şekil 3.2 ‘de ki gibi olur.

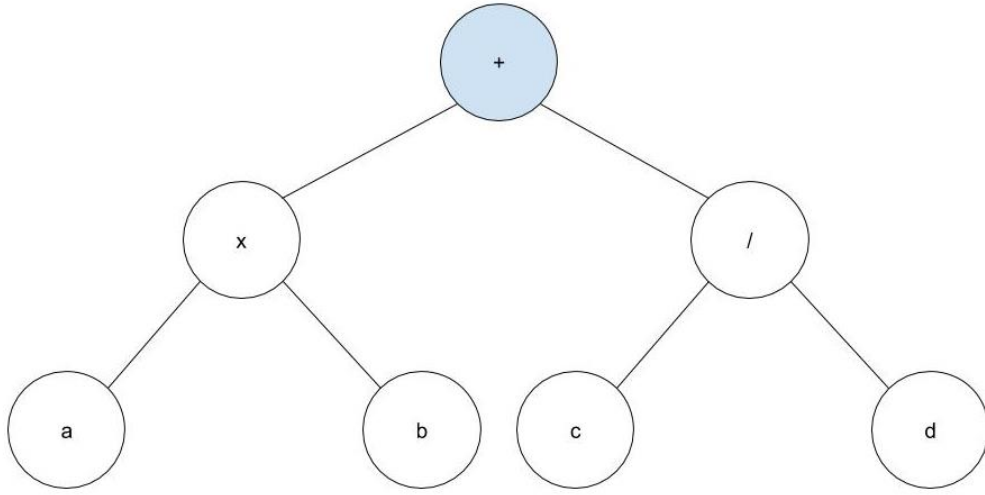


Şekil 3.2. Örnek ağaç yapısı

Ağaç yapısından görüldüğü üzere gendeki sembol sayısı sabit olmasına rağmen bu sembollerin hepsi kullanılmamaktadır. Bir genin fonksiyonel uzunluğu, tuttuğu sembol sayısından daha az olabilmektedir. Bu durum Açık Okuma Çerçevesinin (ORF) getirdiği bir kolaylıktır. Bu yapı ile yapılan değişiklikler kontrol edilmeksizin yeni arama alanlarına giriş yapılmasına olanak sağlar [37].

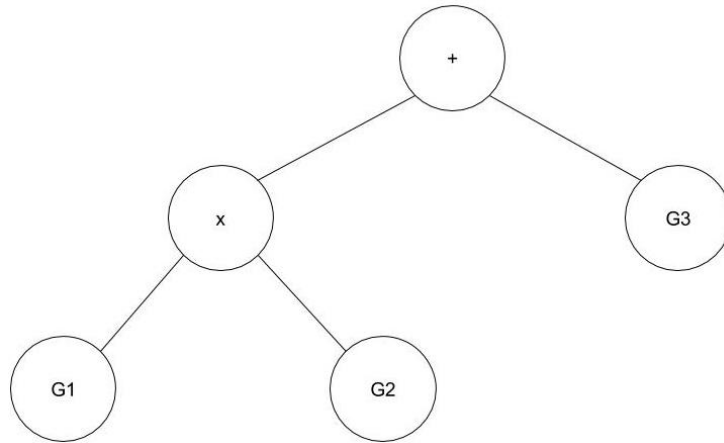
### **Genom (Kromozom)**

Bir veya daha fazla genden oluşan sabit boyuttaki doğrusal sembolik dizidir. Birden fazla genden oluşan kromozomlarda genler statik veya evrimleşebilen şekilde iki ayrı yöntemle birbirine yapışabilir. Örneğin Gene 1: “x,a,b” ve Gene 2: “/,c,d” şeklinde tanımlanmış olsun ve “+” operatörüyle bu iki gen statik olarak bağlandığında, Şekil 3.3 de ki gibi bir ifade ağacı oluşur.



**Şekil 3.3.** Statik bağlı kromozom ifade ağacı

Bağlantıyı statik değil de evrimleşebilen olarak tanımladığımızda homeotik genler ortaya çıkmaktadır. Homeotik genler normal genler gibi baş ve kuyruk kısmından oluşmaktadır. Örneğin, bir kromozomun üç normal geni G1, G2 ve G3 olsun ve kromozom dizilimi “+,x,G3,G1,G2” şeklinde homeotik bir yapıda ise, bu yapının ifade ağacı şekil 3.4 ‘te gösterildiği gibi olmaktadır [37].

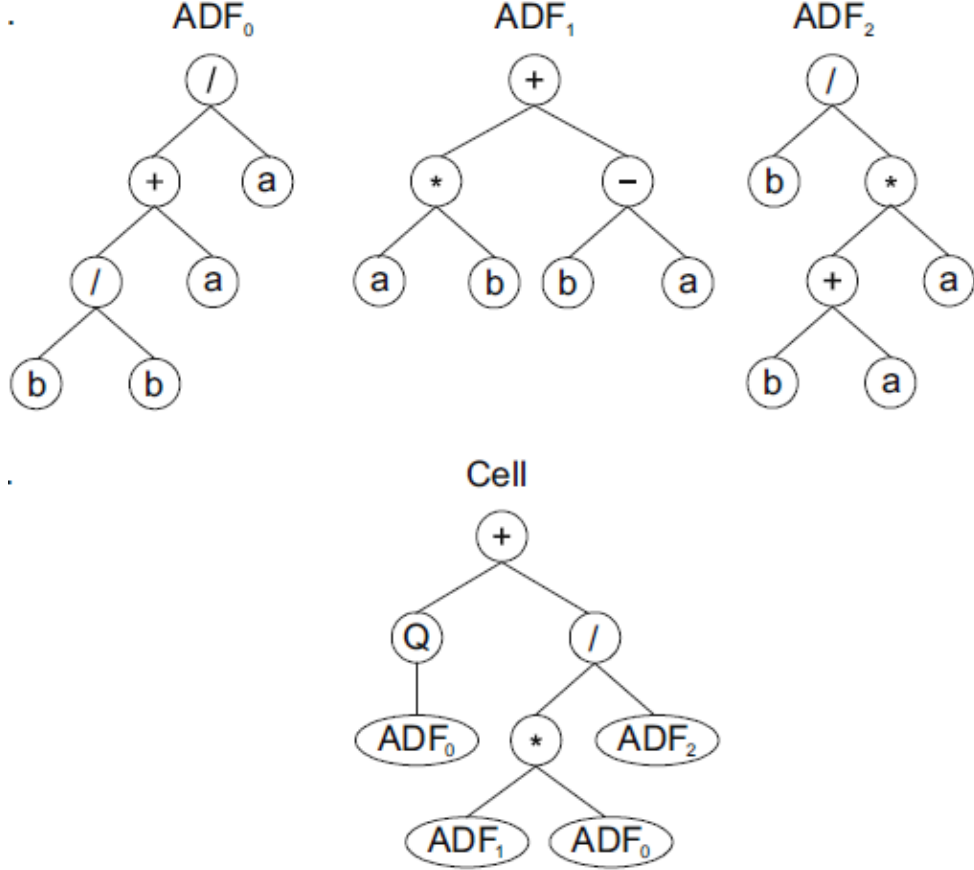


**Şekil 3.4.** Homeotik gen ifade ağacı

Bir diğer çoklu gen yapısı da otomatik tanımlanmış fonksiyon (ADF) kullanan kromozom yapısıdır. Bu yapıdaki her bir gen kendi içerisinde ayrı bir kromozomdur. ADF kullanılarak bu kromozomları içeren genler, yeni tasarlanan kromozomların herhangi bir yerinde kullanılabilir. Bu yapıya örnek olarak dizilimi şekil 3.5‘teki gibi olan kromozomun ifade ağacı şekil 3.6 ‘da gösterilmiştir.

01234560101234560101234560101234560123  
 /+a/abbba+\*-abbabb/b\*+abaab+Q/0\*210212

Şekil 3.5. ADF 'li kromozom (Candida, 2006)



Şekil 3.6. ADF 'li kromozom ifade ağacı (Candida, 2006)

### DC kümesi

Sabit sayılardan oluşan bu küme kuyruk kısmından sonra gelir ve boyutu kuyruk ile aynıdır. Bu küme ile amaç GEP 'de rassal sabit sayıların sağladığı yararlılardan faydalanabilmektir.

### 3.2. Çalışmada Kullanılan GEP Tasarımı

Bu tez çalışması kapsamında iki tür statik kural türetme yöntemi kullanılmıştır. Birincisi çok genli ve operasyon temelli parametre temelli kurallardır. İkincisi ise ADF 'gen yapısına sahip kurallardır. Bu iki yöntem ayrıntılı olarak anlatılmıştır.

### 3.2.1. Çok genli statik kurallar

Bu yöntemde kromozomlar sabit uzunluktaki genlerden oluşmaktadır. Bu yapıda kullanılan kümeler aşağıda açıklanmıştır.

#### *Terminal kümesi*

ATÇ ve EATÇ birçok operasyon temelli parametre bulunmasına rağmen bu parametrelerin hepsi çözümün kalitesinin etkilemez. Bu çalışmada terminal kümesinde olarak literatürde sıkça kullanılan ve anlamlı olduğu düşünülen parametrelere yer verilmiştir. Tablo3.1. 'de bu parametreler gösterilmiştir. “?” işareti ile gösterilen yerlere DC kümesinden sabit sayılar gelecektir.

**Tablo 3.1.** Çok genli terminal kümesi

Terminal	Sembol
Serbest kalma zamanı	r
Teslim zamanı	d
İşlem zamanı	p
Geriye kalan işlem zamanı	Re
Toplam işlem zamanı	P
Operasyon teslim zamanı	ODD
Kritik operasyon teslim zamanı	CR-ODD
Rastgele sabit sayı	?

#### *Fonksiyon kümesi*

Fonksiyon kümesi olarak temel matematiksel operatörler kullanılmıştır. Bu operatörler Tablo 3.2. 'de gösterilmiştir.

**Tablo 3.2.** Çok genli fonksiyon kümesi

Fonksiyon	Sembol
Toplama	+
Çıkarma	-
Çarpma	*
Bölme	/

### 3.2.2. ADF gen yapılı statik kurallar

Bu yöntemde kromozomlar sabit uzunlukta birden fazla ADF ‘nin bağlanmasından oluşmaktadır. Genler yerine önceden tanımlanmış kurallar kullanılmaktadır. Bu yapıda kullanılan kümeler aşağıda açıklanmıştır.

#### *Terminal kümesi*

Operasyon temelli parametrelerin yerine literatürde bilinen ve farklı amaçlar için iyi sonuçlar veren sevk etme kuralları terminal kümesini oluşturur. Tablo 3.3. ‘de kullanılan sevk etme kuralları açıklanmıştır. “?” işareti ile gösterilen yerlere DC kümesinden sabit sayılar gelecektir.

**Tablo 3.3.** *ADF gen yapılı terminal kümesi*

<b>Terminal</b>	<b>Tanım</b>
SPT	En kısa işlem süresi
EDD	En erken teslim tarihi
LWKR	Geriye kalan iş yükü
AT	Sisteme geliş zamanı
TWORK	Toplam işlem zamanı
CR	Kritik oran
SLK	Boş kalma zamanı
ODD	Operasyon teslim zamanı
CRODD	Kritik operasyon teslim zamanı
LnOps	Toplam operasyon sayısı
SOP	Operasyonların boş kalma zamanı
SOPN	Geriye kalan operasyonlar için boş zaman
LRnOps	Geriye kalan operasyonlar sayısı
COVERT	Fazla zaman maliyeti

#### *Fonksiyon kümesi*

Fonksiyon kümesi bir önceki bölümdeki ile aynıdır.

### 3.3. Deneysel Çalışma Ortamı Tasarımı

Benzetim ortamında ATÇ, EATÇ, Akış Türü, Esnek Akış Türü ve Açık Atölye Tipi çizelgeleme test problemlerinin türetilmesi için uygun alt yapı tasarlanmıştır. Bu çalışma kapsamında EATÇ problemine odaklanılmıştır.

#### 3.3.1. Benzetim ortamı tasarımı

Benzetim ortamı verilen parametreler ile problemlerin türetilmesiyle başlar. Problemler türetildikten sonra çizelgeleme işlemi başlar. Çizelgeleme yapılırken iki

önemli adım vardır. Birincisi işlerin makinelere atanması, ikinci ise makinelerin kuyruklarında bulunan işlerden birisini seçmesidir. Kuyruktan iş seçimi sevk etme kuralları ile yapılmaktadır. İşlerin makinelere atanması ise Tay ve Ho tarafından 2004 yılında geliştirilen “Least Waiting Time” algoritması ile yapılmaktadır [39]. Benzetim ortamının sözde kodları Ek-1 ‘de mevcuttur. Bu başlığın devamında problemlerin türetilmesinde kullanılan parametreler ve Least Waiting Time algoritması açıklanmıştır.

