

**ESNEK ATÖLYE TİPİ ÇİZELGELEME PROBLEMİ İÇİN BİR HİBRİD
GENETİK ALGORİTMA YAKLAŞIMI**

BİLAL METO

**YÜKSEK LİSANS TEZİ
BİLGİSAYAR MÜHENDİSLİĞİ**

**TOBB EKONOMİ VE TEKNOLOJİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

NİSAN 2013

ANKARA

Fen Bilimleri Enstitü Onayı

Prof. Dr. Ünver Kaynak

Müdür

Bu tezin Yüksek Lisans derecesinin tüm gereksinimlerini sağladığını onaylarım.

Doç. Dr. Erdoğan DOĞDU

Anabilim Dalı Başkanı

Bilal METO tarafından hazırlanan ESNEK ATÖLYE TİPİ ÇİZELGELEME PROBLEMİ İÇİN BİR HİBRİD GENETİK ALGORİTMA YAKLAŞIMI adlı bu tezin Yüksek Lisans tezi olarak uygun olduğunu onaylarım.

Doç. Dr. Erdoğan DOĞDU

Tez Danışmanı

Doç. Dr. Kadir ERTOĞRAL

Tez Danışmanı

Tez Jüri Üyeleri

Başkan: Yrd. Doç. Dr. A. Murat ÖZBAYOĞLU

Üye: Doç. Dr. Kadir ERTOĞRAL

Üye: Doç. Dr. Erdoğan DOĞDU

Üye: Doç. Dr. Muhammed Fatih DEMİRCİ

Üye: Yrd. Doç. Dr. Salih TEKİN

TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada orijinal olmayan her türlü kaynağa eksiksiz atıf yapıldığını bildiririm.

BİLAL METO

Üniversitesi: TOBB Ekonomi ve Teknoloji Üniversitesi

Enstitüsü: Fen Bilimleri

Anabilim Dalı: Bilgisayar Mühendisliği

Tez Danışmanı: Doç. Dr. Erdoğan DOĞDU

Doç. Dr. Kadir ERTOĞRAL

Tez Türü ve Tarihi: Yüksek Lisans – Nisan 2013

Bilal METO

**ESNEK ATÖLYE TİPİ ÇİZELGELEME PROBLEMİ İÇİN BİR HİBRİD
GENETİK ALGORİTMA YAKLAŞIMI**

ÖZET

Çizelgeleme problemleri üretim sistemleri için önemli ve klasik problemlerden birisidir. Üretim sistemlerindeki çizelgeleme problemlerinin en genel hallerinden birisi de esnek atölye tipi çizelgeleme problemidir. Esnek atölye tipi çizelgeleme probleminin (EATCP) atölye tipi problemden temel farkı bir operasyonun yapılabileceği alternatif makinelerin ya da üretim merkezlerinin olmasıdır. NP (Non deterministic polynomial)-zor sınıfa giren bu önemli problem için sonuç en iyilenmeye çalışıldığında problemin büyüklüğüne bağlı olarak ya problem çözülememekte ya da çok uzun zamanlara ihtiyaç duyulmaktadır. Çalışmamızda daha hızlı ve iyi sonuçlar alan bir meta-sezgisel geliştirilmiştir. Geliştirilen algoritma, amaç olarak son işin tamamlanma zamanını alan, hibrid bir meta sezgisel algoritmadır. Algoritmamız üç temel aşamadan oluşmaktadır; problem rassal değişim sezgiseli, genetik meta sezgisel ve lokal iyileştirme aşaması. Problem rassal değişim sezgiseli, genetik algoritma için kaliteli ilk nesil çözümlerinin bulunması için kullanılmaktadır. İlk nesil oluşturulurken rassal değişim sezgiseli ile operasyonların işleme zamanları üzerinde eksi ve artı yönde yüzdesel değişiklikler yapılarak seçtiğimiz öncelik bazlı kurallar yardımıyla çizelgeler oluşturulmaktadır. Genetik algoritmayla iyileştirilen çözüm nesilleri son aşamada lokal olarak daha da iyileştirilmeye çalışılmaktadır. Literatürdeki yaklaşımlar ile bizim önerimiz olan bazı yaklaşımları harmanladığımız hibrid sezgiselimiz, literatürdeki test problemleri üzerinde denenmiş ve etkin bir şekilde çalıştığı gözlemlenmiştir.

Anahtar Kelimeler: Genetik algoritma, meta sezgisel, esnek atölye tipi çizelgeleme, lokal arama

University: TOBB University of Economics and Technology

Institute: Institute of Natural and Applied Sciences

Science Programme: Computer Engineering

Supervisor: Doç. Dr. Erdoğan DOĞDU

Doç. Dr. Kadir ERTOĞRAL

Degree Awarded and Date : M.Sc. – April 2013

Bilal METO

**A HYBRID PROBLEM PERTURBATION AND GENETIC ALGORITHM
FOR THE FLEXIBLE JOB-SHOP SCHEDULING PROBLEM**

ABSTRACT

One of the important and classical problems for production systems is the scheduling. Furthermore, one of the most general states of the scheduling problems in production systems is the flexible job-shop scheduling problem (FJSP). The main difference of the flexible job-shop scheduling problems from the regular job-shop scheduling problems is that in the flexible job-shop, we assume that an operation of a job can be performed in several alternative work centers, while in the regular job-shop case an operation can only be performed in a single work center. In order to optimize the result for this important problem, which is considered in NP-hard class, it becomes impossible to solve or long time periods are required for a solution. In this study, we suggest a hybrid meta heuristic for FJSP with the objective of minimizing the makespan. Our algorithm has three stages; a problem perturbation heuristic, a genetic meta algorithm, and a local improvement phase. The perturbation heuristic aims at producing a high quality initial generation of solutions for the genetic algorithm. While producing the initial generation, schedules are formed by means of priority rules by making positive negative changes on processing times of the operations with the perturbation heuristic. The genetic algorithm improves the generations of solutions subsequently and finally a local search phase at the end tries to further improve the solution found in the last generation. Our hybrid meta heuristic which is collated by the approaches we suggested and by the others in the literature is tried on the test problems from the literature and is observed that it works efficiently.

Keywords: Genetic algorithm, meta heuristic, flexible job-shop scheduling problem, local search

TEŐEKKÜR

Çalıőmalarım boyunca bilgisine ve deneyimine daima ihtiyaç duyduėum ve beni hiçbir zaman geri çevirmeyen hocam Doç. Dr. Kadir ERTOĐRAL'a, deėerli yardım ve katkılarıyla beni yönlendiren hocam Doç. Dr. Erdoğan DOĐDU'ya, tüm bu süre zarfında yine kıymetli tecrübelerinden faydalandıėım TOBB Ekonomi ve Teknoloji Üniversitesi Bilgisayar Mühendisliėi ve Endüstri Mühendisliėi Bölümleri öğretim üyelerine, her zaman büyük bir ailenin parçası olduėum hissini veren TOBB Ekonomi ve Teknoloji Üniversitesi camiasına ve manevi desteklerinden ötürü ailem ve özellikle babam Şahabettin METO'ya; ayrıca bana çalıőmalarımı tamamlama sürecinde verdiėi destekten ötürü Serpil KOÇ'a teşekkürlerimi sunarım.

İÇİNDEKİLER

ÖZET.....	iv
ABSTRACT.....	v
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	x
ŞEKİLLERİN LİSTESİ	xii
KISALTMALAR	xiii
1. GİRİŞ.....	1
1.1 Atölye Tipi Çizelgeleme	3
1.2 Esnek Atölye Tipi Çizelgeleme.....	4
1.2.1 Varsayımlar	5
1.2.2 EATÇP'nin Matematiksel Gösterimi.....	5
1.3 Çizelge Türleri.....	8
2. LİTERATÜR	10
3. YÖNTEM VE METOTLARA GENEL BAKIŞ	13
3.1 Çizelgeleme Probleminde Öncelik Kuralları.....	13
3.1.1 İlk gelen İlk Servis Alır.....	13
3.1.2 Son Gelen İlk Servis Alır	13
3.1.3 En Kısa İşlem Süresi	13
3.1.4 En Uzun İşlem Süresi.....	13
3.1.5 En Erken Teslim Zamanı	14
3.1.6 En Az Kalan Operasyon Sayısı	14
3.1.7 En Kısa Kalan İşlem Süresi.....	14
3.1.8 En Uzun Kalan İşlem Süresi	14
3.1.9 En Çok Kalan Operasyon Sayısı	14
3.2 Meta-sezgisel Arama Metodu	15
3.2.1 Keşfetme ve Gezinme (Exploration and Exploitation).....	15
3.2.2 Genetik Algoritma.....	15
3.2.2.1 Kodlama	17
3.2.2.2 Uygunluk fonksiyonu	19
3.2.2.3 Seçim süreci	19

3.2.2.4	Rekombinasyon	22
3.2.2.5	Mutasyon	27
3.2.2.6	Sonlama Kriterleri	28
3.2.2.7	Elitizm-Seçkinlik.....	29
3.3	Rassal Bozma (Perturbation).....	30
3.4	EATÇP’de Kullanılan GA Yaklaşımları	32
3.4.1	Kodlama	32
3.4.1.1	Paralel Makine Kromozom Gösterimi	32
3.4.1.2	Paralel İş Kromozom Gösterimi.....	32
3.4.1.3	Gen vd Kromozom Gösterimi	33
3.4.1.4	Makine Seçim ve Operasyon Seçim Kromozom Gösterimi	33
3.4.2	Seçim Yöntemleri.....	34
3.4.3	Çaprazlama.....	35
3.4.3.1	Geliştirilmiş Sıralı Çaprazlama	35
3.4.3.2	Öncelik Bazlı Sıralı Çaprazlama	35
3.4.4	Mutasyon İşlemleri.....	36
4.	ÖNERİLEN YAKLAŞIMA GENEL BAKIŞ	37
4.1	Kodlama	39
4.2	Başlangıç Popülasyonu.....	40
4.2.1	Rassal Bozma (Perturbation).....	43
4.2.2	Atama Problemi	44
4.2.2.1	Atama Yöntemi – “ Approach by Localization “	44
4.2.2.2	Önerilen Atama Yöntemi – Uzun İşlerin Önce Atanması.....	47
4.2.3	Çizelgeleme Aşaması	50
4.3	Çaprazlama	52
4.4	Mutasyon	53
4.4.1	Rassal Makine Ataması.....	54
4.4.2	Operasyon Yer Değişikliği.....	55
4.4.3	Son Operasyon Yerinin Değiştirilmesi	56
4.4.4	En Büyük Boşluğa Uygun Operasyon Yerleştirme	57
4.4.5	En Fazla Üretim Zaman Kazancı	59

4.4.6	En İyi Makineye Atama	60
4.5	Lokal Arama.....	60
5.	ÖNERİLEN SEZGİSELİN UYGULANMASI VE PERFORMANS TESTİ....	62
6.	SONUÇ.....	69
	KAYNAKLAR	71
	EKLER.....	76
	ÖZGEÇMİŞ	82

ÇİZELGELERİN LİSTESİ

Çizelge 3.1 GA Popülasyonu	16
Çizelge 3.2 İkili Kodlama	18
Çizelge 3.3 Sıralı Kodlama	18
Çizelge 3.4 Değer Kodlama	18
Çizelge 3.5 Popülasyonda Bireylerin Sırlaması ve Uygunluk	20
Çizelge 3.6 Rassal Bozma İşlemi için Söзде Kod	31
Çizelge 3.7 Paralel Makine Kromozom Gösterimi	32
Çizelge 3.8 Paralel İş Kromozom Gösterimi	33
Çizelge 3.9 Gen vd Kromozom Gösterimi	33
Çizelge 3.9 MSOS Kromozom Gösterimi	34
Çizelge 3.10 MSOS Kromozom Makine Kısmı	34
Çizelge 4.1 Kullanılan Kromozom Gösterimi	39
Çizelge 4.2 Aktif Çizelgeye Dönüştürme - Söзде Kod	40
Çizelge 4.3 Üretim Zamanları	43
Çizelge 4.4 Rassal Değişiklik Sonrası Üretim Zamanları	43
Çizelge 4.5 Operasyon İşlenme Zamanları	44
Çizelge 4.6 Operasyon İşlenme Zamanları Rassal Değişiklik Sonrası	45
Çizelge 4.7 Atama İşlemi - 1	45
Çizelge 4.8 Atama İşlemi - 2	46
Çizelge 4.9 Atama İşlemi - 3	46
Çizelge 4.10 Atama İşlemi - 4	46
Çizelge 4.11 Atama İşlemi-5	47
Çizelge 4.12 Makine Atamaları	47
Çizelge 4.13 Atama İşlemi - 1	48
Çizelge 4.14 Atama İşlemi - 2	48
Çizelge 4.15 Atama İşlemi - 3	49
Çizelge 4.16 Atama İşlemi - 4	49
Çizelge 4.17 Atama İşlemi - 5	50
Çizelge 4.18 Makine Atamaları	50
Çizelge 4.19 Operasyon İşlenme Süreleri	53
Çizelge 4.20 Rassal Makine Ataması - Birey ve Operasyon Seçilişi	54
Çizelge 4.21 Rassal Makine Ataması – Seçilen Operasyonu İşleyebilecek Makinelerde İşlenme Süreleri	54
Çizelge 4.22 Rassal Makine Ataması – İşlem Sonrası Kromozom	55
Çizelge 4.23 Operasyon Yer Değişikliği - Birey ve Operasyon Seçilişi	55
Çizelge 4.24 Operasyon Yer Değişikliği – İşlem Sonrası Kromozom	56
Çizelge 4.25 Son Operasyon Yer Değişikliği - Birey ve Operasyon Seçilişi	56