### 3.3.1.1. Test verisinin türetilmesi

EATÇ problemlerinin türetilmesinde kullanılan yöntem Tay ve Ho ‘nu 2008 yılında yaptıkları çalışmadan yararlanılarak oluşturulmuştur [5]. Toplamda 108 adet test problemi türetilmiştir. Bu test problemleri türetilirken kullanılan parametreler tablo 3.4 ‘te özetlenmiştir.

**Tablo 3.4.** EATÇ problem parametreleri

Parametre	Değer
Esneklik	100% (EATÇ-100), 50%(EATÇ-50), 20%(EATÇ-20)
$nİş \times nMakine$	$10 \times 5, 20 \times 5, 50 \times 5, 20 \times 10, 50 \times 10, 100 \times 10, 50 \times 15, 100 \times 15, 200 \times 15$
$P_{i,j,k}$	$U[(nMakine)/2, (nMakine)*2]$
$r_i$	$nİş \geq 50$ ise $U[0,40]$ ; diğer durumlarda $U[0,20]$
c	1,2 (sıkı), 1,5 (orta), 2 (gevşek)
$d_i$	$r_i + c \times \sum_{j=1}^{n_i} p_{ij}$

İşlerin serbest kalma zamanları iş sayısına bağlı olarak iki grupta toplanmıştır. Eğer iş sayısı 50 den fazla ise işlerin serbest kalma zamanları ortalaması 40 olan uniform dağılıma uymaktadır. Diğer durumlarda ortalaması 20 olan uniform dağılıma uyan serbest kalma zamanları kullanılmaktadır. Esneklik olarak çalışmada 3 farklı yüzde kullanılmıştır. Teslim zamanını belirlerken kullanılan c parametresinin 3 farklı değeri göz önüne alınmıştır. Teslim sıklık parametresine bağlı olarak her esneklik sınıfı için sıkı, orta, gevşek ve karma olarak 4 farklı veri türetilmektedir. Karma ortamda %33 sıkı, %33 orta ve %34 gevşek c parametresi kullanılarak veri türetilmiştir. EATÇ-100 için 4 farklı c parametresi ve 9 farklı problem boyutu ile toplam 36 problem verisi türetilmiştir. EATÇ-50 ve EATÇ-20 ‘de de 36 ayrı veri türetilmiştir. Toplamda 108 adet



farklı problem türetilmiştir. Her bir esneklik sınıfında 2400 tane iş için Cmax, ortalama akış ve ortalama gecikme değerleri hesaplanmıştır.

### 3.3.1.1. *Least waiting time algoritması*

Tay ve Ho tarafından geliştirilen bu algoritma ile işler atanabileceği makinelerden en uygun olanına atanmasını sağlamaktadır. Bu atama işlemi yapılırken 3 tane değer göz önüne alınır. Bu değer;

- Makinede işlenen mevcut işin atama yapılacağı t anında geriye kalan süresi
- Makinenin kuyruğundaki işlerin toplam süresi
- Atanması düşünülen işin o makinede işlem süresidir.

Bu üç değer toplamının minimum olduğu makineye iş atanmaktadır. Bu algoritmanın sözde kodu Ek-2 'da mevcuttur.

### 3.3.2. GEP parametreleri

Bu çalışma kapsamında hem çok genli hem de ADF genli statik kuralların türetilmesinde kullanılan GEP için aynı parametreler kullanılmıştır ve bu parametreler tablo 3.5 'te listelenmiştir.

**Tablo 3.5.** GEP parametreleri

Parametre	Değer
Popülasyon Büyüklüğü	40
Adım Sayısı	100
Çaprazlama Yüzdesi	100%
Seçkinlik (Elitizm) Yüzdesi	10%
Seçim Yöntemi	Koşullu turnuva
Mutasyon Oranı	3%
Uygunluk Fonksiyonu	$\frac{1}{3}C_{max} + \frac{1}{3}F_{ort} + \frac{1}{3}T_{ort}$
Genin Baş Uzunluğun	5, 10
Gen Uzunluğu	16, 31

#### 3.3.2.1. *Başlangıç neslin oluşturma*

Başlangıç nesli oluştururken genin baş kısmının uzunluğu 5 olan 20 tane ve baş kısmını uzunluğu 10 olan 20 tane kural rastgele oluşturulmuştur. Kromozomları oluştururken baş kısmına hem terminal hem de fonksiyon kümesinden parametreler gelirken, kuyruk kısmına sadece terminal kümesindeki parametreler gelmesine izin

verilmektedir. Baş kısım için bu iki kümeden seçim yapılırken her iki kümenin seçilme olasılıkları birbirine eşittir. Yani ilk olarak %50 olasılıkla bir küme seçilir ve sonra o kümedeki elamanlarından biri rastgele seçilerek baş kısmına eklenir. Dc kümesi için oluştururken 1-10 arasındaki tamsayılardan rastgele kuyruk uzunluğu kadar seçilmiştir.

### **3.3.2.2. Uygunluk fonksiyonu**

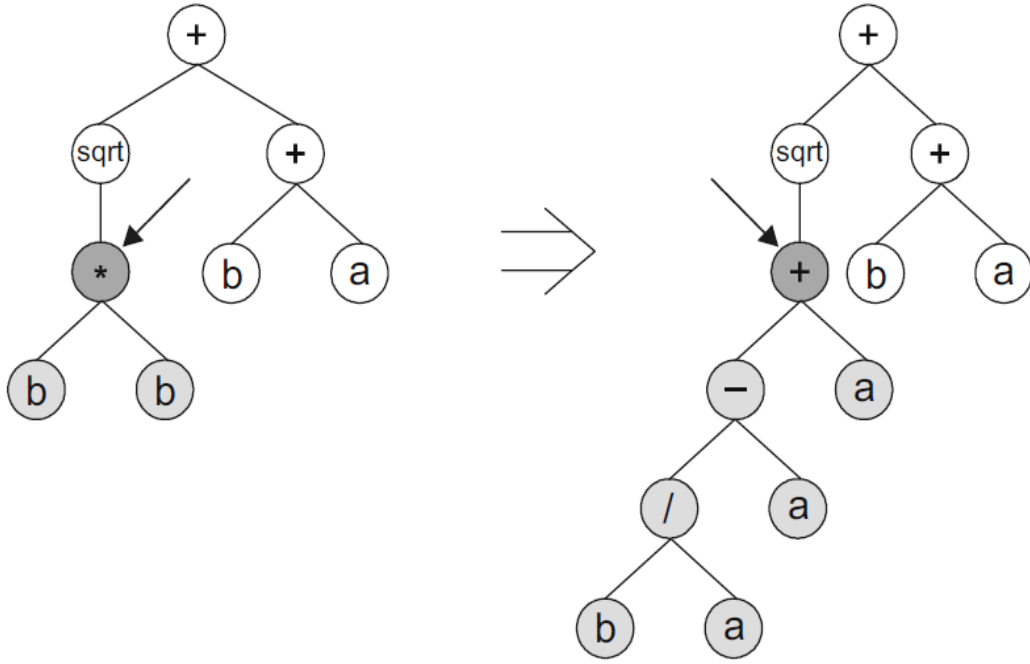
Uygunluk fonksiyonu için toplam tamamlanma zamanı( $C_{max}$ ), ortalama akış ( $F_{ort}$ ) ve ortalama gecikmenin ( $T_{ort}$ ) eşit ağırlıklı doğrusal toplamı kullanılmıştır. Bu toplam ne kadar küçükse kural o kadar etkindir ve bir sonraki nesle aktarım şansı o kadar fazla olmaktadır.

### **3.3.2.3. Seçkinlik (Elitizm)**

Tüm bireyler uygunluk fonksiyonlarına göre sıralandığında ilk %10'luk kısımdaki bireyler direk olarak bir sonraki nesle aktarılmıştır.

### **3.3.2.4. Çaprazlama**

Çaprazlama yüzdesi olarak %100 alınmıştır ve tek noktada çaprazlama işlemi yapılmaktadır. Çaprazlama işlemi kromozom üzerinde yapılmaktadır ve çaprazlama sonucunda kuralın ifade ağacını boyutu değişebilmektedir. Buna örnek olarak şekil 3.7 'te koyu renkteki kısım ile başka kromozomdan gelen kısım birleşince ağacın derinliğinin 3' den 5 'e yükseldiği görülmektedir.



**Şekil 3.7.** Çaprazlama sonrası ağaç derinliği değişimi (Candida, 2006)

### 3.3.2.5. Mutasyon (Elitizm)

Mutasyon oranı olarak %3 kullanılmıştır. Mutasyon yönteminde seçilen bir genin yerine rastgele yeni bir gen yerleştirilmektedir. Bu işlem yapılırken uygunluğun bozulmaması için aşağıdaki maddelere bağlı kalmıştır.

- Baş kısmına mutasyon yapılacaksa fonksiyon, terminal veya Dc kümelerinden herhangi birindeki değer seçilebilir.
- Kuyruk kısmına mutasyon yapılacaksa fonksiyon kümesi haricindeki kümelerdeki herhangi bir eleman seçilerek yerleştirme yapılabilir.

### 3.3.2.6. Seçim

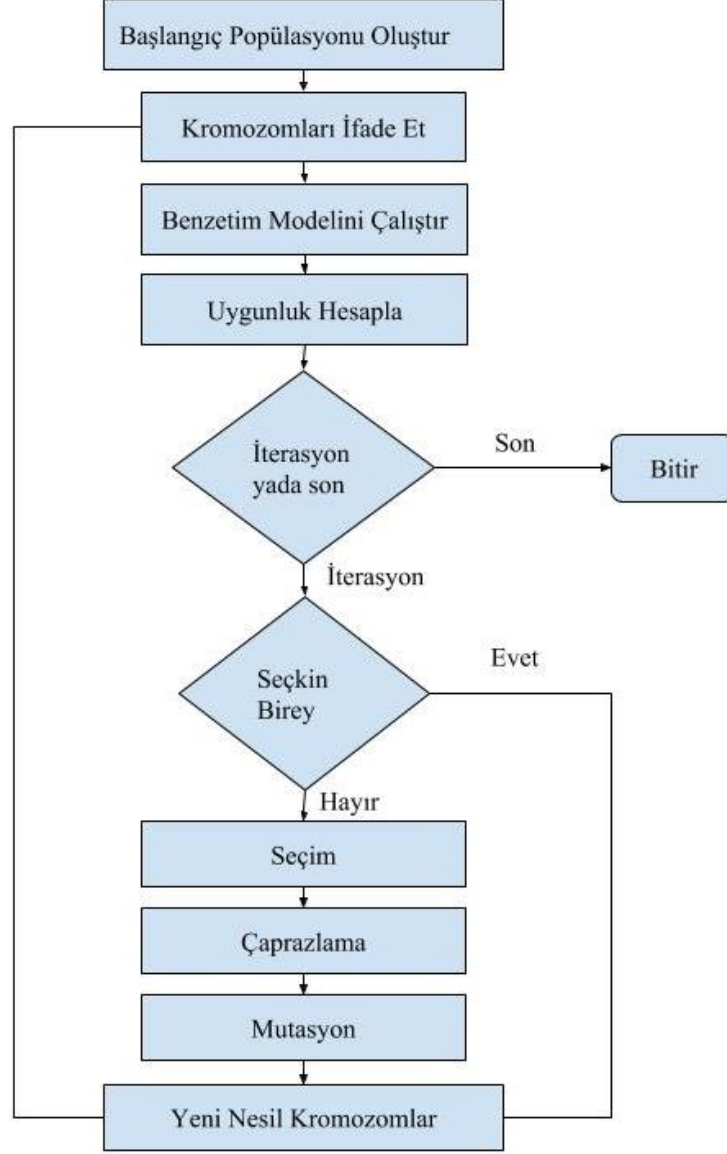
Yeni nesil oluşturmada kullanılacak ebeveynleri seçmek için literatürde bir çok yöntem bulunmaktadır. Bu çalışmada koşullu turnuva yöntemi kullanılmıştır. Bu yöntemde rastgele iki birey alınır ve iyi olan birey %70 olasılıkla çaprazlama işlemi için havuza konulur. %30 olasılıkla kötü birey seçimine izin verilerek erken yakınsama problemi giderilmiştir.

### 3.3.2.7. Durdurma kriteri

Durdurma kriteri olarak adım sayısı seçilmiştir. 100 adımdan sonra program durdurulmuştur.

### 3.3.2.8. GEP benzetim modeli

Kurallar türetilmesinde kullanılan benzetim modelinin akış grafiği şekil 3.8 'de gösterilmiştir.



Şekil 3.8. GEP benzetim ortamı akış şeması

Öncelikle başlangıç bireyler oluşturulmuştur. Daha sonra kromozom yapısı tasvir edilmiştir. Sonraki adımda benzetim modeli çalıştırılarak her kromozom için sonuçlar elde edilmektedir. Bu sonuçlar ile kromozomların uygunluk değerleri hesaplanıp, bu değerlere göre sıralama yapılmaktadır. Eğer son adıma gelinmişse program durdurulup ve eldeki kurallar içinden iyi olanlar seçilmektedir. Son adım değilse sıralanmış kromozomlar içinde elit bireyler bir sonraki nesle aktarılmaktadır. Daha sonra seçim işlemi ile bir sonraki nesil için üreme havuzu oluşturulmaktadır. Bu havuzdaki bireyler

çaprazlama ve mutasyon operasyonlarına tabii tutularak yeni nesil kromozomlar oluşturulup ve tekrar bu döngü devam etmektedir.

### **3.4. Deneysel Sonuçlar ve Analiz**

Açıklanan algoritmalar, Intel® Xenon®, 2.93 GHz çift işlemcili ve 8.00 GB RAM bulunan bir sistemde Python 2.7.10 dili ile uygulanmıştır. Literatürde bulunan ve farklı amaçlar için iyi sonuçlar veren kurallar ile türetilen kurallar benzetim ortamında analiz edilmiştir. 30 tekrar sonucunda elde edilen sonuçlar bu bölümün devamında ayrıntılı olarak incelenmiştir. İlk olarak literatürden seçilen kurallar hakkında kısa bilgiler verilmiştir.

#### **3.4.1. Literatürden seçilen kurallar**

Aşağıda literatürden seçilen kurallar kısaca özetlenmiştir.

1. FIFO, ilk giren ilk çıkar mantığı ile çalışır.
2. SPT, en kısa operasyon süresi olan işe öncelik verilir.
3. EDD, en erken teslim zamanlı işe öncelik verilir.
4. ODD, en erken teslim tarihi olan operasyona öncelik verilir.
5. SPT+LWKR+SLK, Öztürk ve Teymaourifer tarafından geliştirilen bir kuraldır [40].
6. CR, işlerin boş zamanı üzerinden hesaplanan kritik oran temelli bir kuraldır.
7. COVERT, işlerin beklenen boş kalma süresi üzerinden hesaplanan bir ceza fonksiyonu kullanarak oluşturulan kuraldır [41].
8. CRODD, kritik oran ile operasyon teslim tarihinin kurallarının birleşiminden oluşan kuraldır.
9. AT, geliş zamanı erken olana öncelik verilir.
10.  $r+p+2P$ , Tay ve Ho tarafından geliştirilen kuraldır [5].
11.  $7P+11p+12(nOps+r)$ , Tay ve Ho tarafından geliştirilen kuraldır [5].

#### **3.4.2. Türetilen kurallar**

Yapılan çalışma sonucunda türetilen kurallar tablo 3.6 'da özetlenmiştir. Bu kurallardan ilk 8'i ADF 'li yapı ile elde edilen kurallarken, diğer 6 'sı ADF' siz elde edilen kurallardır.

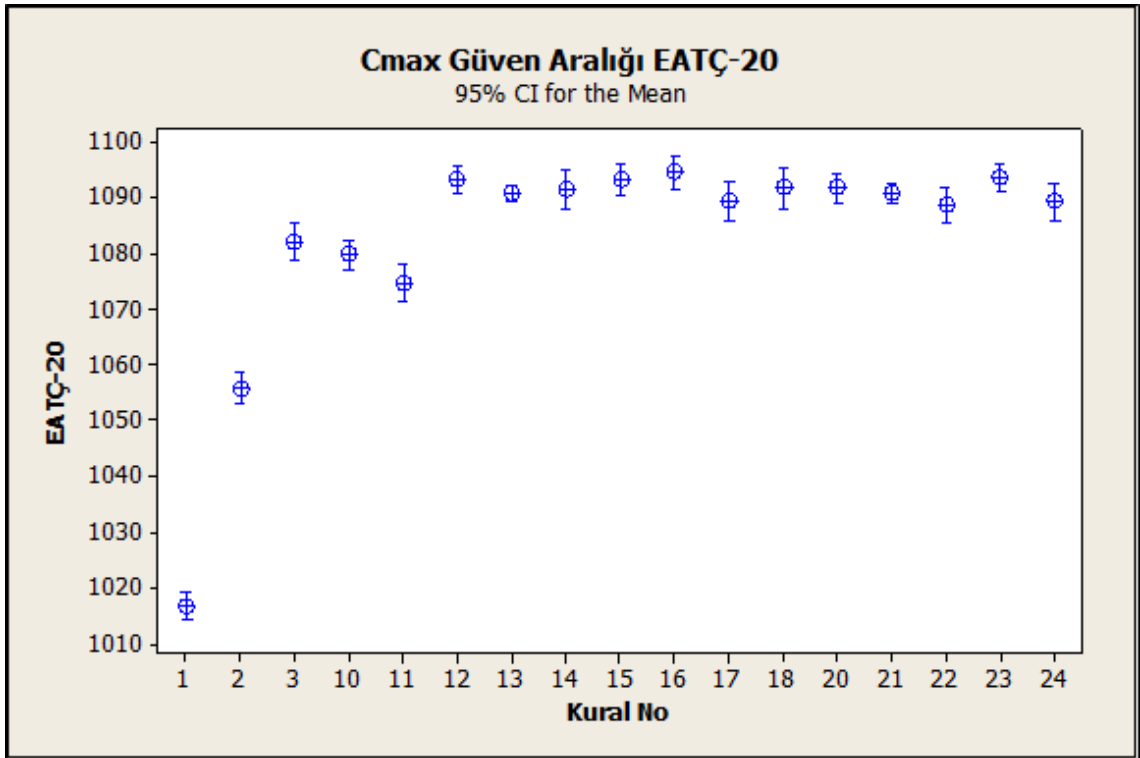
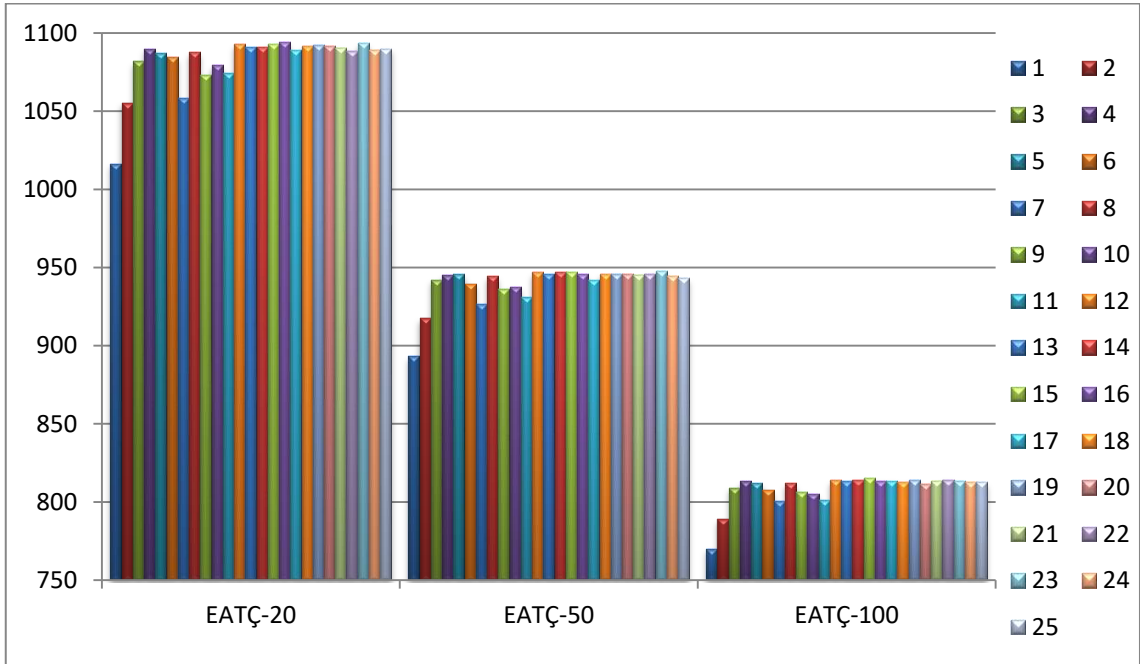
**Tablo 3.6. Türetilen kurallar**

Kural Numarası	Formül	Tür
12	$[(LRnOps * TWORk) + SPT] * LRnOps$	ADF 'li
13	$LRnOps + [(CRODD/9) - (COVERT/SPT)]$	
14	$TWORK/3 + LnOps - LRnOps$	
15	$LRnOps * LWKR$	
16	$(LRnOps + 1) * LWKR$	
17	$[EDD + [(LRnOps + TWORk) / (LWKR - TWORk)] * LnOps] * LRnOps$	
18	$LWKR + [(LRnOps * LRnOps) / (COVERT - COVERT + 0.00000001)] - SOP$	
19	$LRnOps * LWKR + LWKR - 3LnOps$	
20	$(CR/3ODD) * CRODD + ODD$	
21	$ODD + CRODD - Re + (Re + 1)$	
22	$ODD + Re$	
23	$P * Re$	
24	$(ODD - CRODD * CRODD * CRODD) * Re$	
25	$(d - CRODD * CRODD * CRODD) * Re$	

### 3.4.3. Cmax için sonuçlar

Tablo3.7’de de görüldüğü üzere FIFO kuralı bütün esneklik seviyelerinde maksimum tamamlama zamanı (Cmax) için en iyi sonucu vermektedir. FIFO kuralı işlerin geliş zamanlarını göz önüne aldığı için en iyi kural çıkmıştır. İkinci en iyi kural olarak ise SPT kuralı çıkmıştır. İyi sonuçlar veren kurallar ile türetilen kuralların istatistiksel analizi One Way Anova ve Tukey Testi kullanılarak yapılmıştır. Şekil 3.9 ‘da %20 esneklik seviyesi için türetilen kuralları ve iyi sonuç veren kuralların güven aralığı grafiği verilmiştir. Tablo 3.8 ‘de ise %20 esnek seviyesinde seçilen kuralların Tukey testi sonucunda oluşturulan grup durumları gösterilmektedir. Bu iki veriden hareketle ADF ‘li ile ADF ‘siz kurallar Cmax amacı için istatistiksel olarak anlamlı bir farka sahip olmadıkları sonucuna ulaşılmıştır. Türetilen kurallar FIFO’ dan iyi sonuç vermiş olmasalar bile diğer kurallara yakın sonuçlar vermişlerdir. %50 ve %100 esneklik seviyelerinin güven aralığı grafikleri ve Tukey test sonuçları Ek-3 ‘te sunulmuştur.