Çizelge 4.26 Son Operasyon Yer Değişikliği - İşlem Sonrası Kromozom.....	57
Çizelge 4.27 En Büyük Boşluğu Operasyon Yerleştirme - Seçilen Birey ve Operasyon	58
Çizelge 4.28 En Büyük Boşluğu Operasyon Yerleştirme - İşlem Sonrası Kromozom	58
Çizelge 4.29 En Fazla Üretim Zaman Kazancı Mutasyonu için Sözde Kod	59
Çizelge 4.30 En İyi Makine Atama Mutasyonu için Sözde Kod	60
Çizelge 4.31 Lokal Arama için Sözde Kod.....	61
Çizelge 5.1 Brandimarte Problem Seti Özellikleri.....	62
Çizelge 5.2 – Senaryo1 ve Senaryo2 ile Literatür Sonuç Karşılaştırma	64
Çizelge 5.3 – Senaryo3 ve Senaryo4 ile Literatür Sonuç Karşılaştırma	65
Çizelge 5.4 – Senaryo5 ve Senaryo6 ile Literatür Sonuç Karşılaştırma	66
Çizelge 5.5 –Tüm Senaryoları Birbiri ile Kıyaslama.....	67
Çizelge A.1 – Senaryo 1 ile Alınan Sonuçlar	76
Çizelge A.2 – Senaryo 1’de Lokal Arama ile Sağlanan İyileşmeler.....	76
Çizelge A.3 – Senaryo 2 ile Alınan Sonuçlar	77
Çizelge A.4 – Senaryo 2’de Lokal Arama ile Sağlanan İyileşmeler.....	77
Çizelge A.5 – Senaryo 3 ile Alınan Sonuçlar	78
Çizelge A.6 – Senaryo 3’de Lokal Arama ile Sağlanan İyileşmeler.....	78
Çizelge A.7 – Senaryo 4 ile Alınan Sonuçlar	79
Çizelge A.8 – Senaryo 4’de Lokal Arama ile Sağlanan İyileşmeler.....	79
Çizelge A.9 – Senaryo 5 ile Alınan Sonuçlar	80
Çizelge A.10 – Senaryo 5’de Lokal Arama ile Sağlanan İyileşmeler.....	80
Çizelge A.11 – Senaryo 6 ile Alınan Sonuçlar	81
Çizelge A.12 – Senaryo 6’da Lokal Arama ile Sağlanan İyileşmeler.....	81

ŞEKİLLERİN LİSTESİ

Şekil 1 .1 Atölye Tipi Makine Ortamı	3
Şekil 1 .2 Esnek Atölye Tipi Makine Ortamı.....	4
Şekil 1.3 Çizelge Türleri arasındaki ilişkiyi gösteren Venn	9
Şekil 1.4 Aktif Çizelgeye Dönüştürme Örnekleri	9
Şekil 3.1 Rulet Bazlı Seçim Örneği	21
Şekil 3.2 Sıralı Seçim Yöntemi Örneği.....	22
Şekil 3.3 Tek Noktalı Çaprazlama Örneği	23
Şekil 3.4 İki Noktalı Çaprazlama Örneği.....	23
Şekil 3.5 Uniform Çaprazlama Örneği.....	24
Şekil 3.6 Sıra Bazlı Uniform Çaprazlama Örneği.....	25
Şekil 3.7 Sıra Bazlı Çaprazlama Örneği	26
Şekil 3.8 Kısmi Eşlemeli Çaprazlama Örneği.....	26
Şekil 3.9 Döngüsel Çaprazlama Örneği	27
Şekil 3.10 Doğal Seçim ve Genetik Operatörler Döngüsü	30
Şekil 3.14 Geliştirilmiş Sıralı Çaprazlama.....	35
Şekil 3.15 Öncelik Bazlı Sıralı Çaprazlama.....	36
Şekil 4.1 Geliştirilen Algoritmanın Akış Diyagramı	38
Şekil 4.2 Başlangıç Popülasyonu Oluşturma Akış Diyagramı	42
Şekil 4.3 Öncelik Bazlı Kuralların Performansı	52
Şekil 4.4 Kullanılan Çaprazlama İşlemi.....	53
Şekil 4.5 Kromozomun Gantt Şemasında Gösterimi	57
Şekil 4.6 İşlem Sonrası Kromozomun Gantt Şemasında Gösterimi	58

KISALTMALAR

ATP	Atölye tipi problem
ATÇP	Atölye Tipi Çizelgeleme Problemi
EATÇP	Esnek Atölye Tipi Çizelgeleme Problemi
EKİS	Esnek Kısa İşlem Süresi
GA	Genetik Algoritma
KSO	Kuş Sürüsü Optimizasyonu
LA	Lokal Arama
MSOS	Makine Seçim ve Operasyon Seçim Yöntemi
NP	Non deterministic polynomial
vd	ve diğerleri

1. GİRİŞ

Üretim ve hizmet sektöründe müşteri memnuniyeti bir firmanın dikkat etmesi gereken en önemli hususlardan biridir. Firmaların vaat ettikleri sürelerde işlerini teslim etmeleri bu memnuniyeti sürdürmek için ön koşuldur. Rekabetin arttığı ve farklı kaynaklara ulaşımın kolaylaştığı günümüz dünyasında kaynaklarını daha verimli kullanan firmalar bu rekabette öne geçecektir.

Kaynakların verimli kullanılması için işler doğru şekilde planlanmalıdır. Üretim sistemlerinde planlama, alınan siparişlerin doğru zamanda ve doğru kaynakla işlenmesini gerektirmektedir. Bir üretim sisteminde, hammaddeye erişim zamanından, makinelerin doluluk oranlarına ve iş gücünün durumuna kadar planlamayı etkileyen birçok faktör bulunmaktadır. Bu faktörlerin tamamı çizelgeleme işleminde ele alınmalıdır.

Çizelgeleme, kaynakların ve işlerin farklı şekiller aldığı üretim ve hizmet sektörlerinde düzenli olarak kullanılan bir karar verme sürecidir. Kaynaklar atölyedeki makineler, havaalanındaki pistler, inşaat sahasındaki işçiler veya bir hesaplama işleminde kullanılan bilgisayarlar olabilir. İşler ise bir üretim sürecindeki operasyonlar, havaalanındaki kalkış ve inişler ve bir inşaat projesindeki aşamalar olabileceği gibi bilgisayar programlarının çalıştırılması da olabilir. Her operasyonun belli öncelik sırası, en erken başlama zamanı ve bitiş tarihi olabilmektedir. [1]

Üretim çizelgeleme alanındaki çalışmalar 20. yüzyılın ilk dönemlerinde başlamış ve çizelgeleme sürecini en fazla etkileyen üç isim ise Frederick Taylor, Henry Gantt ve S.M. Johnson olmuştur. Taylor ilk defa planlama birimini üretimden ayırmış ve çizelgelemeyi ayrı bir karar verme süreci olarak görmüştür. Gantt ise ünlü Gantt şemasını oluşturarak üretim çizelgeleme yazılımının grafiksel arabiriminin temellerini atmıştır. Her ikisi de 1930lar Amerika'sının büyük çelik işletmelerinde danışman olarak çalışmışlardır. Johnson ise matematiksel analizlerle çizelgeleme işlemini başlatmıştır. [2]

Thomas Morton ve David W. Pentico'nun "Heuristic Scheduling Systems" kitabında anlatıldığı üzere, çizelgeleme alanında 1950'lerde geniş araştırmalar yapılmış

olmasına rağmen 1970'lere kadar popüler bir konu olmaktan uzak kalmıştır. Bu yıllardan itibaren ise başta Japonlar olmak üzere geniş çapta kullanılmaya başlanmış, ancak bu dönemlerde esasen fabrika üretim verimliliğini sağlamak üzere Kanban gibi bilgisayar tabanlı olmayan teknikler kullanılmıştır. [3]

Çizelgeleme ile makinelerin uygunluğu, çalışan devamsızlığı ve malzemelerin zamana göre uygunluğunu gözlemek ve raporlamak mümkün olabilmektedir. Bu tip raporların analiz edilmesi ile yönetim kademesi, işlemlerin üretkenliğini azaltan faktörleri belirleyebilmektedir. Örneğin, çizelgelenmemiş makina yatma süreleri yetersiz bakım çalışmalarına, yüksek işçi devamsızlığı çalışan memnuniyetsizliğine ve malzemelerin uygunsuzluğu başarısız satın alma yönetimine işaret edebilmektedir.

Çizelgeleme üretimde 20.yüzyılın başlarından itibaren kullanılmasına rağmen literatürde yer alması 1950'lerin ilk zamanlarına rastlamaktadır. 1960'larda önemli miktarda dinamik ve tam sayı programlama kullanılarak çalışmalar yapılmıştır [1] ve sonrasında akademisyenlerin oldukça ilgisini çekmiştir.

Bu çalışmada üretim sistemlerinden biri olan esnek atölye tipi çizelgeleme problemleri(EATÇP) için en son biten işin tamamlanma zamanını en küçüklemeyi amaçlayan meta-sezgisel hibrid bir algoritma geliştirilmiştir. Algoritma, Genetik Algoritma(GA) üzerine kurulmuştur. GA'nın performansını etkileyen faktörlerden birisi başlangıç popülasyonunun çeşitliliği ve kalitesidir. Yapılan çalışmada iyi bir başlangıç popülasyonu oluşturulmaya çalışılmış ve bunun için yeni bir yöntem önerilmiştir. Başlangıç popülasyonunun ardından çaprazlama ve mutasyon işlemleri gerçekleştirilmiş ve son popülasyon üzerinde lokal arama yapılarak amaç fonksiyonu iyileştirilmeye çalışılmıştır.

İlk bölümde Esnek Atölye Tipi Çizelgeleme Problemi'nden ve takip eden ikinci bölümde literatürde yapılmış çalışmalardan bahsedilmektedir. Üçüncü bölümde Genetik Algoritmanın genel yapısı ve EATÇP için kullanılan öncelik bazlı kurallar anlatılmaktadır. Dördüncü bölüm yapmış olduğumuz çalışmayı detaylı olarak

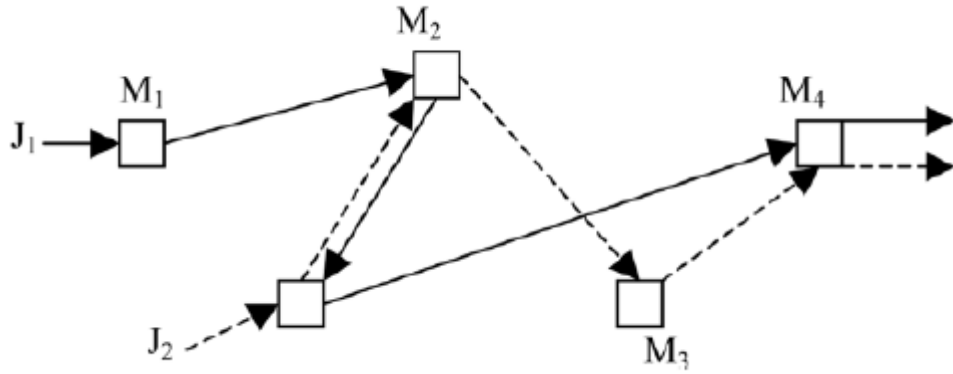
anlatmakta ve beşinci bölümde önerilen metodun uygulaması ve sonuç kısmı bulunmaktadır.

1.1 Atölye Tipi Çizelgeleme

Çizelgeleme birbiri ile yarışan aktivitelerin, zaman içinde ortak kaynaklara dağıtımıdır. Makine çizelgeleme problemi için işler aktiviteleri temsil ederken ortak kaynaklar makinelerdir.

Atölye Tipi Çizelgeleme Problemleri (ATÇP) “ n ” tane iş ve “ m ” adet makineden oluşmaktadır. Her işin teknolojik bakımdan takip etmesi gereken belirli bir makine sıralaması vardır. J_i işinin M_h makinesinde işlenmesi O_{jh} olarak gösterilmekte ve tamamlanma zamanı C_{jh} olarak adlandırılmaktadır.

Herhangi bir ATÇP'nin arama uzayında $(n!)^m$ olası çözüm bulunmaktadır. Garey, Johnson ve Sethi[4] 1976 yılında ATÇP'nin NP-zor sınıfa girdiğini göstermişlerdir. NP-zor bugüne kadar polinom zamanda çözüm bulunamamış ve bunun olamayacağı ispatlanamamış problemler için kullanılmaktadır. ATÇP bu sınıfın en fazla hesaplama zamanı isteyen problemleri arasındadır. Bunun bir göstergesi olarak Muth ve Thompson [5] tarafından oluşturulan 10 makine ve 10 işlik bir problem 20 yıl boyunca çözülememiştir [6].



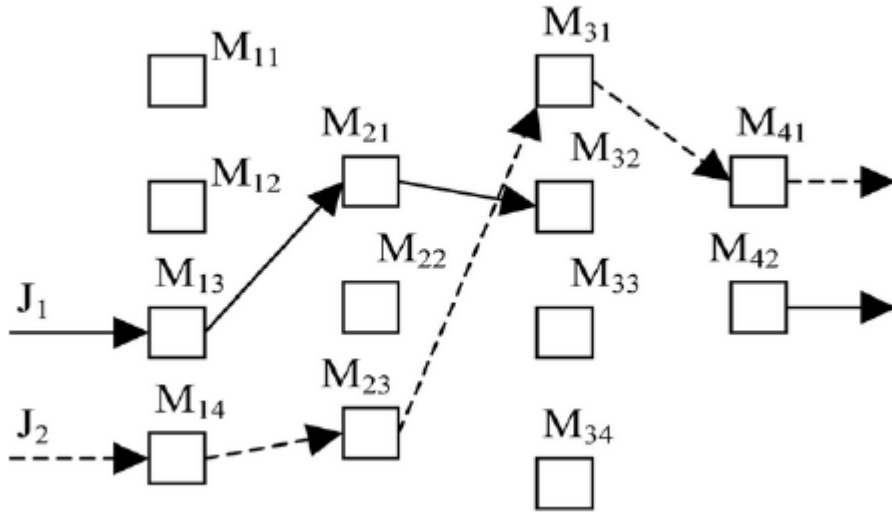
Şekil 1 .1 Atölye Tipi Makine Ortamı [7]

1.2 Esnek Atölye Tipi Çizelgeleme

Esnek atölye tipi çizelgeleme problemi, atölye tipi problemlerinin genişletilmiş halidir. İşlere ait operasyonlar artık bir makinede değil birden fazla makine tarafından işlenebilmektedir. Makinelerin teknolojilerine göre işleme süreleri farklılık gösterebilmektedir. Operasyonları işleyebilen makine sayısına göre iki alt tür problem mevcuttur.

1. “Tam Esnek Atölye Tipi” üretim sistemlerinde tüm operasyonlar sistemde bulunan bütün makineler(M) tarafından işlenebilmektedir.
2. “Kısmi Esnek Atölye Tipi” üretim olarak adlandırılmakta ve her operasyonu işleyebilen bir $M_{j,h} \subset M$ makine kümesi bulunmaktadır.

Operasyonların makinelere atanma problemini de içeren EATÇP, ATÇP’ye göre daha zor bir probleme dönmektedir. Bu sebeple EATÇP de NP-zor problem kümesine girmektedir. [8]



Şekil 1 .2 Esnek Atölye Tipi Makine Ortamı [7]

1.2.1 Varsayımlar

Problem aşağıdaki varsayımlar kabul edilerek çözülmüştür.

- Tüm makineler 0 zamanında boş durumdadır.
- Tüm işler 0 zamanında hazırdır.
- Bir makine aynı andan birden fazla operasyonu işleyemez.
- Her operasyon kesintiye uğramaksızın tamamlanmak zorundadır.
- Her işe ait operasyonların sırası belirlidir ve değiştirilemez.
- Makinenin hazırlık zamanı yoktur ve makineler arası taşıma süresi ihmal edilir.

1.2.2 EATÇP'nin Matematiksel Gösterimi

Aşağıda formülize edilen EATÇ probleminde n iş ve m adet makine bulunmaktadır. Her bir işe ait operasyonlar O_{jh} olarak gösterilmekte ve j işinin h 'inci operasyonu olarak adlandırılmaktadır. Atölyedeki makine kümesi $M = \{M_1, M_2, \dots, M_m\}$ olarak gösterilmektedir. Her operasyon her makinede yapılamadığı için her iş-operasyon işleyebilen bir makine seti bulunmakta ve $M_{j,h} \subset M$ olarak gösterilmektedir. Gösterimlerin tamamı Fattahi [9]'den alınmıştır.

Parametreler

- n : Toplam iş sayısı
 m : Toplam makine sayısı
 $a_{i,j,h}$ $O_{j,h}$ operasyonunu işleyebilen $M_{j,h}$ makine kümesini

$$a_{i,j,h} = \begin{cases} 1 & \text{Eğer } O_{j,h} \text{ operasyonunu makine } i \text{ ile işlenebiliyorsa} \\ 0 & \text{Diğer durumda} \end{cases}$$

- L Oldukça büyük bir sayı
 $P_{i,j,h}$ $O_{j,h}$ operasyonunu makine i 'de işleme süresi

Karar Değişkenleri

C_{max} : En son biten operasyonun tamamlanma zamanı

$$y_{i,j,h} = \begin{cases} 1 & \text{Eğer } O_{j,h} \text{ operasyonunu işlemesi için makine } i \text{ seçilmişse} \\ 0 & \text{Diğer durumda} \end{cases}$$

$$X_{i,j,h,k} = \begin{cases} 1 & \text{Eğer } O_{j,h} \text{ operasyonunu } k.\text{sırada işlemesi için makine } i \text{ seçilmişse} \\ 0 & \text{Diğer durumda} \end{cases}$$

$t_{j,h}$ $O_{j,h}$ operasyonunun işlenmeye başlama zamanı

$T_{m_i,k}$ Makine i 'nin k . sıradaki operasyonu işlemeye başlama zamanı

k_i Makine i 'ye atanmış operasyon sayısı

Aşağıda verilen modelde;

Kısıt(1) en son biten işi belirlemektedir. Kısıt(2) operasyon, makinede işlenmek üzere seçilmişse işleme zamanının hesaplamalara katılmasını sağlamaktadır. Kısıt(3) her işe ait operasyonların aralarındaki öncelik sırasına uymalarını sağlamaktadır. Kısıt(4) her makinenin aynı anda birden fazla operasyonu işlemesini engellemektedir. Kısıt(5-6) her operasyon için operasyon işlenmeye başlanacaksa makinenin boş beklemesini engellemektedir. Kısıt(7) operasyonun, işlenebilecek makineler tarafından işlenmesini sağlarken Kısıt(8) operasyonun makineye atanmasını ve işlerin makine üzerindeki sırasını belirlemektedir. Kısıt(9-10) ise her operasyonun tek makinede ve tek bir sırada işlenmesini sağlamaktadır.

Minimize C_{\max}

s.t.

$$C_{\max} \geq t_{j,h_j} + Ps_{j,h_j} \quad \text{for } j = 1, \dots, n; \quad (1)$$

$$\sum_i y_{i,j,h} \cdot p_{i,j,h} = Ps_{j,h} \quad \text{for } j = 1, \dots, n; \\ h = 1, \dots, h_j; \quad (2)$$

$$t_{j,h} + Ps_{j,h} \leq t_{j,h+1} \quad \text{for} \\ j = 1, \dots, n; h = 1, \dots, h_j - 1; \quad (3)$$

$$Tm_{i,k} + Ps_{j,h} \cdot x_{i,j,h,k} \leq Tm_{i,k+1} \quad \text{for } i = 1, \dots, m; \\ j = 1, \dots, n; h = 1, \dots, h_j; k = 1, \dots, k_i - 1; \quad (4)$$

$$Tm_{i,k} \leq t_{j,h} + (1 - x_{i,j,h,k}) \cdot L \quad \text{for } i = 1, \dots, m; \\ j = 1, \dots, n; h = 1, \dots, h_j; k = 1, \dots, k_i; \quad (5)$$

$$Tm_{i,k} + (1 - x_{i,j,h,k}) \cdot L \geq t_{j,h} \quad \text{for } i = 1, \dots, m; \\ j = 1, \dots, n; h = 1, \dots, h_j; k = 1, \dots, k_i; \quad (6)$$

$$y_{i,j,h} \leq a_{i,j,h} \quad \text{for } i = 1, \dots, m; j = 1, \dots, n; \\ h = 1, \dots, h_j; \quad (7)$$

$$\sum_j \sum_h x_{i,j,h,k} = 1 \quad \text{for } i = 1, \dots, m; k = 1, \dots, k_i; \quad (8)$$

$$\sum_i y_{i,j,h} = 1 \quad \text{for } j = 1, \dots, n; h = 1, \dots, h_j; \quad (9)$$

$$\sum_k x_{i,j,h,k} = y_{i,j,h} \quad \text{for } i = 1, \dots, m; \\ j = 1, \dots, n; h = 1, \dots, h_j; \quad (10)$$

$$t_{j,h} \geq 0 \quad \text{for } j = 1, \dots, n; h = 1, \dots, h_j; \quad (11)$$

$$Ps_{j,h} \geq 0 \quad \text{for } j = 1, \dots, n; h = 1, \dots, h_j; \quad (12)$$

$$Tm_{i,k} \geq 0 \quad \text{for } i = 1, \dots, m; k = 1, \dots, k_i; \quad (13)$$

$$x_{i,j,h,k} \in \{0, 1\} \quad \text{for } i = 1, \dots, m; \\ j = 1, \dots, n; h = 1, \dots, h_j; k = 1, \dots, k_i; \quad (14)$$

$$y_{i,j,h} \in \{0, 1\} \quad \text{for } i = 1, \dots, m; j = 1, \dots, n; \\ h = 1, \dots, h_j; \quad (15)$$

1.3 Çizelge Türleri

Bir çizelge üretilmek istediğinde ve alternatifler değerlendirildiğinde teorik olarak sonsuz sayıda çizelgenin oluşturulabilmesi mümkündür. Bu, iki operasyon arasında istenildiği kadar boş zaman eklenebilmesinden kaynaklanmaktadır. Anlamsız beklemleri engellemek için operasyonların mümkün olduğunca sola kaydırılması gerekmektedir.

Herhangi bir operasyon, çizelge üzerinde bulunan operasyonların yerini değiştirmeden mevcut durumundan daha erken başlayabiliyorsa buna “lokal sola-kaydırma” adı verilmektedir. [10]

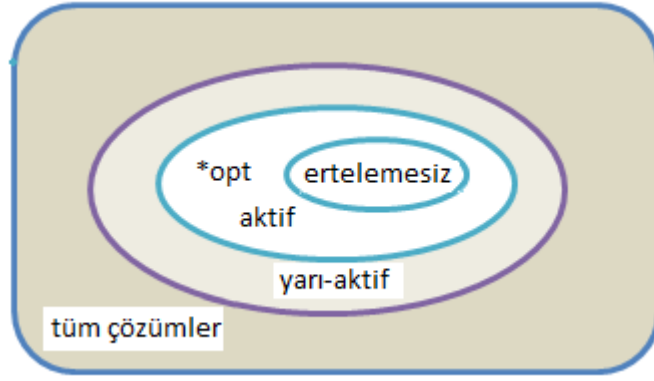
Eğer bir operasyon, operasyonların sıralamasında değişiklik yapılarak ancak başlama zamanlarında herhangi bir ertelemeye sebep olmadan mevcut durumundan daha erken başlayabiliyorsa buna “global sola-kaydırma” adı verilmektedir. [10]

Yarı-aktif çizelge: Çizelge, üzerinde bulunan operasyonların hiçbirisi için lokal sola-kaydırma imkanı bulunmuyorsa yarı aktif olarak adlandırılır.

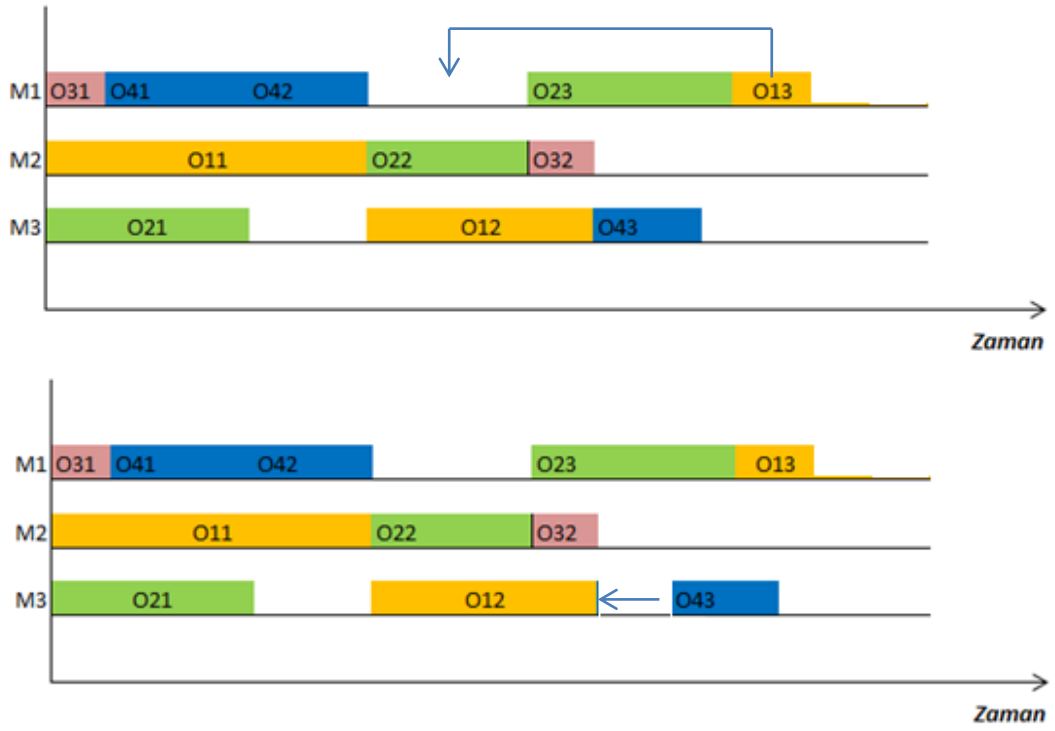
Aktif Çizelge: Yarı aktif hale getirilmiş çizelgede hiçbir operasyonu global sola-kaydırma imkanı bulunmuyorsa bu çizelgelere aktif çizelge denilmektedir. (Bkz şekil 1.4)

Ertelemesiz Çizelge: herhangi bir makine bir işi işleyebilecekken boş durmuyorsa bu çizelge ertelemesiz çizelge olarak adlandırılmaktadır.

Aşağıdaki Venn şemasında evrensel küme, olası tüm çözümleri göstermektedir. Yarı aktif çizelgeler evrensel kümenin bir alt kümesidir. Aktif çizelgeler ise yarı aktif çizelge kümesinin ve ertelemesiz çizelgeler aktif çizelge kümesinin bir alt kümesidir. Son işin tamamlanma zamanını en küçüklemeyi amaç edinen problemler için optimal çözümler aktif çizelgeler arasından çıkmaktadır.



Şekil 1.3 Çizelge Türleri arasındaki ilişkiyi gösteren Venn



Şekil 1.4 Aktif Çizelgeye Dönüştürme Örnekleri

2. LİTERATÜR

Brandimarte(1993) ilk kez hiyerarşik yaklaşımla EATÇP problemini en son biten işin tamamlanma zamanını en küçüklemeye çalışarak bir meta-sezgisel olan tabu arama yöntemi ile çözmüştür. Hiyerarşik yaklaşımda öncelikli olarak operasyonların makinelere ataması yapılırken, daha sonraki aşamada sıralama işlemi yapılmaktadır.[12]

Kacem vd(2002), Brandimarteda[12] olduğu gibi EATÇP'yi hiyerarşik yaklaşımla çözmüşlerdir. Problemi atama ve sıralama problemi olarak ikiye ayırmışlar ve sıralama işleminden önce makinelere atanan toplam işleme süresini dengelemek üzere, "Approach by Localization" adını verdikleri bir atama algoritması geliştirmişlerdir. Bu yöntem birçok çalışmada atama işleminin temeli olarak kullanılmıştır.[13]

Ho ve Tay(2004) GENACE adını verdikleri bir GA geliştirmişlerdir. En son biten işin tamamlanma zamanını esas almış ve başlangıç popülasyonu oluştururken kompozit öncelik bazlı kurallar kullanmışlardır. GA operasyonları sonrası popülasyona katılacak bireyler seçilirken belirli aralıklarla seçilmiş en iyi kromozomlar ile bir sonraki nesle taşınacak bireyler arasında yakınlık karşılaştırması gerçekleştirmekte ve belirli benzerlikteki bireyler bir sonraki nesle alınmaktadır.[14]

Ho vd(2007) daha önceki çalışmalarında oluşturdukları GENACE[14] adlı metoda uyguladıkları yaklaşım ile meta-sezgisel olarak genetik algoritmanın kullanıldığı EATÇP için öğrenme bazlı bir mimari önermişlerdir.[15]

Gao vd(2007) EATÇP için hibrid bir yöntem önermişlerdir. Genetik algoritma ile birlikte lokal arama yöntemi kullanmışlardır. Çizge teorisindeki kritik patika üzerinde, darboğaz öteleme yöntemini kullanarak amaç fonksiyonunda en son tamamlanan operasyonun bitiş zamanını, bir makineye düşen maksimum iş yükünü ve toplam iş yükünü aynı anda en küçüklemeye çalışmışlardır. Geliştirdikleri yöntemle oldukça iyi sonuçlar almışlardır.[8]