**Tablo 3.7.** *Cmax sonuçlar*



**Şekil 3.9.** *%95 güven seviyesinde EATÇ-20 için Cmax güven aralığı*

**Tablo 3.8.** *Cmax Tukey test sonuçları*

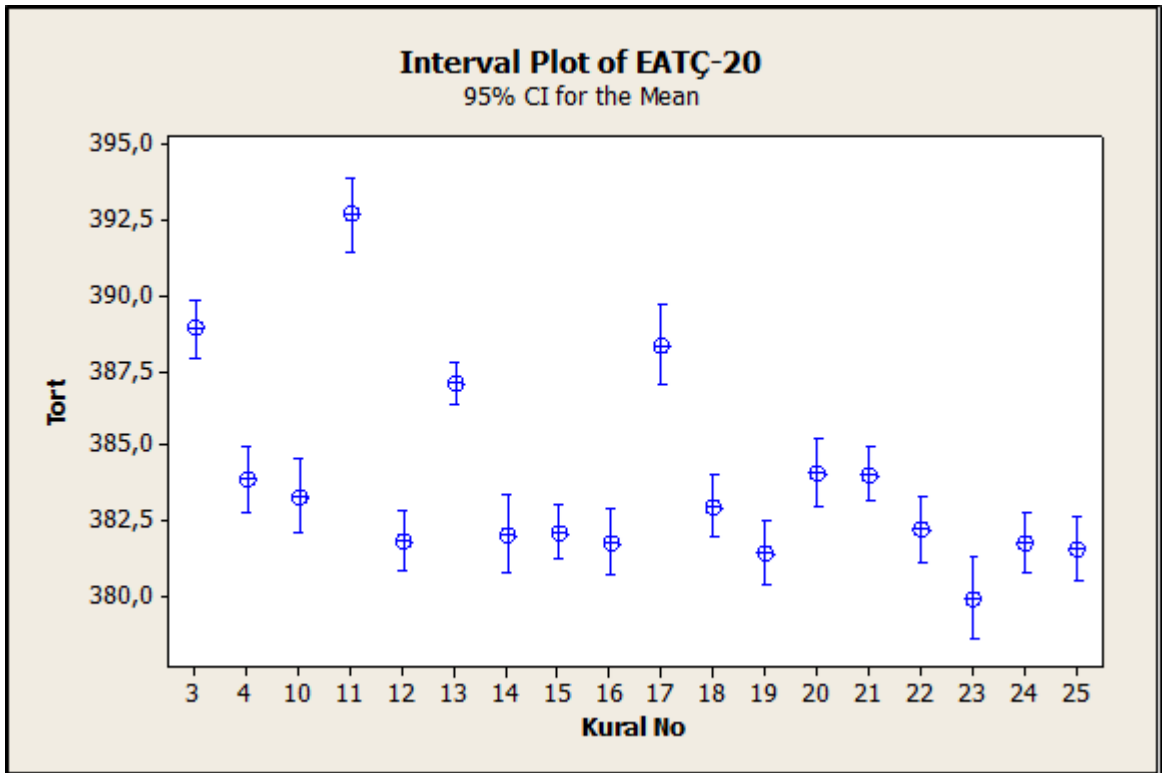
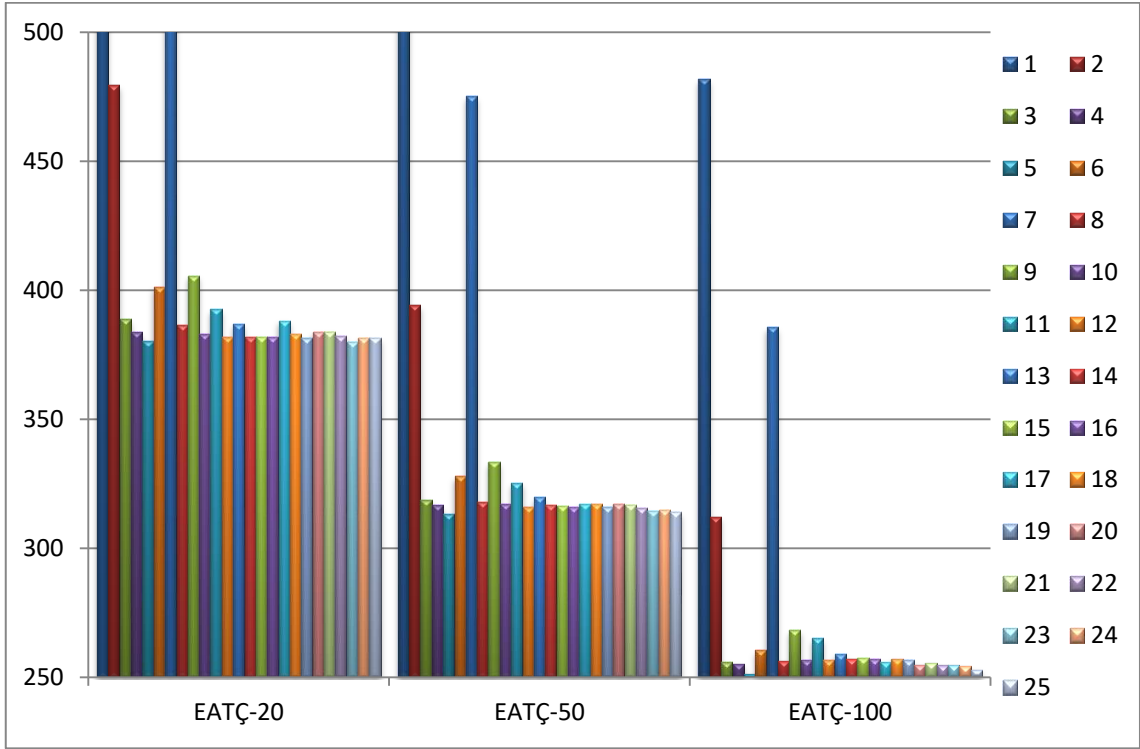
<b>Kural numarası</b>	<b>Örnek sayısı</b>	<b>Ortalama</b>	<b>Grup</b>
1	10	1016,682	E
2	10	1055,533	D
3	10	1082,09	B
10	10	1079,723	B,C
11	10	1074,625	C
12	10	1093,004	A
13	10	1090,837	A
14	10	1091,448	A
15	10	1093,147	A
16	10	1094,401	A
17	10	1089,268	A
18	10	1091,655	A
20	10	1091,585	A
21	10	1090,763	A
22	10	1088,677	A
23	10	1093,63	A
24	10	1089,187	A

#### **3.4.4. Ortalama gecikme (Tort) için sonuçlar**

Tablo 3.9’de görüldüğü üzere ADF yöntemi ile türetilen kurallar tüm esneklik seviyelerinde ortalama gecikme bakımından çok iyi sonuçlar vermektedirler. Ayrıca diğer türetilen kurallarda EDD ve ODD gibi teslim zamanı temelli kurallarla rekabet edebilecek sonuçlar vermiştir. Cmax için en iyi sonuçları veren FIFO ve SPT kuralları bu amaç için kötü sonuçlar vermektedir. Minitab programında yapılan One Way ANOVA ve Tukey Testi ile, Tort değerlerinin analizi yapılmıştır. Tort’nın %20 esneklik seviyesi için Şekil 3.10’ da güven aralığı grafiği ve Tablo 3.10’ da Tukey test sonucu verilmiştir. Bu verilere bakıldığında, türetilen kurallardan çoğu diğer kurallardan istatistiksel olarak daha iyi sonuçlar vermektedir. Düşük esneklik seviyelerinde ADF’li ve ADF’siz kurallar anlamlı bir farka sahip değildir. Fakat esneklik seviyesi arttıkça ADF’siz kurallar daha iyi sonuçlar vermektedir. Bununla birlikte, kurallar kendi içerisinde tutarlı bir yapıya sahiptir ve bu kullanılan yöntemin ne kadar etkin olduğunun kanıtıdır. %50 ve %100 esneklik seviyelerinin güven aralığı ve Tukey test sonuçları Ek-4’de verilmiştir.



**Tablo 3.9.** *Tort sonuçlar*



**Şekil 3.10.** *%95 güven seviyesinde EATÇ-20 için Tort güven aralığı*

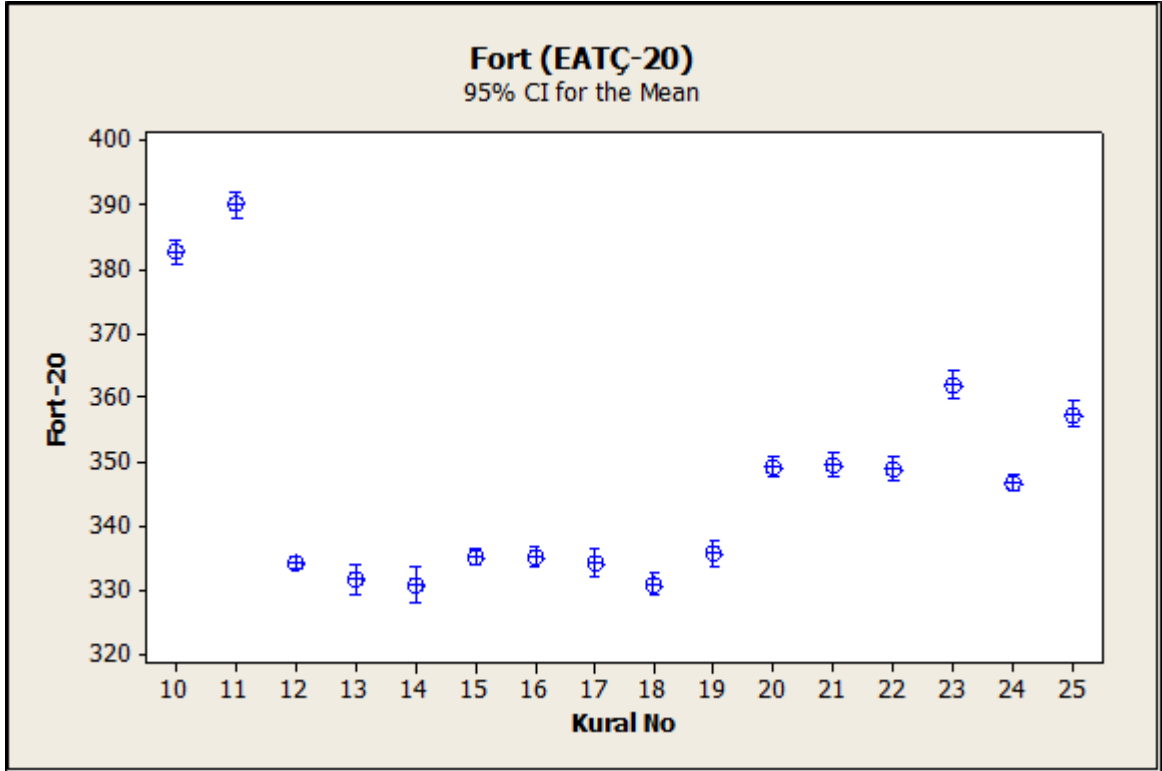
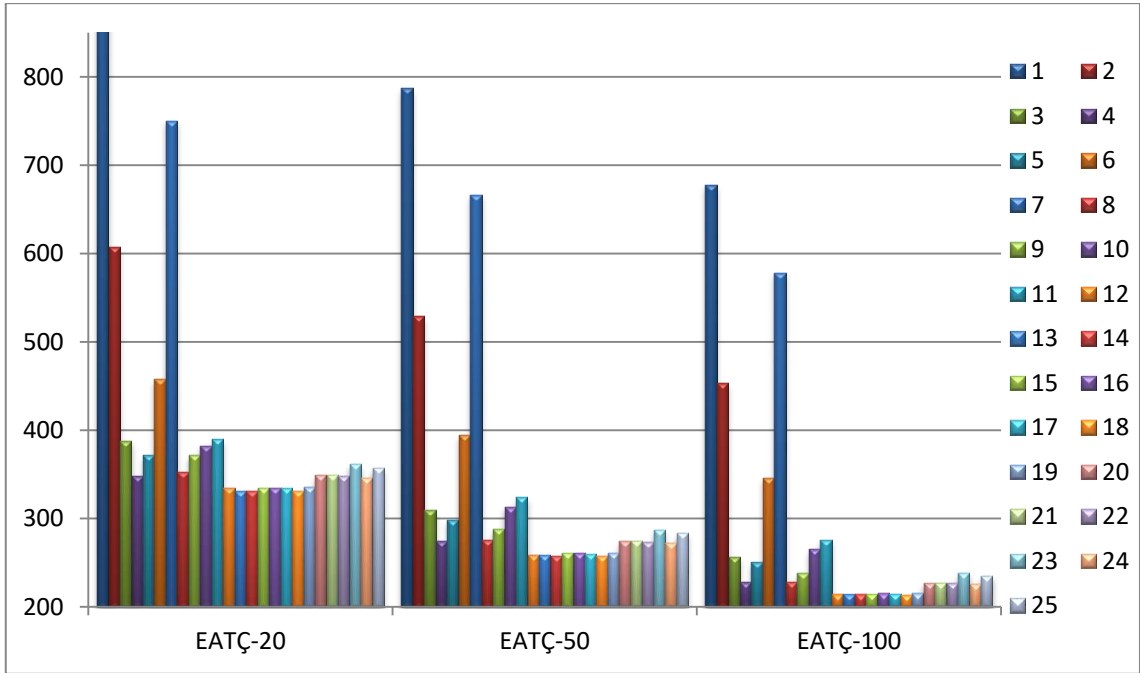
**Tablo 3.10.** *Tort Tukey test sonuçları*

Kural numarası	Örnek sayısı	Ortalama	Grup
3	10	388,907	B
4	10	383,87	C,D
10	10	383,317	C,D,E
11	10	392,706	A
12	10	381,835	C,D,E,F
13	10	387,096	B
14	10	382,06	C,D,E,F
15	10	382,113	C,D,E,F
16	10	381,79	C,D,E,F
17	10	388,352	B
18	10	382,987	C,D,E
19	10	381,406	E,F
20	10	384,077	C
21	10	384,041	C
22	10	382,199	C,D,E,F
23	10	379,925	F
24	10	381,754	C,D,E,F
25	10	381,567	D,E,F

#### **3.4.5. Ortalama akış (Fort) için sonuçlar**

Tablo 3.11 'de görüldüğü üzere ADF ile türetilen kural14 %20 ve %50 esnekliklerde ortalama akış için en iyi sonucu verirken yine ADF ile türetilen kural18 %100 esneklik seviyesinde en iyi sonucu vermiştir. ADF 'li kurallar, ADF 'siz kurallar ve iyi sonuç veren kurallar One Way Anova ve Tukey testi ile istatistiksel olarak analizi edilmiştir. Fort değeri için %20 esneklik düzeyinde, Tablo 3.12 Tukey test sonuçlarını ve şekil 3.11 güven aralığı grafiğini göstermektedir. Bu analizler sonucunda, %20 esneklik seviyesinde ADF ile türetilen kurallar kendi aralarında istatistiksel olarak anlamlı bir farka sahip değildir. Bu kurallar ADF 'siz kurallar ve literatürdeki diğer kurallardan istatistiksel olarak daha iyi sonuç vermişlerdir. Diğer türetilen kurallarda literatürdeki kurallardan daha iyi sonuçlar vermişlerdir. %50 ve %100 esneklik düzeyleri için güven aralığı grafikleri ve Tukey test sonuçları Ek-5 'te sunulmuştur.