Gao vd(2008) yine genetik algoritmayla birlikte kritik patika üstünde bulunan herhangi bir operasyonun yerini değiştirerek üç amacı birden minimize etmeye çalışan bir hibrid algoritma geliştirmişlerdir. EATÇP’de aynı işe ait farklı operasyonlar arasında öncelik kuralı bulunduğundan bir işin en erken başlayabileceği ve en geç biteceği zamanlar belirlidir. Geliştirdikleri yöntem ile kritik patika üzerinden seçilen operasyon, bu operasyonun işlenebileceği zaman aralığında, birbirini takip eden iki operasyon arasında maksimum bekleme zamanına sahip uygun makineye yerleştirilmektedir.[16]

Pezella vd(2008) EATÇP’yi genetik algoritma yöntemi çözmüşlerdir. Literatürde kullanılmış farklı stratejileri birleştirerek en son biten işin tamamlanma zamanı üzerinde iyi sonuçlar elde edilmiştir. Arama işlemi yapılırken yüksek popülasyon ve jenerasyon sayısı kullanılmıştır. [17]

Zhang(2010) EATÇP problemi için genetik algoritma ile tabu arama yöntemini birlikte kullanmışlardır. Genetik algoritma geliştirilen yöntemin ana çatısını oluştururken, belli bir oranla genetik operasyonlar yapmak yerine tabu arama yöntemi ile lokal aramalar gerçekleştirmişlerdir.[18]

Tang vd(2011) yayınladıkları makalede en son biten operasyonun tamamlama zamanını en küçükleme için genetik algoritma ile birlikte yine bir popülasyon tabanlı arama algoritması olan Kuş Sürüsü meta-sezgiselini kullanmışlardır.[19]

Zhang vd(2008) EATÇP için bir genetik algoritma geliştirmişlerdir. Başlangıç popülasyonu üç farklı metot ile oluşturulmuş ve bunun neticesinde iyi sonuçlar elde etmişlerdir. Bu üç yaklaşım başlangıç popülasyonunda yüzdesel olarak farklılık göstermektedir. Kendi geliştirdikleri lokal ve global seçim adını verdikleri yöntem sırasıyla popülasyonun %30 ve %60’ını oluştururken, geri kalan %10 rassal olarak oluşturulmuştur.[20]

Al-Hinai ve ElMekkawy(2011) EATÇP’de en son operasyonunun bitme zamanını en küçükleme için hibrid genetik bir algoritma geliştirmişlerdir. Lokal arama yöntemini geliştirdikleri algoritmaya ekleyerek GA’nın performansını arttırmışlardır. Belli sayıda GA jenerasyonu oluşturulmuş ve lokal arama yapmışlar ve aramaya sokulan

bireyde yapılan deęişiklik sonucunda belirledikleri kriter sayısınca iyileşme görölmedięi durumda lokal aramayı sonlandırmışlardır.[21]

Zhang vd(2011) Genetik Algoritma kullanarak EATÇP’de en son biten işin tamamlanma zamanını en küçüklemeye çalışmışlardır. Global Seçim ve Lokal Seçim adını verdikleri yöntemle iyi bir başlangıç popölasyonu oluşturmuşlar ve uyguladıkları GA operasyonları ile iyi sonuçlar elde etmişlerdir.[22]

Girish ve Jawahar(2009) kuş sürüsü optimizasyon yöntemi ile EATÇP’de en son biten operasyonun tamamlama zamanını en küçüklemek için bir algoritma geliştirmişlerdir. Bu algoritması iki bölümden oluşmuştur. İlk bölümde makine atama işlemini yaparak problemi ATÇP’ne çevirmiş ve ikinci bölümde Giffler and Thompson yöntemi ile çizelgeler oluşturmuşlardır.[23]

Xia ve Wu(2005), kuş sürüsü optimizasyon (KSO) yöntemi ile birlikte Tavlama Yöntemini çok amaçlı bir fonksiyon için bir arada kullanmışlardır. KSO ile oluşan sürüdeki her bir birey için tavlama yöntemi ile belli komşuluklarda arama gerçekleştirmişlerdir.[24]

3. YÖNTEM VE METOTLARA GENEL BAKIŞ

3.1 Çizelgeleme Probleminde Öncelik Kuralları

İşlerin sıralamasında aşağıdaki öncelik kuralları genel olarak kullanılan kurallardır.[25]

3.1.1 İlk gelen İlk Servis Alır

Çizelgelemede ilk olarak gelen emirlere ait iş, ilk olarak sisteme alınır. Ardından gelen işler yine geldikleri sırada çizelgede sıralanır.

3.1.2 Son Gelen İlk Servis Alır.

Çizelgelemede son olarak gelen iş, ilk olarak sisteme alınır. Ardından gelen işler yine geldikleri sıranın tersi şekliyle çizelgeye dahil edilirler.

3.1.3 En Kısa İşlem Süresi

En Kısa İşlem Süresi (EKİS) toplam akış süresini minimize etmeye çalışan öncelik bazlı bir atama kuralıdır. Tüm işler arasından operasyon süresi en kısa olandan en uzun olana göre işler sıralanarak çizelgeye dahil edilir. Eğer işler arasında birbirine bağımlılık söz konusu ise atanabilecek mümkün çözümler ile çizelge oluşturulmaya başlanır ve her atama işleminden sonra mümkün iş kümesi güncellenerek çizelgelemeye devam edilir.

3.1.4 En Uzun İşlem Süresi

Paralel makinelerin bulunduğu ortamda makine yükünü dengelemek için kullanılır. Uzun süreli işler daha önce planlanırken sona kısa işlerin kalması sağlanır, bu sayede çizelgeye son işler eklendiğinde makinaların iş yüklerinde büyük bir fark oluşmaması sağlanır. EKİS benzeri çizelgeleme işlemi yapılmaktadır ancak bu sefer küçük operasyon süreli işler yerine büyük operasyon süresine sahip işler önceliklidir.

3.1.5 En Erken Teslim Zamanı

Siparişler içerisinde erken teslim edilecek işler sıralanarak çizelgeleme yapılır. Bu sayede gecikmelerin önüne geçilmeye çalışılmakta ve müşteri memnuniyeti artmaktadır.

3.1.6 En Az Kalan Operasyon Sayısı

Her bir işin birden çok operasyondan geçeceği durumda sipariş tesliminin gecikmesini engellemek için kullanılır. Siparişler operasyon sayısına göre küçükten büyüğe göre sıralanır. Buna göre çizelgeleme işlemi gerçekleştirilir.

3.1.7 En Kısa Kalan İşlem Süresi

Siparişlerin toplam üretim sisteminde kalma zamanını kısaltmak için kullanılır. Toplam üretim süreleri küçükten büyüğe sıralanır. Bu sıralamaya göre işler çizelgeye katılır.

3.1.8 En Uzun Kalan İşlem Süresi

Kapasite kullanım oranının daha yüksek olması amaçlanmaktadır. İşler arasından en uzun toplam işleme süresine sahip işin ilk operasyonu çizelgenerek başlanır. Her çizelgeleme adımından sonra kalan toplam işlem süreleri tekrar kontrol edilir ve en büyük kalan işlem süresine sahip işin sıradaki operasyonu çizelgeye dahil edilir.

3.1.9 En Çok Kalan Operasyon Sayısı

Bu kuralda operasyon sayısı yüksek olan işlerin daha erken çizelgenmesi sağlanır. Her atama işleminin ardından işlerin kalan operasyon sayıları kontrol edilir ve en yüksek kalan operasyon sayısına sahip işin sıradaki operasyonu çizelgeye dahil edilir.

3.2 Meta-sezgisel Arama Metodu

Kompleks kombinasyonel problemlerde optimum çözümleri bulmak genellikle çok uzun zaman almaktadır. Bu nedenle bu tip uzun zaman alan problemlerde iyi sonuçlar üreten ancak kısa zamanda sonuca ulaşan algoritmalar tercih edilmektedir. Bu aramalar, rasgele arama veya problemin parçalarını sırasıyla çözen algoritmalar olabileceği gibi, probleme has bilgileri kullanarak arama uzayını daha etkili ve etkin şekilde gezen ve yerel optimumlardan daha kolay kurtulabilen algoritmalar da olabilmektedir. Daha iyi sonuçlar veren son tanıma ait algoritmalara meta-sezgisel algoritmalar denilmektedir. [26]

3.2.1 Keşfetme ve Gezinme (Exploration and Exploitation)

Bir arama algoritmasının etkili olabilmesi için hem yeni arama uzaylarını keşfedebilmesi, yeni ve bilinmeyen arama uzaylarında dolaşabilmesi hem de bilinen arama uzayında olabilecek çözümleri gezebilmesi gerekmektedir. Bu iki ihtiyaç birbirini ile çelişmektedir. Bu sebeple iyi bir arama algoritması bu ikisini dengeleyebilmelidir. Sadece rastgele arama yapan bir algoritma yeni yerler keşfederken, sadece tepe tırmanma algoritması yapan bir arama algoritması bilinen alanlarda gezmektedir.[27]

Holland [28] Genetik Algoritmanın bu iki ihtiyacı aynı zamanda gerçekleştirdiğini göstermiştir.

3.2.2 Genetik Algoritma

Genetik Algoritma eşeyli üreme metaforunun optimizasyon problemlerine aktarılması yöntemiyle çalışan bir arama mekanizmasıdır.[29] Doğada olduğu gibi iyi olanın seçilmesi ve üreme yöntemi ile yeni bireyler oluşturularak aramaya devam edilir. [28]

Genel olarak bireyler ikili sayılarla gösterilir ancak probleme has sembol setiyle de temsil edilmektedirler. Kromozomu oluşturan bu semboller belli bir çözümleri ifade

etmektedir. Gezen satıcı problemi düşünülürken kromozom rotayı oluştururken, her bir şehir bir gene denk gelmektedir.

Önemli noktalardan bir tanesi problemin semboller ile nasıl temsil edileceğidir. Problemi genetik algoritmaya uygun gen yapısına çevirirken farklı çözümler aynı kromozom gösterimi ile temsil edilebilmektedir. Bu da arama sırasında karışıklığa sebep olabilmektedir.

Genetik algoritmanın konvansiyonel çözüm yönteminden farklılaşan yanı bir çözüm havuzu ile arama yapılmasıdır. Bu çözüm havuzu “popülasyon” olarak adlandırılmaktadır. Havuzdaki birey sayısı parametrik olarak belirlenir ve çözüm zamanını ve sonucu etkileyen ana etkenlerden biridir. Birey sayısının az olması erken yakınsamalara, birey sayısının fazla olması ise hesaplama zamanı açısından gereksiz zaman kaybına sebebiyet verebilmektedir

Çizelge 3.1 GA Popülasyonu

Kromozom 1	1 0 1 1 0 1 0 1 1
Kromozom2	1 0 0 0 0 1 1 1 1
Kromozom3	1 1 1 1 0 0 0 0 1
Kromozom4	0 0 0 1 1 1 0 1 0

GA'nın önemli avantajlarından bir tanesi arama uzayında farklı olası çözümleri rahatlıkla gezebilmesidir. [30]

Bir genetik arama algoritması aşağıdaki adımları takip etmektedir. [31]

Başlangıç Popülasyonu: Arama işleminin yapılabilmesi için bir başlangıç popülasyonuna ihtiyaç duyulmaktadır. Genellikle rassal olarak oluşturan başlangıç popülasyonuna probleme has bilgi kolaylıkla eklenerek ilk popülasyon oluşturulabilmektedir.

GA'nın genel adımları aşağıdaki şekildedir. (Bkz Şekil 3.10)

1. Değerlendirme: Popülasyon oluşturulduktan sonra bireylerin amaç fonksiyonuna göre uygunluk değeri hesaplanır.
2. Seçim: Havuzun içinde uygunluk değerine göre bireylerin seçilmesidir. İyi bireylerin seçilmeleri yani doğal seçim bu aşamada gerçekleşir. Genel fikir iyi çözümlerin daha çok tercih edilmesi üzerinedir.
3. Rekombinasyon: İki ya da daha fazla bireyin özelliklerini kullanarak yeni bireyler oluşturulmasıdır. Yeni oluşan birey (offspring) ebeveynlerinin özelliklerini taşıyacak fakat hiçbirine tam olarak benzemeyecektir [30].
4. Mutasyon: Kombinasyon birden fazla bireyin özelliklerinin kullanılması ile oluşurken, mutasyon tek birey üzerinde belirlenen bir veya daha fazla özelliğinde rassal değişiklik yapılmasıdır.
5. Yerine Koyma (Replacement): Yeni oluşan veya üzerinde değişiklik yapılan bireylerin eskileri ile yer değiştirmesidir. Birey havuzundan bir kısım birey çıkarılırken yerine yeni oluşan bireyler konmaktadır.
6. 2-6 arasındaki adımlar durma kriterine gelinceye kadar tekrarlanır.

Bu adımlar aşağıda daha detaylı olarak anlatılmaktadır.[29,31]

3.2.2.1 Kodlama

Problemin GA ile yürütülebilmesi için uygun kodlama ile GA'nın yapısına yani kromozom formatına dönüştürülmesi gerekmektedir. Kodlama her bir çözümün genler aracılığı ile gösterilme işlemidir. Probleme has kodlamalar gerçekleştirilse de genel olarak aşağıdaki kodlamalar kullanılmaktadır.

İkili kodlama

En yaygın kullanılan kodlama şeklidir. Genler 0 ve 1 değerlerini almaktadır. Her bir değer çözümün bir karakteristiğini temsil etmektedir. Aşağıda bir örneği görüntülenebilmektedir.

Çizelge 3.2 İkili Kodlama

Kromozom 1	0 1 1 0 1 0 0 0 1 0
Kromozom 2	1 1 1 0 0 1 1 0 1 1

Sıralı Kodlama

Her bir kromozom bir sayı dizisinden oluşmaktadır. Kimi durumlarda genetik operasyon sonrası düzeltme işlemi yapılması gerekmektedir. Çoğunlukla sıralama problemleri için kullanılmakta ancak bu durumlarda da genetik operasyon sonrası düzeltmelere ihtiyaç duyulmaktadır. Aşağıda bir örneği gösterilmektedir.

Çizelge 3.3 Sıralı Kodlama

Kromozom 1	3 4 5 1 6 7 8 10
Kromozom 2	4 5 6 7 8 1 3 10

Değer Kodlama

Her kromozom belirli değerlerden oluşmaktadır. Bu kodlama ile problemin karakteristiği rahatlıkla kromozoma yansıtılabilmektedir. Bu tip problemleri ikili kodlama ile yansıtmak oldukça zordur. Ancak genellikle probleme has genetik operasyonların tasarlanması gerekmektedir.

Çizelge 3.4 Değer Kodlama

Kromozom	ABJCEDEGEHKEFEKGE
Kromozom	(geri), (ileri),(sağ),(sol)

3.2.2.2 Uygunluk fonksiyonu

GA seçim mekanizmaları üzerine kurulu bir algoritmadır ve bu seçimlerde kontrol ölçütü uygunluk değeridir. Uygunluk değeri birçok problemde amaç fonksiyonu olarak değerlendirilebilir. Bir bireyin kalitesi “ f ” ile gösterilen uygunluk değeri ile belirlenmektedir.

GA operasyonlarının çalışabilmesi için bireyler probleme özgün bir kodlama işleminden geçirilirler. Bireylerin uygunluk fonksiyonunun hesaplanabilmesi için ilk olarak kodlanmış kromozomun çözümlenmesi gerekmektedir. Daha sonra amaç fonksiyonu hesaplanarak uygunluk fonksiyonu bulunur. Yüksek uygunluk fonksiyonu her zaman bireyin daha kaliteli bir birey olduğunu göstermektedir.

3.2.2.3 Seçim süreci

Seçim işlemi genel olarak iki sınıf altında toplanabilir. Oran bazlı seçim yöntemleri bireylerin uygunluk değerlerinin, toplam uygunluk değerine oranlanarak seçim işleminin yapıldığı yöntemken sıra bazlı seçim yöntemi bireylerin uygunluk değerine göre sıralanarak seçim işleminin yapıldığı yöntemdir.

Oran bazlı seçim yönteminde bir bireyin uygunluk değerinin toplama oranla yüksek olduğu durumlarda hızlı yakınsama sorunu ile karşılaşılabilir. Bu da arama uzayının bir kısmının kapsanması ve lokal optimumda takılma riskini doğurmaktadır.

Sıralama bazlı seçim daha etkili bir yöntemdir ve çözümü optimal çözümlere taşıyabilmektedir.

Seçim süreci aşağıdaki veri seti kullanılarak anlatılacaktır.

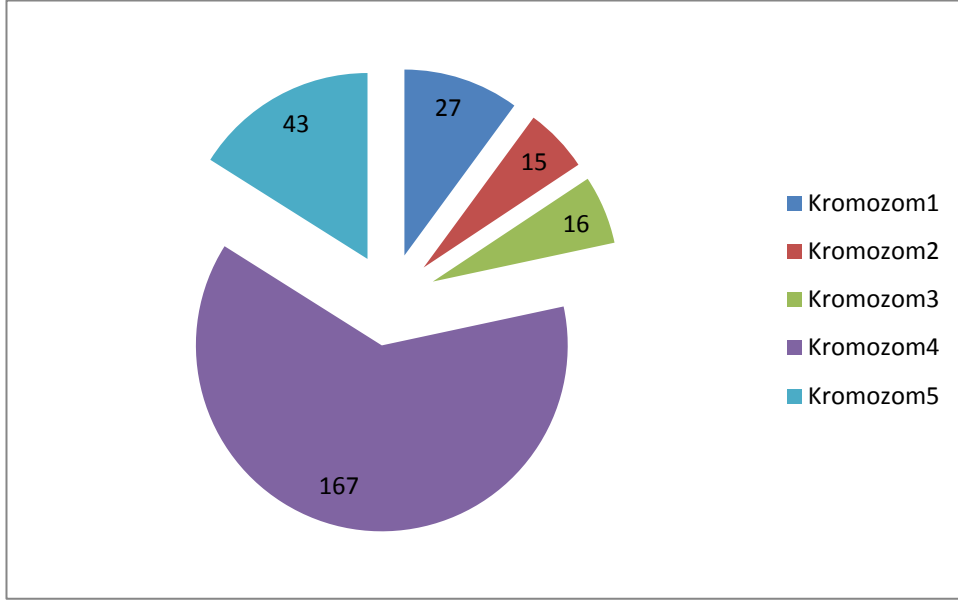
Çizelge 3.5 Popülasyonda Bireylerin Sırlaması ve Uygunluk Değerleri

Populasyon	Uygunluk Fonksiyon Değerleri	Küçükten Büyüğe Sıralama
Kromozom1	27	3
Kromozom2	15	1
Kromozom3	16	2
Kromozom4	167	5
Kromozom5	43	4

Rulet Bazlı Seçim Yöntemi

Her birey rulet tekeri üzerinde uygunluk fonksiyonunun büyüklüğüne yer alır. Rulet çevrilir ve durduğunda denk gelen birey seçilmiş olur.

- 1. Tüm bireylerin uygunluk fonksiyonlarını topla*
- 2. Bireyin uygunluk fonksiyon değerini adım 1de bulunan değere böl*
- 3. Adım 2'de bulunan değerleri toplayarak bireyler için rulet üzerinde aralık belirle*
- 4. Rassal bir sayı üret*
- 5. Rassal değer den geldiği aralıktaki bireyi seç*

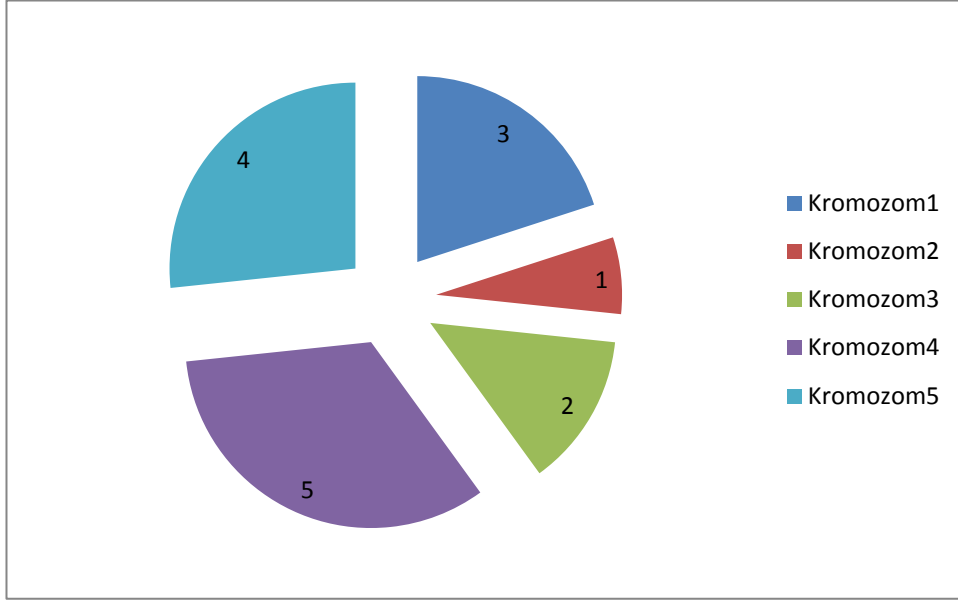


Şekil 3.1 Rulet Bazlı Seçim Örneği

Sıralı Seçim Yöntemi

Bu yöntem rulet bazlı seçim yöntemine benzemektedir, farklı olarak uygunluk fonksiyon değerleri doğrudan kullanılmak yerine bireyler bu değerlere göre küçükten büyüğe doğru sıralanarak, sıralamaya paralel değerler atanır. Genellikle lineer olarak artan değerler atanmaktadır.

1. Tüm bireylerin uygunluk fonksiyonlarına göre küçükten büyüğe sırala
2. Belli bir fonksiyona göre artan diziliş sırasıyla bireylere değer ata
3. Bu değerleri topla
4. Bireyin atandığı değeri adım 3'te bulunan değere böl
5. Adım 4'te bulunan değerleri toplayarak bireyler için aralık belirle
6. Rassal bir sayı üret
7. Rassal değerın denk geldiği aralıktaki bireyi seç



Şekil 3.2 Sıralı Seçim Yöntemi Örneği

Turnuva Seçim Yöntemi

Rassal olarak seçilen ve turnuva havuzuna giren bireyler birbirleri ile yarışır. Turnuva havuzu büyüklüğü parametrik olarak belirlenir. Turnuva havuzundakiler arasından en iyi birey seçilir. Genel olarak 2 birey turnuva havuzuna girmektedir.

1. *Rassal olarak n birey seç*
2. *N birey arasından uygunluk değeri en yüksek olan bireyi seç*

3.2.2.4 Rekombinasyon

Çaprazlama genellikle iki bireyin alınarak, bunlardan yeni çocuk bireylerin oluşturulmasına denilmektedir. Amaç popülasyonun daha iyi bireyler ile devam etmesidir. Bireyler genellikle belirli bir olasılık değerine göre seçilmekte ve çaprazlama işlemine sokulmaktadır. Çaprazlama işleminde şu adımlar uygulanmaktadır.

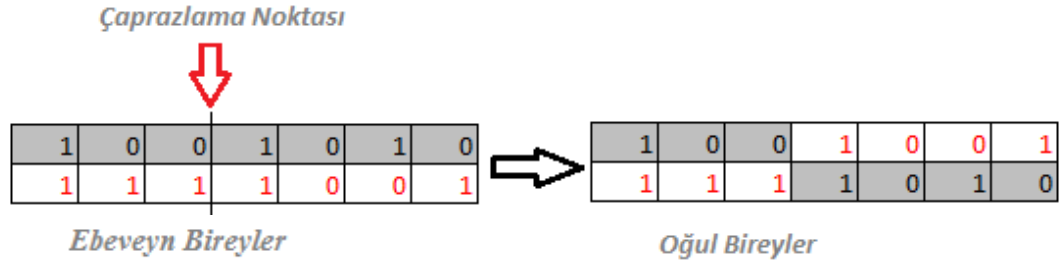
1. *Rassal olarak iki birey seçilir.*
2. *Kromozom üzerinden belirli bir alan seçilir.*

3. İki birey arasında bu alan birbiriyle yer değiştirir.

Probleme has çaprazlama operatörleri de tasarlanabilmektedir. Sık kullanılan çaprazlama yöntemleri aşağıda ele alınmaktadır.

Tek Noktadan Çaprazlama

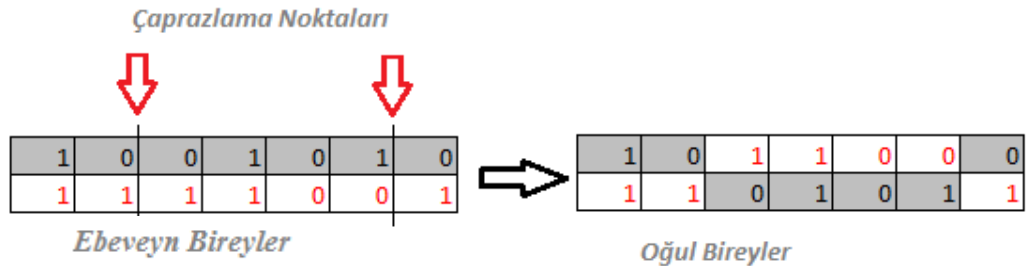
Geleneksel genetik algoritma bu yöntemi kullanmaktadır. İki birey seçilmekte ve bu bireyler üzerinde belirlenen bir noktadan sonrası kesilerek değerler yer değiştirir. Bu şekilde yeni bireyler üretilmiş olmaktadır.



Şekil 3.3 Tek Noktalı Çaprazlama Örneği

İki Noktalı Çaprazlama

Bu yöntem tek noktalı çaprazlamaya benzer şekilde işlemektedir. Ancak bir değil iki nokta seçilmekte ve arada kalan değerler iki birey arasında yer değiştirilerek yeni bireyler oluşturulmaktadır. Kromozomun başında ve sonunda kaliteli genetik bilgi bulunduğu durumlarda tek noktalı çaprazlamadan kaynaklı bir kayıp söz konusuysen iki noktalı çaprazlama ile bunun üstesinden gelinmektedir.

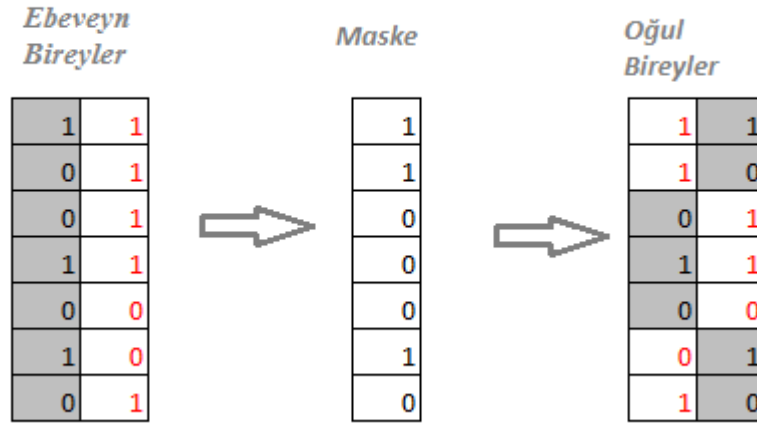


Şekil 3.4 İki Noktalı Çaprazlama Örneği

Uniform Çaprazlama

Yine yoğun şekilde kullanılan yöntemlerden bir de uniform çaprazlamadır. [33,34]. Kromozom üzerindeki tüm genler üzerinde kontrollü seçim yapılmaktadır. İlk olarak kromozom uzunluğuna göre ikili değerlerden oluşan bir maske üretilmekte ve maske üzerindeki değerlere göre hangi bireyden genlerin geleceği belirlenmektedir. İki birey üzerinden yapılacak çaprazlama için 0 ve 1 değerleri kullanılır. Bu değerler rassal olarak üretilir ve genellikle oran olarak 0,5 kullanılır

Aşağıda grafik üzerinde bu durum örneklenmiştir.