**Tablo 3.11.** Fort sonuçlar



**Şekil 3.11.** %95 güven seviyesinde EATÇ-20 için Fort güven aralıkları

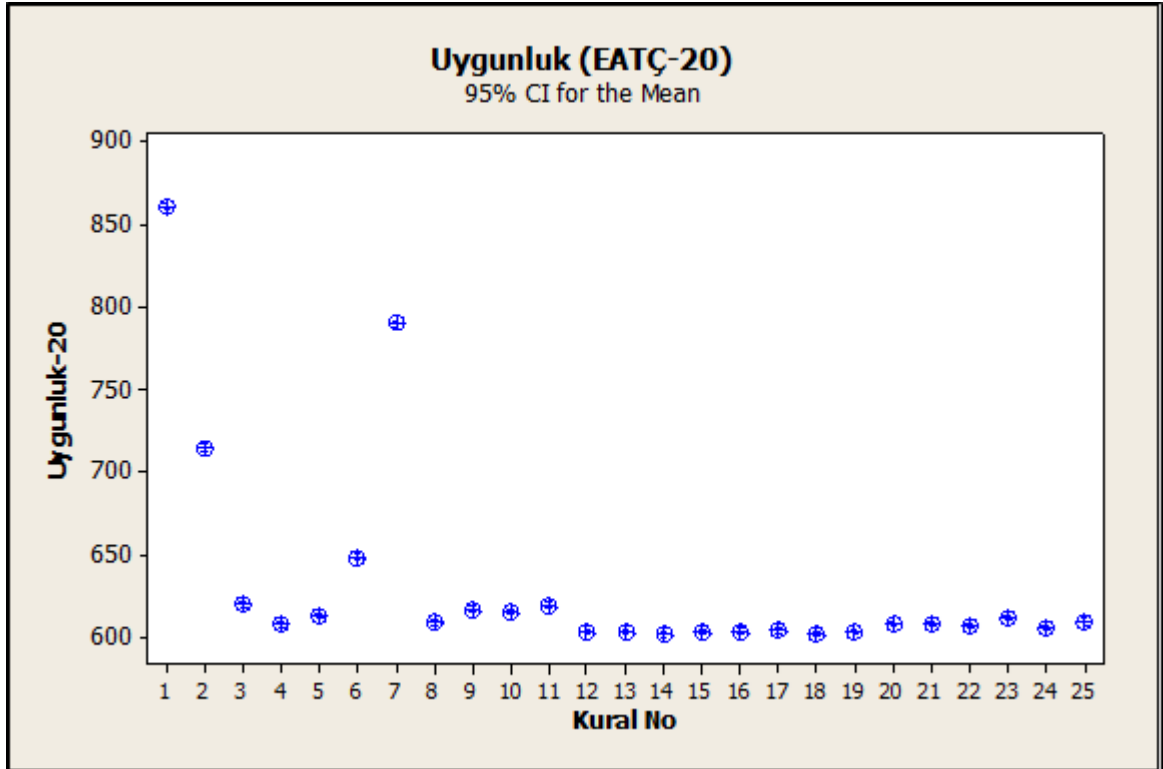
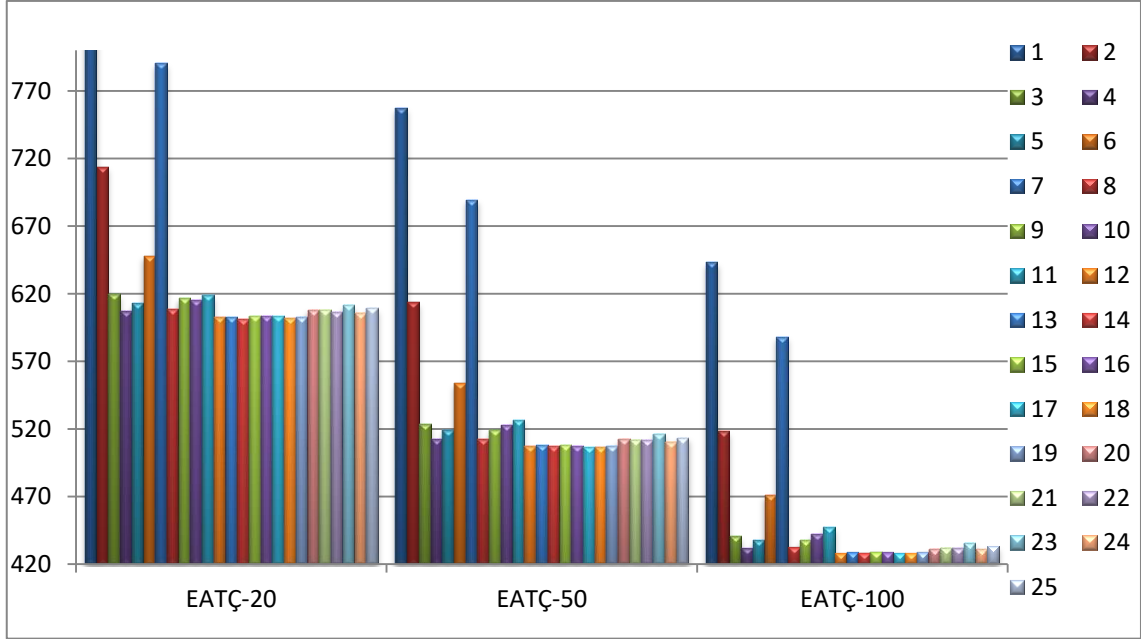
**Tablo 3.12.** Fort Tukey test sonuçları

Kural numarası	Örnek sayısı	Ortalama	Grup
10	10	382,634	B
11	10	389,975	A
12	10	334,275	F,G
13	10	331,64	F,G
14	10	330,937	G
15	10	335,223	F
16	10	335,279	F
17	10	334,306	F,G
18	10	330,961	G
19	10	335,672	F
20	10	349,074	E
21	10	349,338	E
22	10	348,815	E
23	10	361,936	C
24	10	346,662	E
25	10	357,323	D

#### 3.4.6. Çok amaçlı fonksiyon için sonuçlar

Tablo 3.13 'de görüldüğü üzere ADF ile türetilen kurallar tüm esneklik seviyelerinde en iyi sonuçları vermişlerdir. Diğer yöntem ile türetilen kurallar ise literatürdeki kuralların hepsinde tüm esneklik seviyelerinde ya daha iyi sonuçlar yada benzer sonuçlar vermişlerdir. Literatürde bulunan kurallar arasında kıyaslama yaptığımızda en iyi sonuçları veren kurallar CRODD ve ODD kurallarıdır. Genel olarak FIFO, COVERT, SPT ve CR kuralları çok amaçlı problemlerde en kötü sonuçları veren kurallar olarak göze çarpmaktadır. Ayrıca istatistiksel olarak sonuçları analiz ettiğimizde ADF 'li kurallar kendi aralarında ve ADF 'siz türetilen kurallar kendi aralarında anlamlı bir farka sahip değildirler. Bu da türetme yönteminin tutarlılığını gözler önüne sermektedir. Türetilen kurallar ve diğer tüm kuralların %95 güven seviyesinde %20 esneklik düzeyi için yapılan varyans analizleri sonucunda oluşan güven aralığı grafiği şekil 3.12' de gösterilmiştir. Ayrıca yapılan Tukey testi ile elde edilen gruplar Tablo 3.13 'te gösterilmiştir. %50 ve %100 esneklik seviyelerinin güven aralığı grafikleri ve Tukey test sonuçları Ek-6 'da sunulmuştur.

**Tablo 3.13. Çok amaçlı sonuçlar**



**Şekil 3.12. %95 güven seviyesinde EATÇ-20 için çok amaçlı güven aralığı**

**Tablo 3.14.** Çok amaçlı test sonuçları

<b>Kural numarası</b>	<b>Örnek sayısı</b>	<b>Ortalama</b>	<b>Grup</b>
1	10	861,04	A
2	10	714,25	C
3	10	619,76	E
4	10	607,52	J,K,L
5	10	613,13	G,H
6	10	648,19	D
7	10	790,43	B
8	10	609,18	I,J,K
9	10	616,97	E,F
10	10	615,23	F,G
11	10	619,1	E
12	10	603,04	M,N
13	10	603,19	M,N
14	10	601,48	N
15	10	603,49	M,N
16	10	603,82	M,N
17	10	603,98	L,M,N
18	10	601,87	N
19	10	603,27	M,N
20	10	608,25	I,J,K
21	10	608,05	J,K
22	10	606,56	J,K,L,M
23	10	611,83	G,H,I
24	10	605,87	K,L,M
25	10	609,63	H,I,J

#### 4. SONUÇLAR ve TARTIŞMA

Sevk etme kuralları, uygulama açısında kolay olduđu ve iyi sonuçlar verdiđi için dinamik çizelgeleme ortamlarında yaygın olarak kullanılmışlardır. Literatürde basit ve kompozit yapıda iyi sonuçlar veren birçok sevk etme kuralı vardır. Bu kurallar farklı amaçlar için oldukça iyi sonuçlar vermektedir. Fakat çok amaçlı bir probleme iyi çözümler üretmekte çok başarılı değillerdir. Bu kurallar genellikle yetkin kişilerin becerileri ile türetildikleri için çok kapsamlı bir yapıya sahip değillerdir. Üretim parametrelerin etkinliđi ortamdaki ortama farklılıklar göstermektedir. Bu yüzden üretim ortamlarının önemli özelliklerini benzetim ile tespit ederek ve bu özelliklerden yararlanarak yeni kurallar üretmek oldukça önemlidir. Genel olarak iş temelli parametreler literatürde yaygın olarak kullanılmaktadır, fakat benzetim sonuçları operasyon temelli parametrelerin daha hassas olduğunu göstermiştir. Bu iki tür parametrelerin doğrusal ve doğrusal olmayan kombinasyonları ile oluşan kurallar ile daha iyi sonuçlar elde edilmektedir. Ayrıca, literatürde bulunan kuralların doğrusal ve doğrusal olmayan birleşimi ile birleşik kural üretme yöntemi oldukça etkili sonuçlar vermektedir. Bu çalışma kapsamında, yeni sevk etme kuralı çıkarımı için evrimsel öğrenme temelli GEP kullanılmıştır. Çizelgeleme problemlerinde daha önce bu yaklaşım kullanılmamıştır. Bu yaklaşım ile hem üretim parametrelerini temel alan kurallar hem de literatürdeki kuralların birleşimi ile elde edilen kurallar türetilmiştir. Fonksiyon setinde bulunan “+” ve “-” operatörlerinin yanında “x” ve “/” operatörleri kullanılarak, hem doğrusal hem de doğrusal olmayan birleşimlerle etkin kurallar türetilmiştir. GEP sayesinde verimli olan parametre ve ADF ‘ler ayıklanarak kuralların performansları artırılmıştır. Yapılan çalışmalar One Way Anova ile istatistiksel olarak analiz edilmiş ve güven aralığı grafikleri çizilmiştir. İstatistiksel çalışmalar sonucunda bileşik kurallar çok amaçlı problemler için basit kurallardan daha iyi sonuçlar verdiđi gözlemlenmiştir. Ayrıca ADF yapıları kurallar, ADF yapısız kurallardan çok amaçlı problemler için daha iyi sonuç verdikleri gözlemlenmiştir. Buna karşın, ADF ‘siz kurallar birçok problemde literatürdeki diđer kurallardan daha iyi sonuçlar vermiştir.

Bu çalışma ileriki çalışmaların temeli olarak görülebilir. Çünkü bu çalışma ortamı gerçek hayata oldukça uygundur. Fakat ideal sistem için tasarım yapılmıştır. İleriki çalışmalarda makinelerin bozulmaları, makine tamir süreleri ve buffer kullanımı gibi gerçek hayat parametreleri kullanılarak daha gerçekçi bir ortam üzerinde çalışmalar yapılabilir. Ayrıca operasyonları makinelere atarken daha farklı ve etkin algoritmaların

kullanımı ile daha iyi sonuçlar elde edilebilir. Amaç fonksiyonu bakımından daha farklı amaçlar göz önüne alınabilir ve bu amaçların öncelikleri birbirinden farklı olabilir.