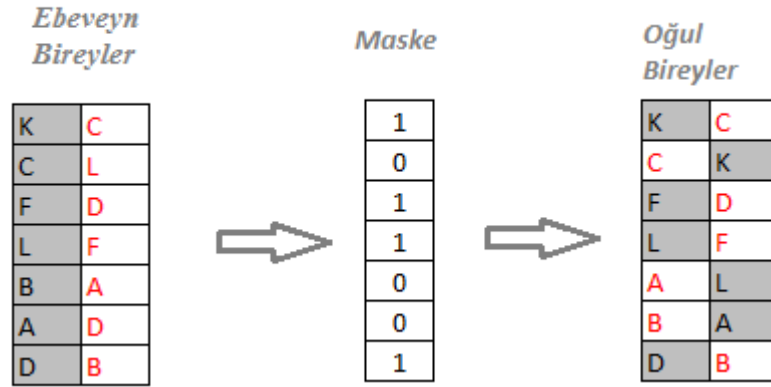
Şekil 3.5 Uniform Çaprazlama Örneği

Sıra Bazlı Uniform Çaprazlama

Sıra bazlı uniform çaprazlama yönteminde uniform çaprazlamaya benzer olarak yine bir maske üretilmekte, ilk olarak master bireyden maskeye göre genler kopyalanmakta, kromozom üzerinde kalan boşluklar diğer bireydeki diziliş sırasına göre kromozoma yerleştirilmektedir.

K noktalı çaprazlama veya uniform çaprazlama yöntemleri permutasyon bazlı kodlama ile oluşturulmuş arama problemlerinin yapısına uymamaktadır. Bu tip problemlerde probleme has onarma işlemlerine ihtiyaç duyulmaktadır.

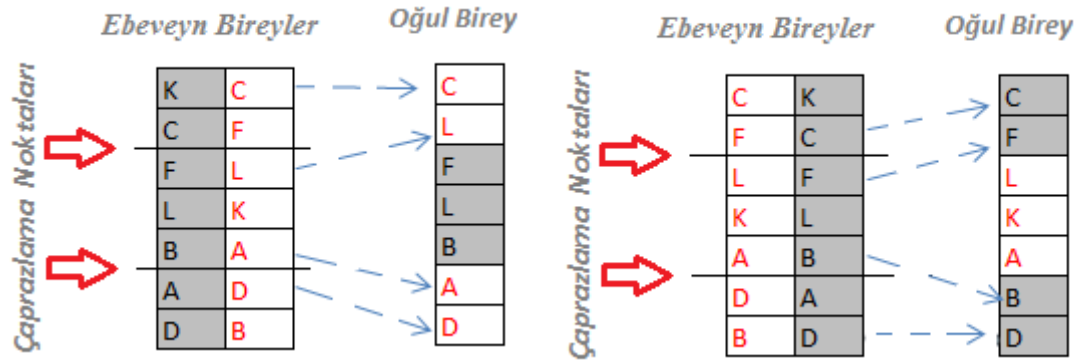
Aşağıda gezgin satış problemi için çaprazlamanın nasıl yapıldığı gösterilmektedir. Her harf bir şehre denk gelmektedir.



Şekil 3.6 Sıra Bazlı Uniform Çaprazlama Örneği

Sıra Bazlı Çaprazlama

Sıra Bazlı Çaprazlama[35], sıra bazlı uniform çaprazlamaya benzemektedir. Çaprazlama şeklini belirleyen maske her bir gen için oluşturulmak yerine iki noktalı rassal değer ile belirlenen alanın tamamı sabit kalırken geri kalan alan değişikliğe tabi tutulmaktadır. Diğer işlemler sıra bazlı uniform çaprazlamada olduğu gibi gerçekleştirilir.

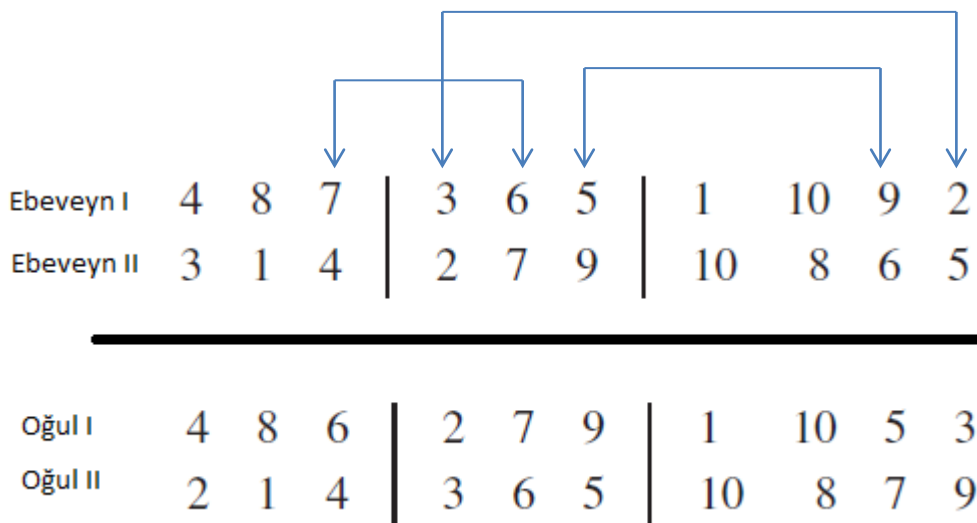


Şekil 3.7 Sıra Bazlı Çaprazlama Örneği

Kısmi Eşlemeli Çaprazlama Yöntemi

Bu yöntem [36] permütasyon bazlı kodlama ile oluşturulmuş arama yöntemleri için kullanılmaktadır. İki nokta arasında kalan genler diğer bireyin aynı alanda kalan genleri ile yer değiştirmektedir. Birey üzerinde oluşan boşluklar yine aynı birey üzerindeki değişikliğe uğrayan genlerin sırasına göre yerleştirilmektedir.

Aşağıdaki örnekte bu durum açıklanmaktadır.

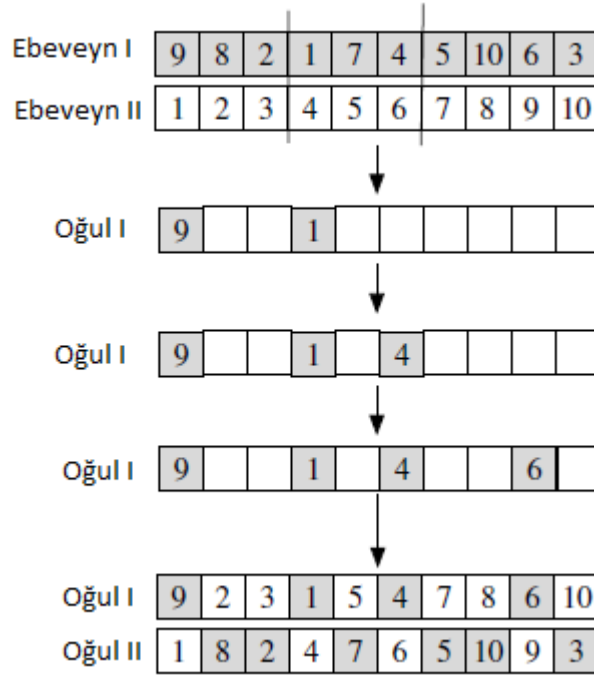


Şekil 3.8 Kısmi Eşlemeli Çaprazlama Örneği

Döngüsel Çaprazlama

Döngüsel Çaprazlama [37] işlemi belirlenmiş iki bireyden biri üzerindeki genin seçilmesi ile başlar. Bu genin kromozom üzerinde aynı sırada diğer bireyde denk geldiği gen bulunur. Bu gen yine ilk bireydeki yerine yerleştirilerek işleme devam edilir. Daha önce yerleştirilmiş bir gen bulununcaya kadar döngü yukardaki şekilde ilerler ve daha önce yerleştirilmiş bir gen bulunduğu anda döngüsel atama tamamlanır. Geri kalan genler, işlemin başlatılmadığı birey üzerindeki gen sırasına göre yerleştirilir.

Aşağıdaki örnekte bu durum açıklanmaktadır.



Şekil 3.9 Döngüsel Çaprazlama Örneği [30]

3.2.2.5 Mutasyon

Çaprazlama işleminden sonra kromozomlara mutasyon işlemi uygulanır. Genel amaç arama uzayını lokal minimum alandan kurtarıp farklı arama uzaylarında daha iyi bireyler aranmasını sağlamaktır. Çaprazlama belirli bir alan içinde arama işlemi

gerçekleştirirken, mutasyon farklı arama alanlarının keşfedilmesine olanak sağlamaktadır. Mutasyon genelde basit bir arama yöntemi olarak görülür ancak keşif alanını genişleterek arama uzayını genişletir. Eğer problem içindeki her çözüm arama uzayında erişilebiliyorsa bu arama uzaylarına ergodik adı verilmektedir. Mutasyon işlemi arama uzayının ergodik olmasını sağlamaktadır.[27]

Farklı mutasyon teknikleri kullanılmakta ve probleme has daha iyi bireyler bulmak üzere tasarlanabilmektedir. Bu tarz mutasyonların arama uzayını lokal optimumlara çekme ihtimali bulunmaktadır. Bu sebeple mutasyon operatörü tasarlanırken dikkatli davranılmalıdır.

Mutasyonda tek bir bireyin genleri üzerinde değişiklik yapılmaktadır. Belirlenen bir olasılık değerine göre birey üzerinde mutasyon yapıp yapılmayacağına karar verilir.

Ters Çevirme

İkili gösterimleri için ters çevirme mutasyon işlemi gerçekleştirilmektedir. 0 ve 1 değerini alan genler üzerinde 1 değeri 0'a, 0 değeri 1'e dönüştürülmektedir.

Yer Değiştirme

İki farklı gen belirlenerek bunların yerleri değiştirilir.

Komşu İki İş Değiştirme

Rassal olarak bir nokta seçilir ve yan yana duran iki gen yer değiştirir.

3.2.2.6 Sonlama Kriterleri

Daha önce anlatılan kavramlar, tek nesil üzerinde yapılan işlemleri açıklamaktadırlar. Tek nesil üzerinde yapılan işlemler belirli kriterler sağlanıncaya kadar yeni nesillerin

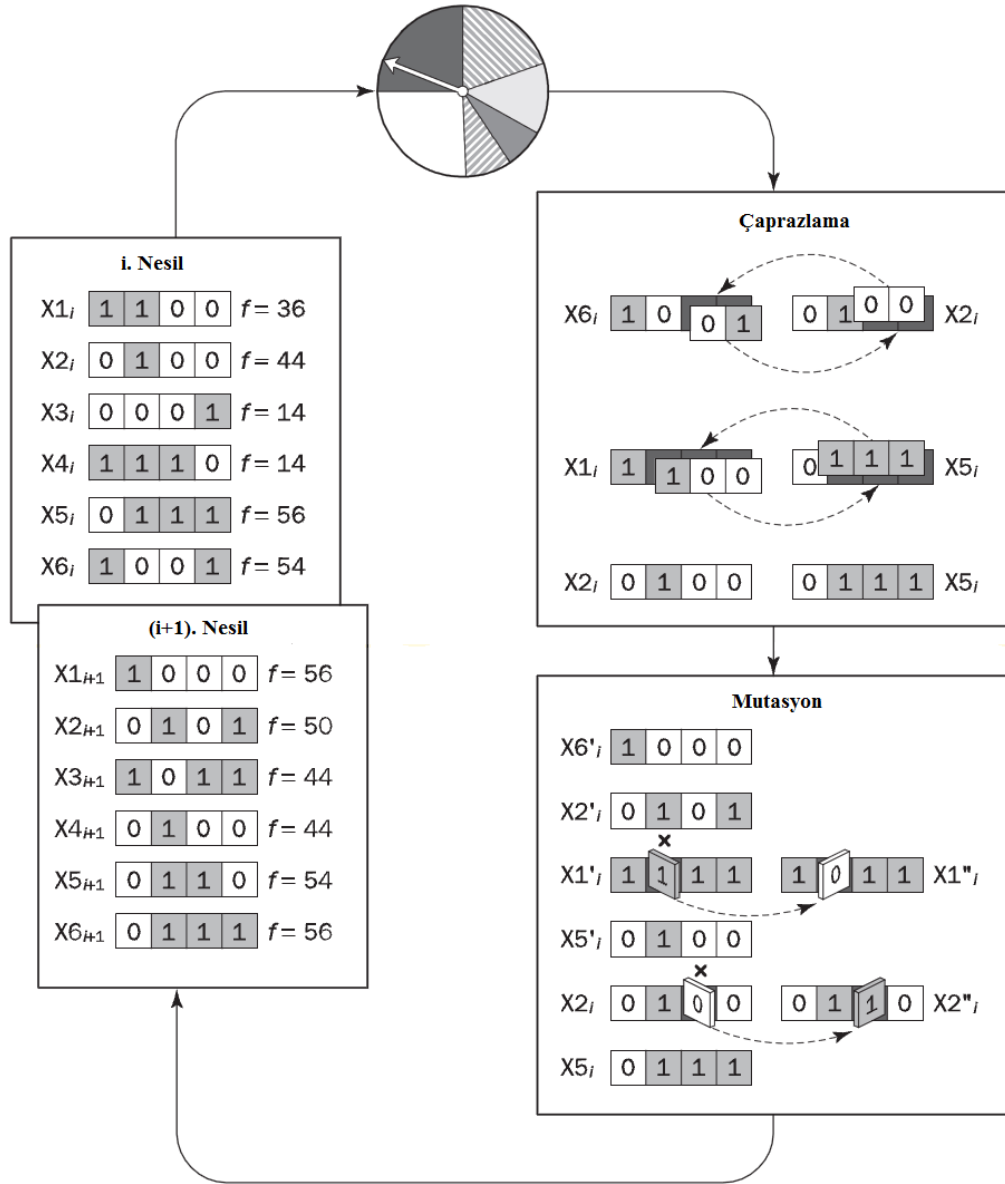
üretilmesi ile devam ettirilmektedir. Aşağıdaki koşullardan bir tanesi ya da birkaç tanesi GA adapte edilerek programın sonlanma zamanı belirlenir.

- Belirlenmiş Nesil Sayısına ulaşıldığında
- Maksimum çalışma süresine ulaşıldığında
- Uygunluk fonksiyonunda belirli bir nesil boyunca değişiklik olmadığında
- Uygunluk fonksiyonunda belirli bir zaman boyunca değişik olmadığında

3.2.2.7 Elitizm-Seçkinlik

GA, yapısı gereği ilerleyen nesillerde iyi çözümleri kaybedebilmektedir. Ayrıca birçok çalışma GA'nın sonuca ulaşabilmesi için uzun hesaplama süresi gerektiğini göstermektedir [38]. Seçkinlik stratejisi ise GA'nın performansını arttırmakta etkili bir yöntemdir[39] .

Seçkinlik stratejisi ile en iyi çözümler bir sonraki nesle geçmekte, böylece genetik yapıdaki kaymalar azalmaktadır. Popülasyon büyüklüğüne göre belirli sayıda birey mutasyon ve çaprazlama işlemine uğramadan önce bir sonraki nesle aktarılmaktadır. Tüm GA boyunca en iyi bireyler bir sonraki nesle iletilmektedir.



Şekil 3.10 Doğal Seçim ve Genetik Operatörler Döngüsü[40]

3.3 Rassel Bozma (Perturbation)

Storer vd [41] ATÇP problemi için yeni bir arama uzayı önermişlerdir. Operasyonların işlenme zamanlarında P_{ij} (i işi ve j operasyonu göstermektedir) artı ve eksi yönde değişiklik yaparak çözmeye çalıştıkları problem üzerinden yapay

işlenme zamanları SP_{ij} ile yeni bir problem üretmişlerdir. En Kısa İşlem Süresi yöntemini kullanarak operasyonları sıralamaktadırlar ve asıl sonucu elde etmek istediklerinde aynı sıralamayı gerçek işlenme zamanları ile oluşturmaktadırlar.

Problem üzerinde yaptıkları değişiklik uniform dağılmış değerler eklemek ya da çıkartmaktır.

$$SP_{ij} = P_{ij} \pm U_{ij}, U_{ij} \sim U(-\theta, \theta)$$

θ çok büyük olduğunda arama işlemi rassal aramaya dönerken oluşan çizelgeler aktif çizelgeler dönmektedir. θ 'nın küçük olduğu durumda ise üretilen çizelgeler ertelemesiz çizelgeler olmaktadır. Daha önceki araştırmalar sezgisel aramaya ertelemesiz çizelge ile başlamanın aktif çizelgeye oranla daha iyi sonuçlar verdiğini göstermiştir [11].

Aşağıda ISP_{ij} olarak oluşturulan yeni problemin komşuluklarında arama yapılmakta ve daha iyi bir problem setine rastlanıncaya kadar bu setin baz olarak kullanıldığı sözde kod görüntülenmektedir.

Çizelge 3.6 Rassal Bozma İşlemi için Sözde Kod

En iyi $M = \infty$
 $ISP_{ij} = P_{ij}$ her i, j için
A ve C arasında S defa tekrarla
(A) 1 ve 4 arasında N defa tekrarla
1. $SP_{ij} = ISP_{ij} \pm U_{ij}, U_{ij} \sim U(-\theta, \theta)$ her i, j için
2. EKİS SP_{ij} 'ye uygula
3. En son biteninin tamamlanma zamanını bul
4. IF $M < \text{En iyi } M$
 EN iyi $M = M$ BSP $_{ij} = ISP_{ij}$
(B) $ISP_{ij} = BSP_{ij}$ her i, j için
(C) A adımına git

Bu yöntem kolay şekilde farklı çizelgeleme problemlerine ve farklı arama metotlarına uygulanabilmektedir [41]. Yaptığımız çalışmada bu yöntem başlangıç popülasyonunu oluşturmak için kullanılmıştır.

3.4 EATÇP’de Kullanılan GA Yaklaşımları

3.4.1 Kodlama

GA algoritmasının uygulanabilmesi için önemli gerekliliklerden bir tanesi kromozom kodlama şeklidir. Bu yöntemle ilgili olarak gerçekleştirilebilen GA operasyonları ve gerekli olan onarma işlemleri değişiklik göstermektedir. EATÇP probleminde sıklıkla aşağıdaki kodlamalar kullanılmaktadır.

3.4.1.1 Paralel Makine Kromozom Gösterimi

Paralel Makine Gösterimi [42] ile kromozom makine listesi şeklinde gösterilmektedir. Her bir makine kromozomda bir satır olarak gösterilmektedir.

Çizelge 3.7 Paralel Makine Kromozom Gösterimi

M1	(i, J _i , T _{ijh})	...
M2		...
M3	(i', J _{i'} , T _{i'jh})	...
...		
Mn

i = Operasyon numarasını

J_i = Bu operasyona denk gelen iş numarasını

T_{ijh} = O_{ij} operasyonunun M_h makinesinde başlama zamanını göstermektedir.

3.4.1.2 Paralel İş Kromozom Gösterimi

Kromozom iş listesi olarak gösterilmektedir. Her bir iş kromozomda bir satır olarak gösterilmekte ve operasyonların öncelik sıralarına göre dizilmektedir.

M_i = Operasyonun hangi makinede işleneceğini

$T_{m_i} = i$ operasyonunun m makinesinde başlama zamanını göstermektedir.

Çizelge 3.8 Paralel İş Kromozom Gösterimi

J1	(M_1, T_{M1})	(M_2, T_{M2})		
J2	(M_5, T_{M5})	(M_1, T_{M1})	(M_5, T_{M5})	
J3	(M_2, T_{M2})	...		
...				
Jn		

3.4.1.3 Gen vd Kromozom Gösterimi

Bu gösterimde[8] işlere ait operasyonlara sıra numarası verilmektedir. Kromozomda ilk makine atamaları kromozomda farklı bir satırda tutulmaktadır. Örneğin 15 numaralı operasyon 4. İşin 3. operasyonunu temsil etmektedir. Bu gösterimde GA operatörleri kolaylıkla uygulanabilmektedir.

Çizelge 3.9 Gen vd Kromozom Gösterimi [8]

Öncelik Sırası	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Operasyon Sırası	5	13	9	1	14	2	15	10	6	3	11	4	7	16	12	8
Makine Ataması	2	2	3	4	4	3	4	1	4	3	2	1	1	3	4	4

3.4.1.4 Makine Seçim ve Operasyon Seçim Kromozom Gösterimi (MSOS)

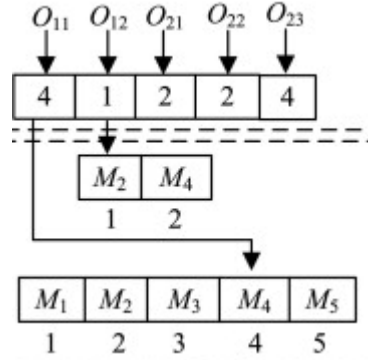
Bu gösterimde[22] makine seçimi ile operasyon seçimi kromozom üzerinde aşağıdaki şekilde tutulmaktadır. Operasyonlar kromozomda ait oldukları iş numaraları ile dizilmektedir. Kromozom üzerindeki sıralarına bakılarak hangi operasyon oldukları belirlenmektedir.

Çizelge 3.9 MSOS Kromozom Gösterimi [22]

Makine Seçimi (MS)					Operasyon Seçimi (OS)				
4	1	2	2	4	2	2	1	1	2

Her bir operasyon için ayrı bir liste olarak hangi makinelerin bu operasyonu işleyebildiği tutulmaktadır. Makine Seçimi bölümünde işlemi gerçekleştirecek asıl makine numarası yerine işlemi gerçekleştirebilecek makineler listesindeki sıra numarası tutulmaktadır. Böylece daha az onarım işlemi ile GA operasyonları gerçekleştirilebilmektedir.

Çizelge 3.10 MSOS Kromozom Makine Kısmı[22]



3.4.2 Seçim Yöntemleri

Genel olarak ikili seçim, çoklu seçim ve lineer sıralı seçim yöntemi kullanılmaktadır. Geçmişte rulet teker seçim yöntemi sıklıkla kullanılırken bugün turnuva seçim yöntemi ile sıkça karşılaşılmaktadır. Rulet seçim yönteminde bireyler uygunluk değerlerinin büyüklüğü oranında seçilme şansına sahipken turnuva yönteminde rasgele seçilen bireyler arasından en yüksek uygunluk değerine sahip birey seçilmektedir. Turnuva seçim yöntemi ile çok iyi bireylerin çoğunlukla seçilmesi ve erken yakınsama probleminin önüne geçilmektedir.

3.4.3 Çaprazlama

Sıklıkla aşağıdaki iki tür çaprazlama ile karşılaşılmaktadır.

3.4.3.1 Geliştirilmiş Sıralı Çaprazlama

Bu yöntemde ebeveyn üzerinden iki nokta seçilerek oğul birey aktarılırken kalan operasyonlar ikinci ebeveyn üzerinde sıra gözetilerek oğul bireye aktarılmaktadır.

Öncelik Sırası	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Operasyon Sırası	5	13	9	1	14	2	15	10	6	3	11	4	7	16	12	8
Makine Ataması	2	2	3	4	4	3	4	1	4	3	2	1	1	3	4	4

Operasyon Sırası	9	13	5	1	14	2	15	10	6	3	11	4	12	7	8	16
Makine Ataması	4	3	1	4	4	3	4	1	4	3	2	1	2	4	3	1

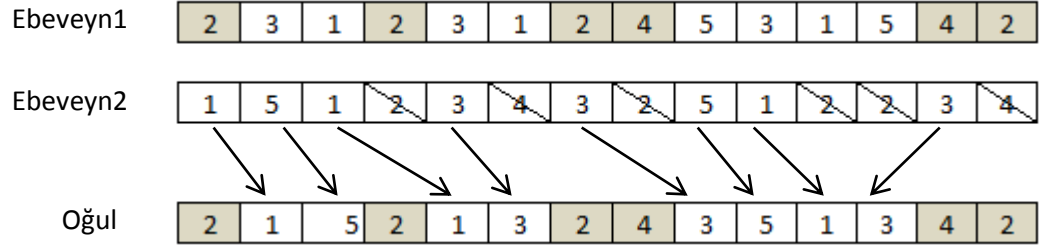
Operasyon Sırası	9	1	13	10	5	14	2	15	11	6	12	7	3	8	4	16
Makine Ataması	4	1	3	2	1	3	4	1	2	4	2	4	3	3	2	1
Öncelik Sırası	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16

Şekil 3.14 Geliştirilmiş Sıralı Çaprazlama

3.4.3.2 Öncelik Bazlı Sıralı Çaprazlama

Bu yöntem belli sayıda iş numarası belirlenerek, bu işe ait operasyonlar bir ebeveyninden gelirken kalan operasyonlar ikinci ebeveyn üzerindeki sıra gözetilerek oğul bireye aktarılmaktadır.

Aşağıdaki örnekte 2 ve 4 numaralı işler master bireyden gelecek şekilde seçilmiş ve buna göre çaprazlama işlemi yapılmıştır.



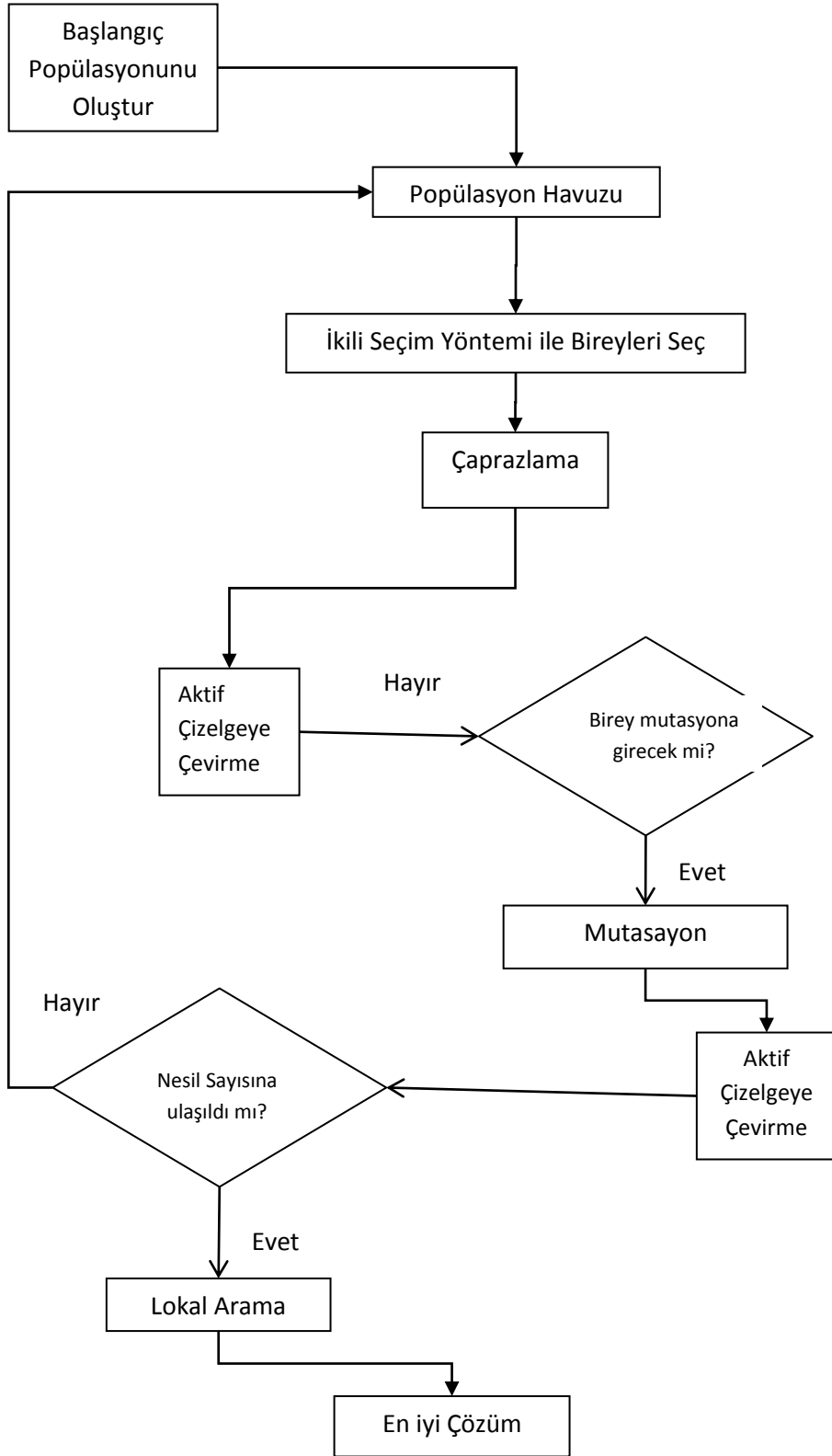
Şekil 3.15 Öncelik Bazlı Sıralı Çaprazlama

3.4.4 Mutasyon İşlemleri

Bu işlem genellikle çizelge üzerinde bulunan operasyonun atandığı makine veya çizelgedeki sırasının değiştirilmesi ile yapılmaktadır. Fakat EATÇP yapısına uygun daha iyi çözümler üretebilecek akıllı mutasyonlarla da sıklıkla karşılaşılmaktadır.