## KAYNAKÇA

- [1] Pinedo, M. L. (2008). *Scheduling Theory Algorithms and Systems*. New York: Springer Science+Business Media.
- [2] Pinedo, M. L. (2005). *Planning and Scheduling in Manufacturing and Services*. New York: Springer Science+Business Media, Inc.
- [3] Jain, A., ve Meeran, S. (1999). Deterministic job-shop scheduling: Past, present and future. *European Journal Of Operational Research*, 390-434.
- [4] Lawler, E. L. (1993). Sequencing and scheduling: Algorithms and complexity. E. L. Lawler, J. K. Lenstra, A. H. Kan, ve D. B. Shmoys içinde, *Handbooks in Operations Research and Management Science* (s. 445-522).
- [5] Tay, J. C., ve Ho, N. B. (2008). Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Computers ve Industrial Engineering*, 453-473.
- [6] Rajendran, C., ve Holthaus, O. (1997). Efficient dispatching rules for scheduling in a job shop. *International Journal of Production Economics*, 87-105.
- [7] Rajendran, C., ve Holthaus, O. (1999). A comparative study of dispatching rules in dynamic Flowshops and Jobshops. *European Journal of Operational Research*, 156-170.
- [8] Johnsan, S. M. (1954). Optimal Two- and Three-Stage Production Schedules with Set-up Times Included. 61-68.
- [9] Hanssmann, F., ve Hess, S. W. (1960). A linear programming approach to production and employment scheduling. *Management science*, 46-51.
- [10] Heald, M., ve Karp, R. M. (1962). A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied Mathematics* , 10(1), 196-210.
- [11] Gupta, S. K. (1982). N jobs and m machines job-shop problems with sequence-dependent set-up times. *The International Journal of Production Research*, 20(5), 643-656.
- [12] Brucker, P., ve Schlie, R. (1990). Job-shop scheduling with multi-purpose machines. *Computing*, 45(4), 369-375.
- [13] Lin, S.-C., Goodman, E. D., ve III, W. F. (1997). A Genetic Algorithm Approach to Dynamic Job Shop Scheduling Problem. *ICGA*.

- [14] Demirkol, E., Mehta, S., ve Uzsoy, R. (1997). A Computational Study of Shifting Bottleneck. *Journal of Heuristics*, 111–137.
- [15] Pezzella, F., ve Merelli, E. (2000). A tabu search method guided by shifting bottleneck for the job shop scheduling problem. *European Journal of Operational Research*, 120(2), 297-310.
- [16] Zhang, G., Gao, L., Li, X., ve Li, P. (2008). Variable Neighborhood Genetic Algorithm for the Flexible Job Shop Scheduling Problems. *Intelligent Robotics and Applications*, 503-512.
- [17] Li, X., ve Gao, L. (2016). An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. *International Journal of Production Economics*, 93-110.
- [18] Yazdania, M., Aletib, A., Khalilia, S. M., ve Jolaia, F. (2017). Optimizing the sum of maximum earliness and tardiness of the job shop scheduling problem. *Computers ve Industrial Engineering*, 12-24.
- [19] Yin, L., Li, X., Gao, L., Lu, C., ve Zhang, Z. (2017). A novel mathematical model and multi-objective method for the low-carbon flexible job shop scheduling problem. *Sustainable Computing: Informatics and Systems*, 15-30.
- [20] Mansouri, S. A., Aktas, E., ve Besikci, U. (2016). Green scheduling of a two-machine flowshop: Trade-off between makespan and energy consumption. *European Journal of Operational Research*, 248(3), 772-788.
- [21] Wang, Y.-C., Wang, M.-J., ve Lin, S.-C. (2017). Selection of cutting conditions for power constrained parallel machine scheduling. *Robotics and Computer-Integrated Manufacturing*, 43, 105-100.
- [22] Blackstone, J., Phillips, D., ve Hogg, G. (1982). A state-of-the-art survey of dispatching rules for manufacturing job shop operations. *The International Journal of Production Research*, 20(1), 27-45.
- [23] Rochette, R., ve Sadowski, R. P. (1976). A statistical comparison of the performance of simple dispatching rules for a particular set of job shops. *The International Journal of Production Research* , 14(1), 63-75.
- [24] Panwalkar, S. S., ve Iskander, W. (1977). A survey of scheduling rules. *Operations research*, 25(1).

- [25] Anderson, E. J., ve Nyirenda, J. C. (1990). Two new rules to minimize tardiness in a job shop. *The International Journal of Production Research*, 28(12), 2277-2292.
- [26] Pierreval, H., ve Mebarki, N. (1997). Dynamic scheduling selection of dispatching rules for manufacturing system. *International Journal of Production Research* , 35(6), 1575-1591.
- [27] Chiang, T.-C., ve Fu, L.-C. (2007). Using dispatching rules for job shop scheduling with due date-based objectives. *International journal of production research*, 45(14), 3245-3262.
- [28] Sels, V., Gheysen, N., ve Vanhoucke, M. (2012). A comparison of priority rules for the job shop scheduling problem under different flow time- and tardiness-related objective functions. *International Journal of Production Research*, 50(15), 4255-4270.
- [29] Doh, H.-H., Yu, J.-M., Kim, J.-S., Lee, D.-H., ve Nam, S.-H. (2013). A priority scheduling approach for flexible job shops with multiple process plans. *International Journal of Production Research*, 51(12), 3748-3764.
- [30] Jing, H., ve Süer, G. A. (2015). A dispatching rule-based genetic algorithm for multi-objective job shop scheduling using fuzzy satisfaction levels. *Computers ve Industrial Engineering*, 86, 29-42.
- [31] Lu, H., Huang, G. Q., ve Yang, H. (2011). Integrating order review/release and dispatching rules for assembly job shop scheduling using a simulation approach. *International Journal of Production Research* , 49(3), 647-669.
- [32] Jayamohan, M. S., ve Rajendran, C. (2000). New dispatching rules for shop scheduling: a step forward. *International Journal of Production Research*, 38(3), 563-586.
- [33] Xiong, H., Fan, H., Jiang, G., ve Li, G. (2017). A simulation-based study of dispatching rules in a dynamic job shop scheduling problem with batch release and extended technical precedence constraints. *European Journal of Operational Research*, 257(1), 13-24.
- [34] Chen, B., ve Matis, T. (2013). A flexible dispatching rule for minimizing tardiness in job shop scheduling. *International Journal Production Economics*, 141, 360-365.

- [35] Elhüseyni, M. (2012). Hipotetik bir tekstil atölyesinde dinamikaik çizzelgelenmesinde yollama kurallarını benzetim tekniğiyle analizi. Yüksek Lisan Tezi, İstanbul Teknik Üniversitesi, İstanbul.
- [36] Pezzella, F., Morganti, G., ve Ciaschetti, G. (2008). A genetic algorithm for the flexible job-shop scheduling problem. *Computers ve Operations Research*, 35(10), 3202-3212.
- [37] Ferreira, C. (2006). *Gene expression programming: mathematical modeling by an artificial intelligence*. Springer.
- [38] Ferreira, C. (2002). Gene expression programming in problem solving. *Soft computing and industry*, 635-653.
- [39] Ho, N. B., ve Tay, J. C. (2004). GENACE: An efficient cultural algorithm for solving the flexible job-shop problem. *Evolutionary Computation*, 1759-1766.
- [40] Ozturk, G., ve Teymourifar, A. (2017). New dispatching rules and due date assignment model for dynamic job shop scheduling problems. Accepted to *International Journal of Manufacturing Research*.
- [41] Ruessell, R. S., Dar-el, E. M., ve III, B. W. (1987). A comparative analysis of the COVERT job sequencing rule using various shop performance measures. *International Journal of Production Research*, 1523-1540.

## EK-1

### Algorithm 1: Simulation general steps at each iteration

---

**Data:** problem set, PRs set  
**Data:** schedules (each PR)

- 1 **Call function:** Read data set;
- 2 **foreach**  $PR \in PRs\ set$  **do**
- 3     start time  $\leftarrow \min (Job_j.R)$ , for  $j = 1, \dots, N$ ;
- 4     **Call function:** Initialize matrices;
- 5     Assign operations randomly to machines with random order and start them;
- 6     **Call function:** Update operations and machines state according to the new started operations;
- 7     Remove *start time events* from *next events list*;
- 8     **while** *not finished operations number*  $\geq 1$  **do**
- 9         **Call function:** Find next time, and events;
- 10        Go forward to next time (current time  $\leftarrow$  next time);
- 11        **if** *any operations are released* **then**
- 12            **Call function:** Least waiting time assignment;
- 13        **end**
- 14        **Call function:** Update machines states;
- 15        **if** *any operations are started just now* **then**
- 16            **Call function:** Update operations and machines state according to the new started operations;
- 17        **end**
- 18        Remove *current time events* from *next events list*;
- 19     **end**
- 20     **Call function:** Draw Gantt chart;
- 21     **Call function:** Write results of related PR;
- 22     Evaluate results;
- 23     Continue;
- 24 **end**

---

**Algorithm 2: Initialize matrices**

---

**Data:** not released operations set, not finished operations set, free machines set, released operations set

**Result:** *Updated* not released operations set, not finished operations set, free machines set, released operations set, next events list

```
1 not finished operation set  $\leftarrow$  all operations;
2 foreach machine  $\in$  machines set do
3   | machines waiting list  $\leftarrow \phi$ ;
4 end
5 free machines set  $\leftarrow$  all machines;
6 released operations set  $\leftarrow$  first operations of the first
   released jobs;
7 next events list  $\leftarrow \phi$ ;
8 foreach job  $\in$  jobs set do
9   | next events list  $\leftarrow \{Job_j.R, operation_{j\sigma,r}\}$ ;
10 end
```

---

---

**Algorithm 3: Update operations and machines state according to the new started operations**

---

**Data:** problem set, recently started operations and machines, next event list

**Result:** updated operations and machines, next event list

```
1 foreach operation  $o \in$  operations do
2   | Update operations start time ( $ost_o$ );
3   | Update operations finish time ( $oft_o$ );
4   | Add  $\{operation_o.oft, related\ machine, m\}$  to next
   | events list;
5   | if  $o$  Is Not the last operation of the related job then
6     | Update operations start time ( $ost_{o+1}$ );
7     | Add  $\{operation_{o+1}.ost, operation_{o+1}, r\}$  to
   | next events list;
8   | end
9 end
10 foreach machine  $m \in$  machines do
11   | Update waiting list ( $Mwl_m$ );
12   | Update waiting list workload ( $Mwlw_m$ );
13   | Update last start time ( $Mlst_m$ );
14   | Update last finish time ( $Mlft_m$ );
15   | Remove machine  $m$  from free machines set;
16 end
```

---

---

**Algorithm 4: Find next time and events**

---

**Data:** next events list

**Result:** next time, released operations set, finished operations set, free machines set, machines event set

```
1 released operations set  $\leftarrow \phi$ ;  
2 machines event set  $\leftarrow \phi$ ;  
3 next time  $\leftarrow \min(\text{events list}\{i,1\})$ , for  $i = 1, \dots, N$ ;  
4 next events  $\leftarrow \{\text{events} \in \text{events list} \mid \{i,1\} = \text{next time}\}$ , for  $i = 1, \dots, N$ ;  
5 foreach event  $i$  in next events do  
6   if event $\{i,3\} = r$  then  
7     Add related operation to released operations set;  
8   else if event $\{i,3\} = m$  then  
9     Remove related operation to not finished operations set;  
10    Add related machine to free machines set;  
11    Add related machine to machines event set;  
12  end  
13 end  
14
```

---

---

**Algorithm 6: Update machines state**

---

**Data:** problem set, recently assigned operations and machines events, PR

**Result:** updated machines state, started operation

```
1 if any operations are assigned then  
2   if related machine is free then  
3     Start it;  
4     Update workload ( $Mwlw_m$ );  
5   else  
6     Update machine  $m$  assigned operations ( $Maom$ );  
7     Update machine  $m$  waiting list ( $Mwl_m$ );  
8     Update machine  $m$  waiting list workload ( $Mwlw_m$ );  
9   end  
10 end  
11 if there are any machines in machine event set then  
12   if there are any operations in its waiting list then  
13     Select an operation according to the dispatching rule;  
14     Start it;  
15     Update waiting list ( $Mwl_m$ );  
16     Update waiting list workload ( $Mwlw_m$ );  
17   end  
18 end
```

---

## EK-2

---

**Algorithm 5: Least waiting time assignment**

---

**Data:** problem set, released operations set, ( $t$ )

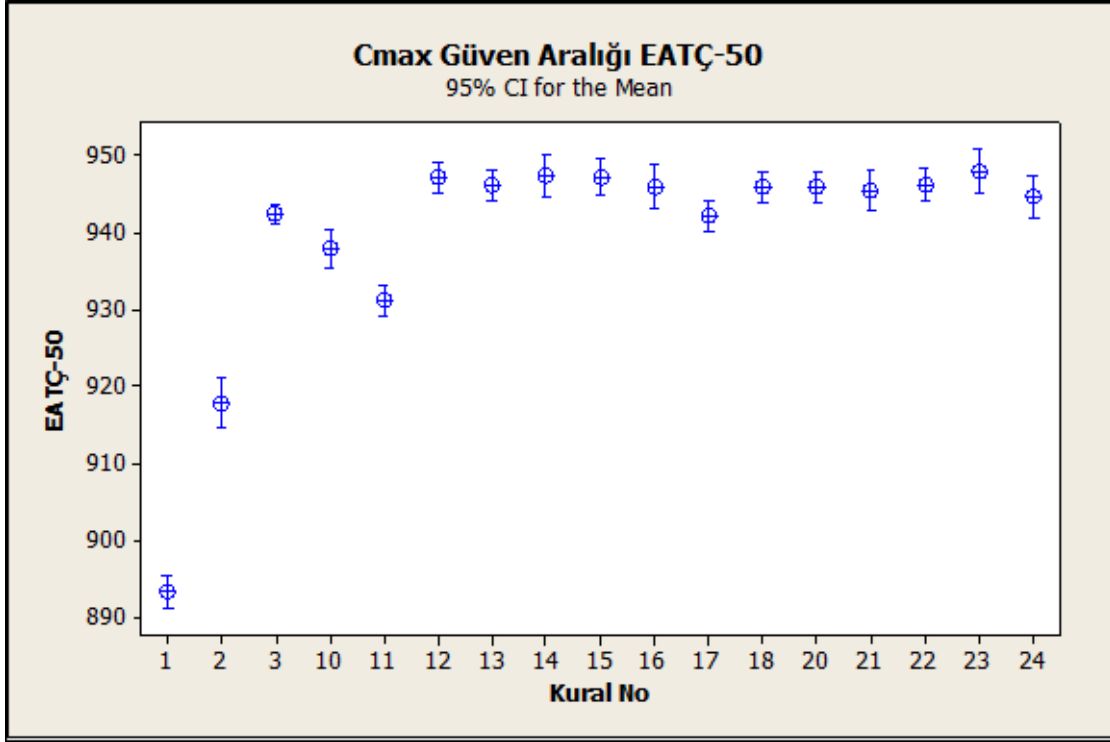
**Result:** Updated  $m_o$ ,  $Mwl_m$ ,  $Mwk_m$

```
1 foreach operation  $o \in$  released operations set do
2   foreach machine  $m \in$  operation  $o$  machines set do
3     if machine  $m$  is busy then
4        $w_{m_1} \leftarrow$  remaining time of machine  $m$ ;
5     else
6        $w_{m_1} \leftarrow 0$ ;
7     end
8      $w_{m_2} \leftarrow$  total workload in machine  $m$  waiting
       list ( $Mwl_m$ );
9      $w_{m_3} \leftarrow P_{om}$ ;
10     $w_m \leftarrow w_{m_1} + w_{m_2} + w_{m_3}$ ;
11  end
12  Select machine with  $\min(w_m)$  and Assign
    operation  $o$  to it;
13  Update  $m_o$ ,  $Mwl_m$ ,  $Mwk_m$ ;
14  Update operation  $o$  assigned machine ( $oam_o$ );
15 end
```

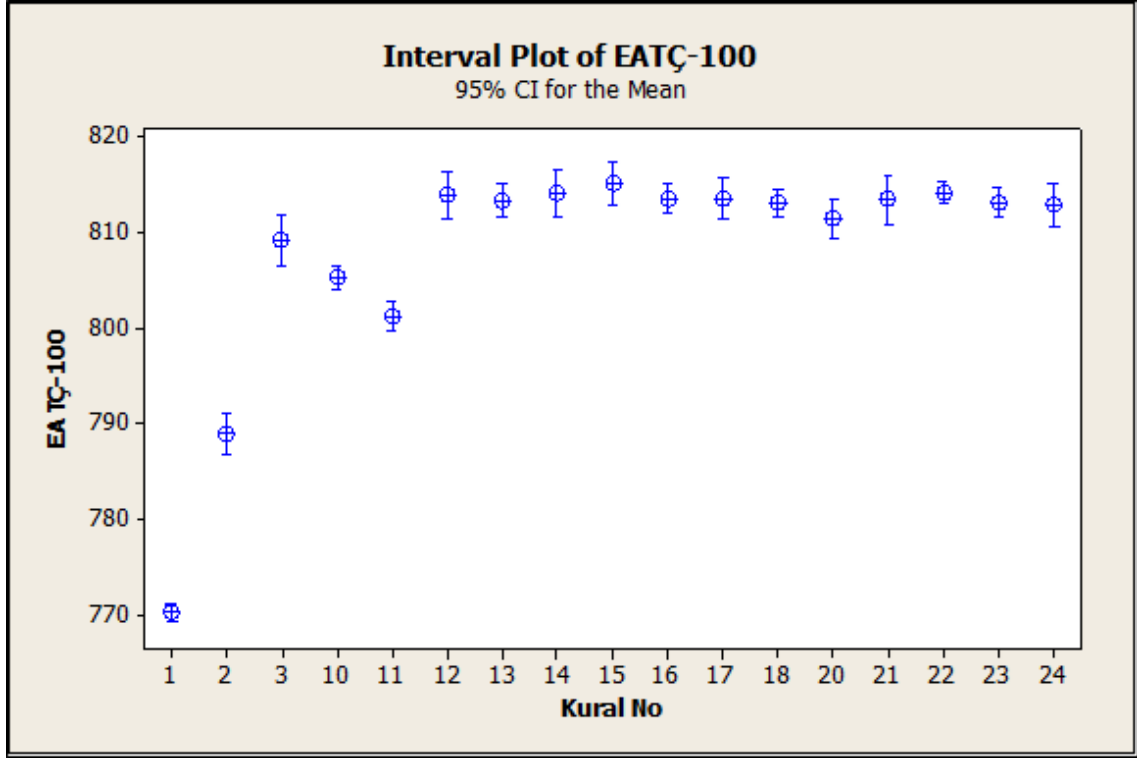
---



### EK-3

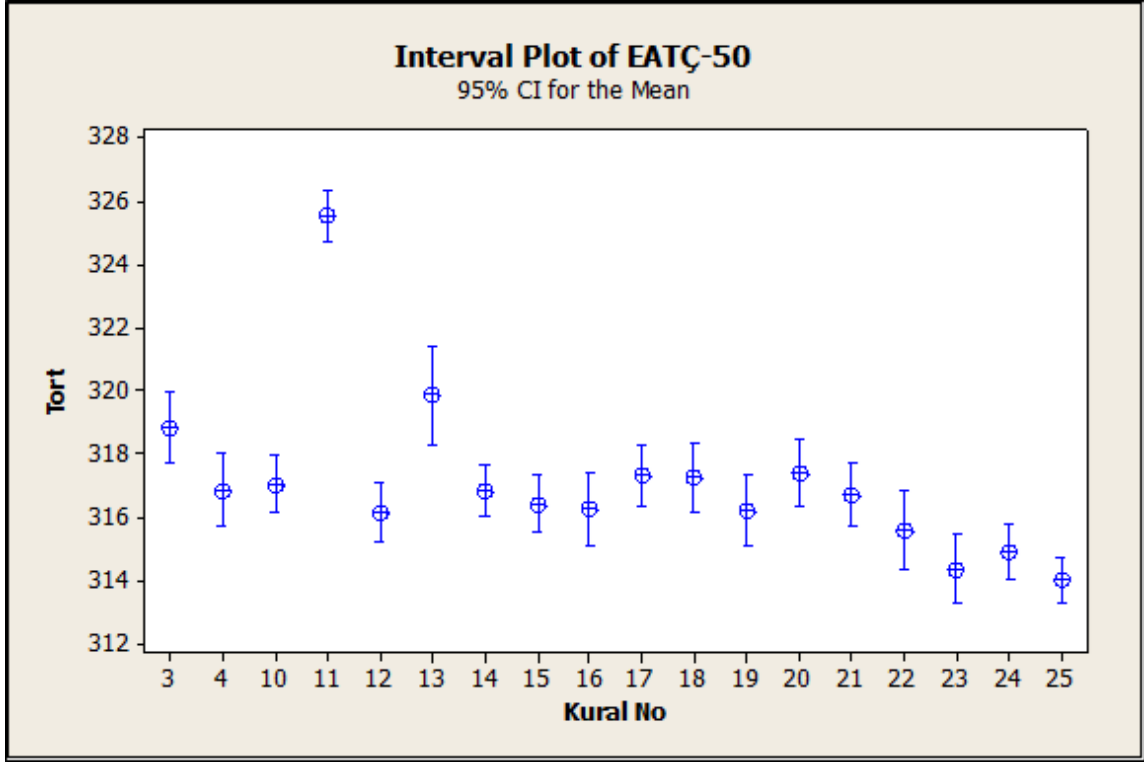


Kural numarası	Örnek sayısı	Ortalama	Grup
1	10	893,309	F
2	10	917,707	E
3	10	942,295	B,C
10	10	937,864	C
11	10	931,154	D
12	10	947,059	A,B
13	10	945,957	A,B
14	10	947,322	A,B
15	10	947,148	A,B
16	10	945,898	A,B
17	10	942,162	B,C
18	10	945,87	A,B
20	10	945,728	A,B
21	10	945,38	A,B
22	10	946,154	A,B
23	10	947,881	A
24	10	944,601	A,B

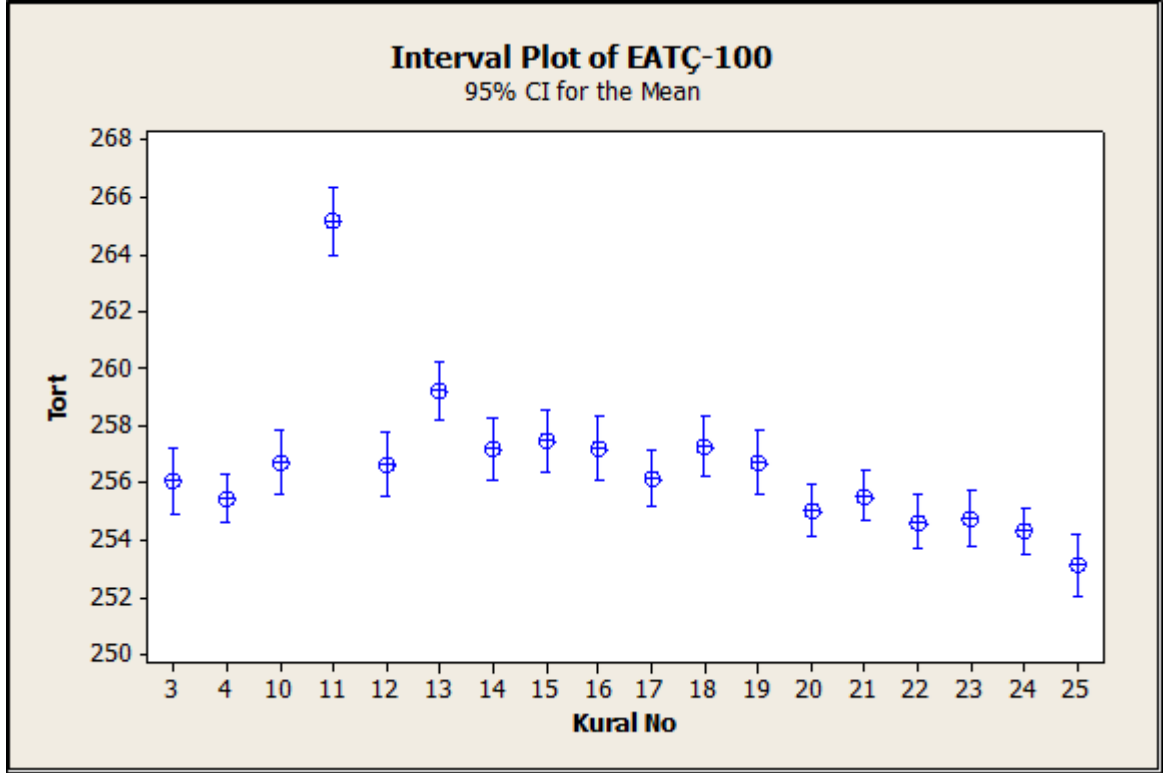


Kural numarası	Örnek sayısı	Ortalama	Grup
1	10	770,273	F
2	10	788,887	E
3	10	809,302	B,C
10	10	805,408	C,D
11	10	801,305	D
12	10	813,954	A
13	10	813,458	A,B
14	10	814,272	A
15	10	815,24	A
16	10	813,682	A
17	10	813,622	A,B
18	10	813,147	A,B
20	10	811,481	A,B
21	10	813,521	A,B
22	10	814,297	A
23	10	813,253	A,B
24	10	812,94	A,B

## EK-4

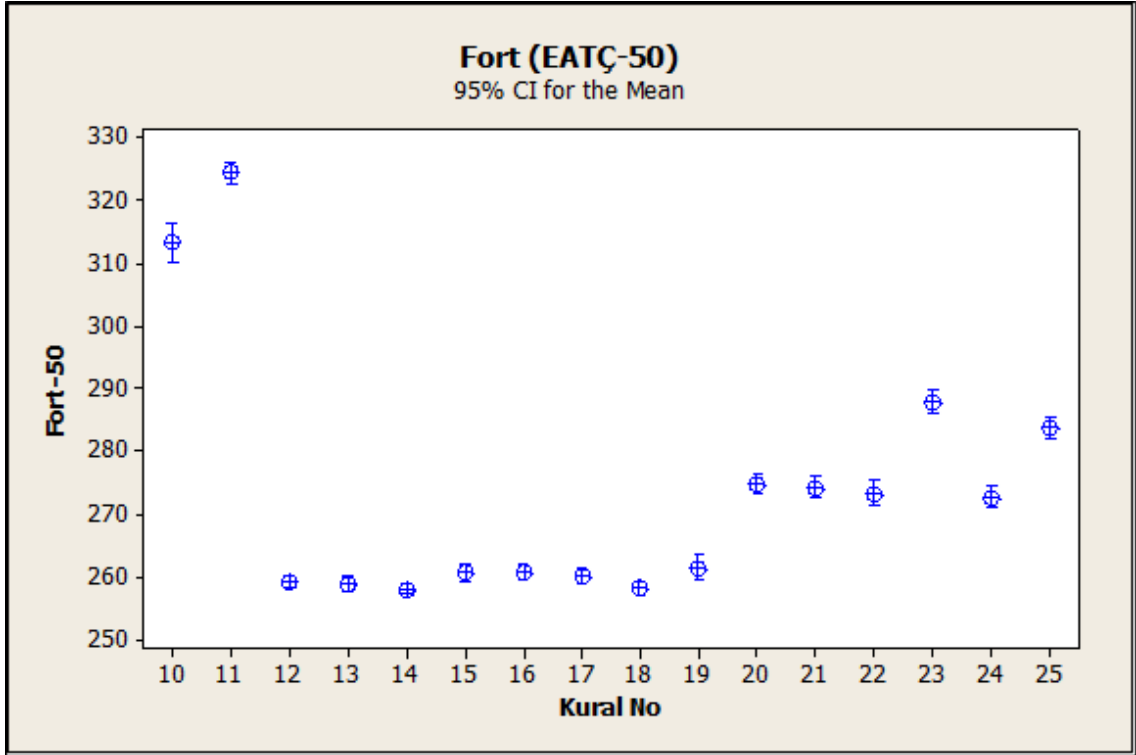


Kural numarası	Örnek sayısı	Ortalama	Grup
3	10	318,811	B,C
4	10	316,856	C,D,E
10	10	317,041	C,D,E
11	10	325,525	A
12	10	316,173	D,E,F,G
13	10	319,845	B
14	10	316,849	C,D,E
15	10	316,423	D,E,F
16	10	316,259	D,E,F,G
17	10	317,31	C,D
18	10	317,237	C,D
19	10	316,211	D,E,F,G
20	10	317,383	C,D
21	10	316,708	C,D,E
22	10	315,587	D,E,F,G
23	10	314,378	F,G
24	10	314,901	E,F
25	10	314,026	G

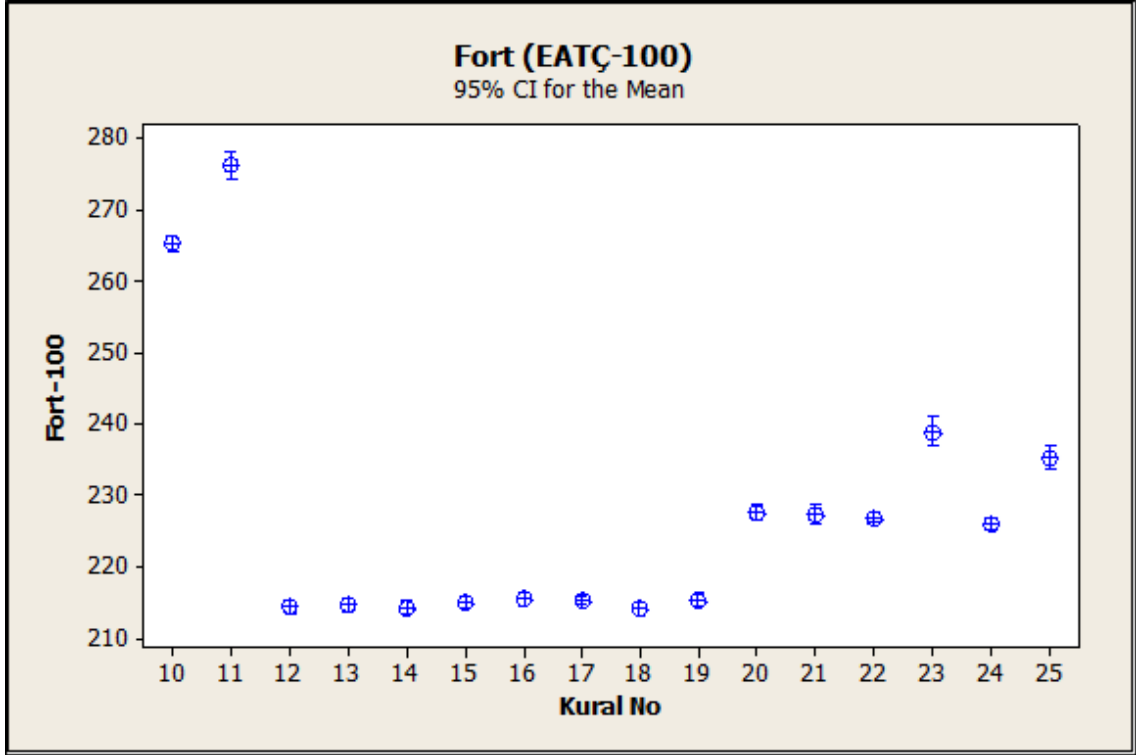


Kural numarası	Örnek sayısı	Ortalama	Grup
3	10	256,039	C,D,E,F
4	10	255,428	C,D,E,F
10	10	256,717	C,D,E
11	10	265,14	A
12	10	256,634	C,D,E
13	10	259,203	B
14	10	257,151	B,C,D
15	10	257,433	B,C
16	10	257,169	B,C,D
17	10	256,155	C,D,E,F
18	10	257,243	B,C,D
19	10	256,701	C,D,E
20	10	255,021	D,E,F,G
21	10	255,52	C,D,E,F
22	10	254,627	E,F,G
23	10	254,719	E,F,G
24	10	254,294	F,G
25	10	253,122	G

## EK-5

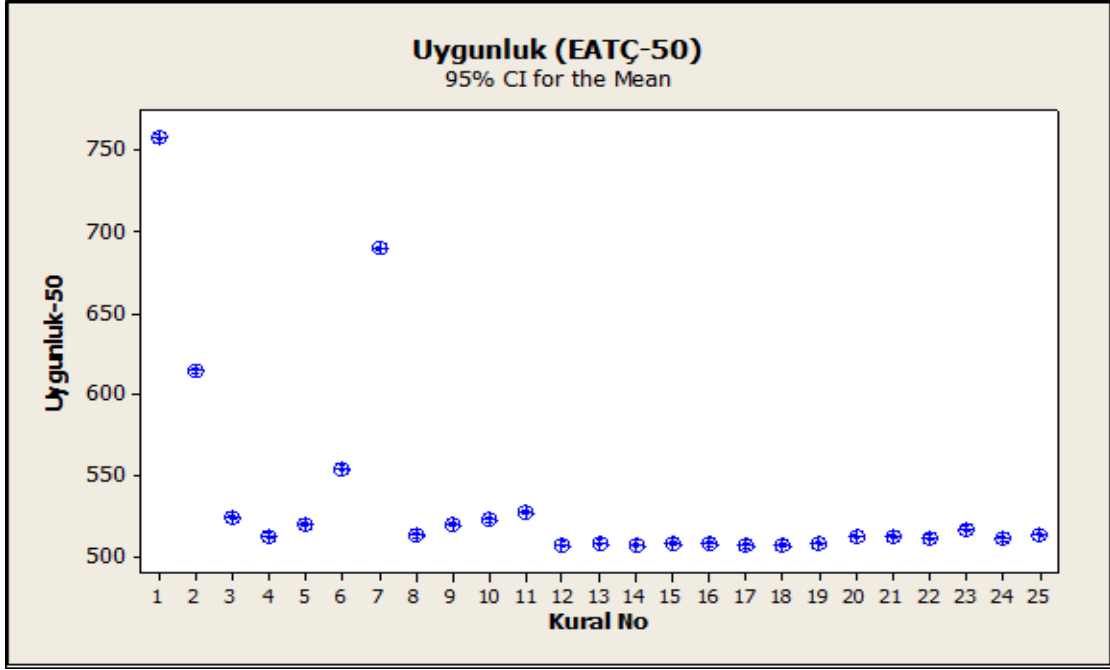


Kural numarası	Örnek sayısı	Ortalama	Grup
11	10	324,444	A
10	10	313,415	B
23	10	287,79	C
25	10	283,722	D
20	10	274,734	E
21	10	274,268	E
22	10	273,313	E
24	10	272,597	E
19	10	261,476	F
16	10	260,745	F
15	10	260,657	F
17	10	260,141	F
12	10	259,279	F
13	10	259,004	F
18	10	258,157	F
14	10	258,076	F

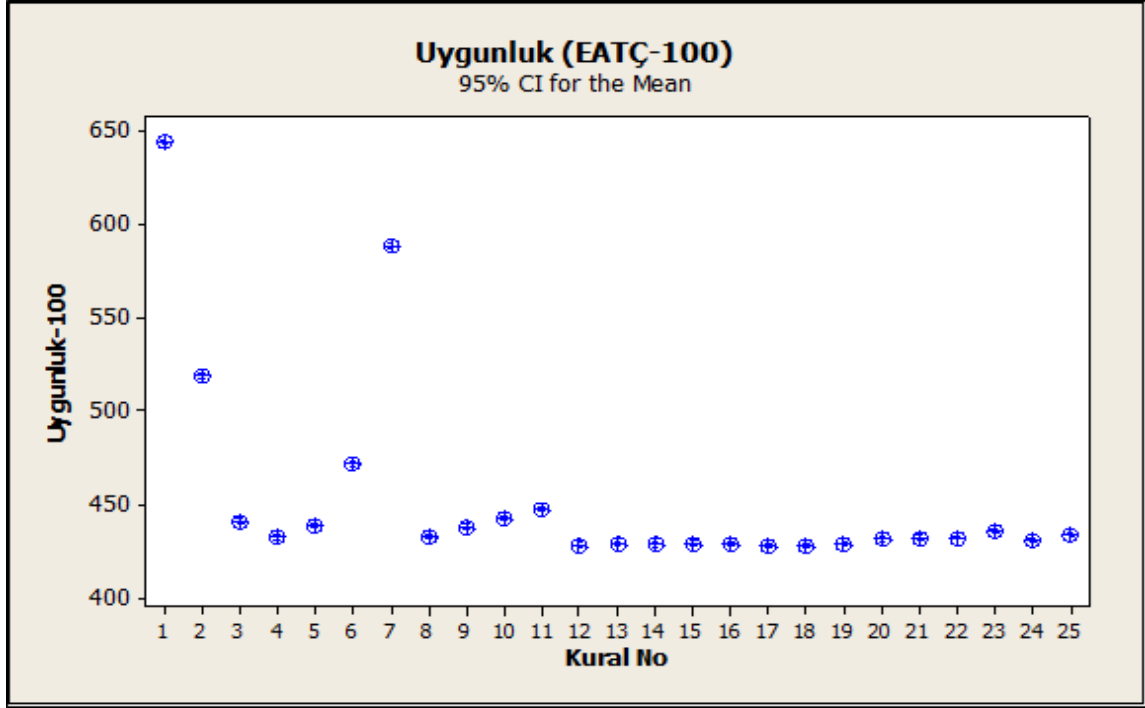


Kural numarası	Örnek sayısı	Ortalama	Grup
11	10	276,077	A
10	10	265,233	B
23	10	238,861	C
25	10	235,23	D
20	10	227,599	E
21	10	227,267	E
22	10	226,851	E
24	10	225,918	E
16	10	215,398	F
19	10	215,386	F
17	10	215,256	F
15	10	214,941	F
13	10	214,676	F
12	10	214,406	F
14	10	214,29	F
18	10	214,079	F

## EK-6



Kural numarası	Örnek sayısı	Ortalama	Grup
1	10	757,68	A
2	10	613,69	C
3	10	523,51	F
4	10	512,47	I
5	10	519,42	G
6	10	554,06	D
7	10	689,7	B
8	10	512,91	I
9	10	519,51	G
10	10	522,77	F
11	10	527,04	E
12	10	507,5	K
13	10	508,27	J,K
14	10	507,42	K
15	10	508,08	J,K
16	10	507,63	J,K
17	10	506,54	K
18	10	507,09	K
19	10	507,9	J,K
20	10	512,62	I
21	10	512,12	I
22	10	511,68	I
23	10	516,68	G,H
24	10	510,7	I,J
25	10	513,65	H,I



Kural numarası	Örnek sayısı	Ortalama	Grup
1	10	643,33	A
2	10	518,39	C
3	10	440,78	F
4	10	432,37	I,J
5	10	438,13	G
6	10	471,53	D
7	10	588,28	B
8	10	432,44	I,J
9	10	437,94	G
10	10	442,45	F
11	10	447,51	E
12	10	428,33	M
13	10	429,11	K,L,M
14	10	428,57	L,M
15	10	429,2	K,L,M
16	10	428,75	K,L,M
17	10	428,34	M
18	10	428,16	M
19	10	428,76	K,L,M
20	10	431,37	I,J,K
21	10	432,1	I,J
22	10	431,92	I,J
23	10	435,61	G,H
24	10	431,05	J,K,L
25	10	433,68	H,I