4. ÖNERİLEN YAKLAŞIMA GENEL BAKIŞ

Bu bölümde çalışmada yapılan metodlar anlatılmaktadır. İlk olarak kromozom kodlamasından bahsedilmekte, sonrasında başlangıç popülasyonunun nasıl oluşturulduğu anlatılmaktadır. Başlangıç popülasyonu oluşturma işlemi operasyonların makinelerde işleme süreleri rassal olarak karıştırılarak başlamakta, ardından operasyonlar belli kurallara göre makinelere atanmaktadır, son olarak da bazı öncelik kuralları kullanılarak operasyonların makinelerde çizelgenmesi sağlanmaktadır. Başlangıç popülasyonunun ardından popülasyon GA döngüsüne girmektedir. Bu döngü içinde her nesil çaprazlama ve mutasyon işlemlerinden geçmektedir. Yeterli sayıda nesil oluştuğunda GA uygulaması bitmekte ve son popülasyon lokal bir aramaya sokularak arama süreci tamamlanmaktadır. (Bknz Şekil 4.1)



Şekil 4.1 Geliştirilen Algoritmanın Akış Diyagramı

4.1 Kodlama

Üreme işlemlerinde baz olarak kromozomun kullanılması gibi EATÇP’de de muhtemel sonuçlar üreten her çizelge diğer bir deyişle her bir birey kromozom olarak adlandırılmaktadır. Bu çalışmada kullanılan kodlama işleminde iş-operasyon-makine numarası tek bir gen üzerinde gösterilmektedir. Temel olarak her bir gen bir operasyonu temsil etmektedir. EATÇP’de aynı işe ait operasyonlar arasında öncelik sırası bulunduğundan kodlama sırasında bu sıra her zaman göz önünde bulundurulur. İşlemler başlama zamanlarına göre kromozomun yapısına dizilmektedirler.

Aşağıdaki örnekte ilk gen için operasyon sırasındaki “1-1” gösterimi iş-operasyon eşleşmesini, 1. işin 1. operasyonu olduğunu ve 2 nolu makine de işlendiğini göstermektedir. Ayrıca ilk sırada olmasından dolayı üretimin başlangıç işlemi olduğunu göstermektedir.

Kromozomda üçüncü sırada bulunan “2-1” nolu operasyon, “1-2” nolu operasyondan sonra geldiği ve bu operasyonla aynı makinada işlendiği için işlenmeye başlayabilmesi için “1-1” ve “1-2” nolu işlemlerin tamamlanması gerekmektedir.

Çizelge 4.1 Kullanılan Kromozom Gösterimi

Operasyon Sırası	1-1	1-2	2-1	1-3	3-1	3-2	2-2	2-3
Makine Ataması	2	1	1	3	2	3	2	4

Bu yapıdaki bir kromozom kodlaması mümkün olan tüm çözüm kümesini kapsamakta ve makine atamaları ve çizelgeleme kararlarını tek bir gen üzerinde temsil etmektedir. [17]

Önemli bir ayrıntıda, genetik operasyonlar sırasında oluşan tüm bireylerin aktif çizelgeye dönüştürdüğü ve nesil havuzuna bu şekilde aktarıldığıdır. (Bkz Şekil 4.2)

Çizelge 4.2 Aktif Çizelgeye Dönüştürme - Sözde Kod

Kromozom üzerindeki operasyonların sıralaması alınır

FOR $i=1$ to n // n toplam iş sayısı

$k_i \leftarrow 0$ // k_i i operasyonu için atanmış operasyon sayısı

ENDFOR.

FOR $s=1$ to N // N toplam operasyon sayısı

$O \leftarrow K_{rmz}(s)$ s. sıradaki operasyon

$m \leftarrow M(s)$ s. sıradaki operasyona atanmış makine no

$j \leftarrow J(s)$ s. sıradaki operasyonun ait olduğu iş

$k_i \leftarrow k_i + 1$

O operasyonun işleme süresi için m makinesinde soldan sağa, $j-1$ işinin başlama zamanını da dikkate alarak sığabileceği boşluğu ara.

Eğer böyle bir aralık bulunursa O operasyonunu buraya yerleştir yoksa makinenin en sonuna.

ENDFOR.

Operasyonların kromozom üzerine yerleştirilmesi, operasyonların başlangıç zamanına göre yapılmakta ve önce başlayan iş kromozomda daha önlere yer almaktadır.

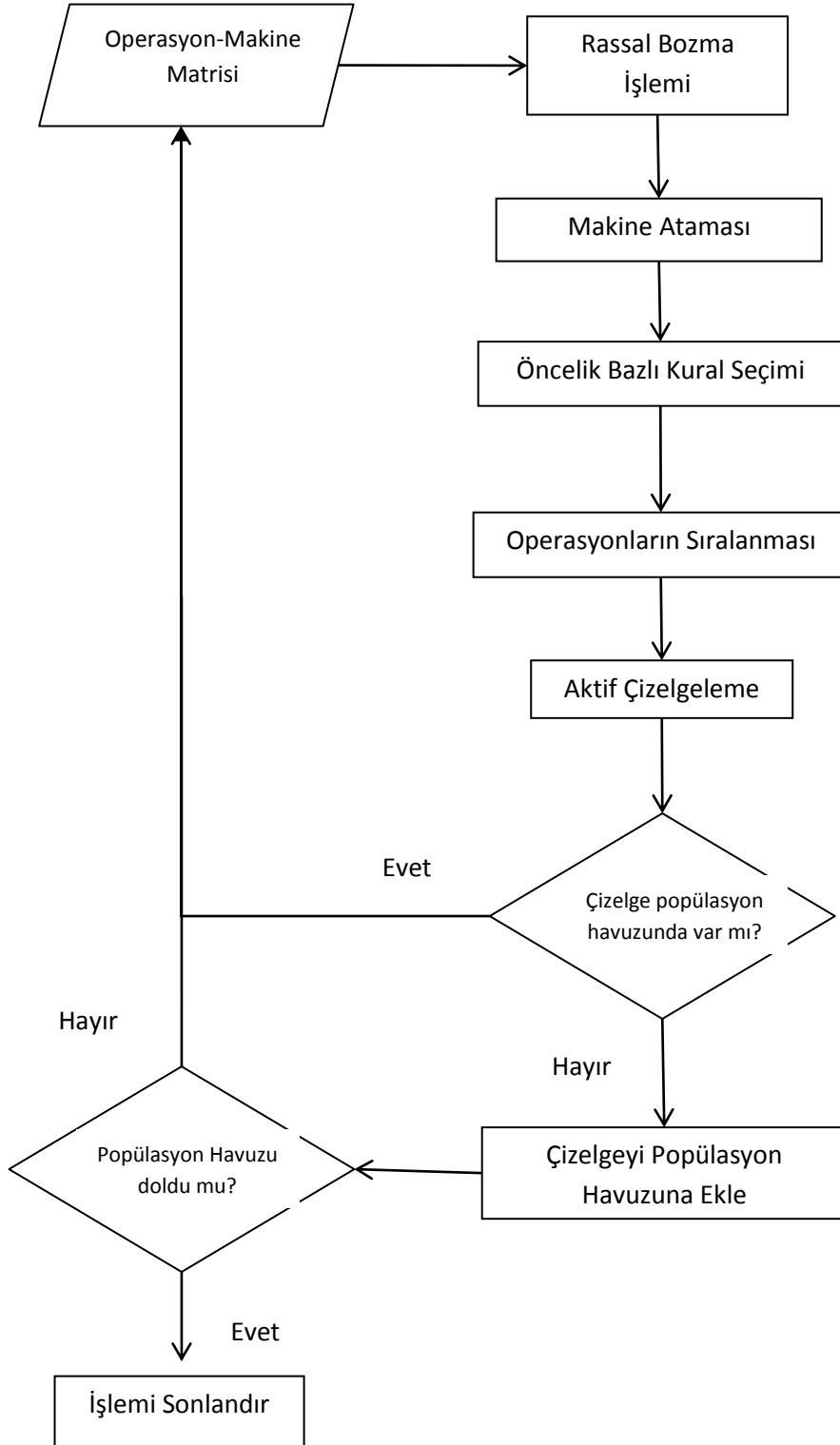
4.2 Başlangıç Popülasyonu

GA mevcut çözüm üzerinde değişiklik ve alternatif çözümler arasında veri alışverişi yaparak çalışmaktadır. Bu sebeple üzerinde işlem yapabileceği bireylere yani çözümlere ihtiyaç duymaktadır. Farklı yöntemler kullanılarak ilk bireylerden oluşan başlangıç popülasyonu elde edilebilmektedir.

Bu çalışmada başlangıç popülasyonu atamaların ve çizelgelemelerin iki farklı problemmiş gibi ele alındığı hiyerarşik yaklaşım ile oluşturulmuştur. (Bkz Şekil 4.2) Atama kısmında işlere bağlı operasyonların makine eşleştirmeleri yapılırken, çizelgeleme bölümünde belirli öncelik kurallarına uyarak operasyonların makine üzerinde sıralaması ve başlangıç bitiş zamanları belirlenmektedir.

İlk popülasyonda her bir birey oluşturulmadan önce operasyonların üretim zamanlarında aşağıda anlatacağımız rassal bozma işlemi uygulanmaktadır. Bunun ardından problemler atama bölümünde anlatılan şekilde makinelerle eşleştirilmekte ve ardından daha önce de genel olarak bahsettiğimiz farklı öncelik bazlı kurallar ile çizelgelenmektedir. Her bir rassal bozma işlemi sonrası bir birey oluşturulmaktadır. Her bir bireyin sıralama işlemi aşağıda bahsedilecek dört öncelik bazlı çizelgeleme kuralından birinin rassal olarak seçilerek kullanılması ile meydana gelmektedir.

İlk popülasyonda çeşitliliğin artırılması için benzer bireylerin popülasyon havuzuna alınması engellenmiştir. Bunu yaparken havuzdaki bireylerin uygunluk değerleri yeni oluşturulan bireyin uygunluk değeri ile kıyaslanmakta, eğer aynı değere sahip bir birey bulunmuşsa, uygunluk değerlerinin eşleştiği birey ile iki noktalı operasyon-makine eşleşmesi kontrol edilerek bireyin havuza alınıp alınmayacağına karar verilmektedir.



Şekil 4.2 Başlangıç Popülasyonu Oluşturma Akış Diyagramı

4.2.1 Rassal Bozma (Perturbation)

Genetik Algoritmada iyi sonuçlar elde edilebilmesini sağlayan faktörlerden bir tanesi başlangıç popülasyonun çeşitliliği ve kalitesidir. Rassal bozma işlemi ile iyi bir popülasyon çeşitliliğinin ve kalitesinin sağlanması amaçlanmaktadır.

Rassal bozma, gelen işe ait operasyonların farklı makinelerde işleme sürelerini belirlenen bir parametreye göre değiştirme işlemidir. Rassal olarak her bir operasyon-makine eşleşmesinin işleme süresini yüzdesel olarak arttırmak veya azaltmak suretiyle yeni operasyon-işlem süresi matrisi çıkartılmakta ve yeni oluşan matrise göre operasyonlar belirli atama yöntemleri kullanılarak makinalara atanmaktadır. Rassal bozma parametresi yüzdesel olarak belirlenmekte, her bir operasyon-makine eşleşmesi için rassal olarak üretilen değerler ile işlem süreleri karıştırılmaktadır.

Geliştirilen algoritmada yüzde 30 oranında işleme süreleri üzerinde artı ya da eksi yönde değişiklik yapılacak şekilde çözüm kurgulanmıştır. İşleme süreleri üzerinde farklılaşmaya gidilerek iyi başlangıç popülasyon oluşmasına sebep olabilecek yeni işleme süreleri oluşturulmuştur.

Aşağıdaki örnekte karışıklık sonrası operasyon süreleri görüntülenmektedir.

Çizelge 4.3 Üretim Zamanları

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		13	13	3	3
2	11	8	11		5
3	11	7	23		14

Çizelge 4.4 Rassal Değişiklik Sonrası Üretim Zamanları

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		14,8	13	2,4	2,8
2	9,7	6,6	12,6		5,1
3	12,3	6,3	23,3		17,4

4.2.2 Atama Problemi

Atama işlemleri, farklı makinalarda işlenebilecek operasyonun hangi makinede işleneceğinin belirlendiği adımdır. İki farklı yöntem kullanılarak atama işlemi gerçekleştirilmektedir. Bu iki yöntemle göre problemler ayrı ayrı çözülmüştür, herhangi bir problem için aynı anda kullanılmamaktadır. Yöntemlerden birisi Kacem'in[13] önerdiği "Approach by Localization" yöntemi iken diğeri bu yöntemden yola çıkarak bizim önerdiğimiz yeni bir atama yöntemidir. Approach by Localization yöntemi, operasyonun iş üzerindeki sırasına bakmaksızın makine üzerindeki toplam iş yükünü minimize edecek şekilde operasyonların makine atamalarını gerçekleştirmektedir.

4.2.2.1 Atama Yöntemi – "Approach by Localization"

Başlangıç aşamasında makinalar üzerinde herhangi bir iş bulunmadığı varsayılmaktadır. Atama işlemi, işlenme süresi en küçük olan operasyonun ilgili makineye atanması ile başlamaktadır. Atama yapıldıktan sonra makine üzerindeki iş yükü arttığından, ilgili makinada işlenebilecek operasyonların işleme süreleri, atama yapılan operasyonun işlem süresi kadar artırılmakta ve atama yapılan operasyon listeden çıkartılmaktadır. Kalan operasyon listesi üzerinden yeni oluşturulan tabloda işlenme süresi en küçük olan operasyon-makine ikilisi bulunarak atama işlemi gerçekleştirilir. Aşağıda bu işlemin nasıl yapıldığı detaylı olarak gösterilmektedir.

Çizelge 4.5 Operasyon İşlenme Zamanları

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		13	13	3	3
2	11	8	11		5
3	11	7	23		14

Operasyonların işlenme sürelerinin rassal değişiklik yapıldıktan sonraki durumu aşağıda görüntülenmektedir.

Çizelge 4.6 Operasyon İşlenme Zamanları Rassal Değişiklik Sonrası

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		14,8	12,5	2,4	2,8
2	9,7	6,6	12,6		5,1
3	12,3	6,3	23,3		17,4

Atama işlemleri aşağıdaki gibi başlamaktadır. İşlenme zamanı en küçük olan operasyon seçilir ve ataması gerçekleştirilir.

Çizelge 4.7 Atama İşlemi - 1

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		14,8	12,5	2,4	2,8
2	9,7	6,6	12,6		5,1
3	12,3	6,3	23,3		17,4

Atama işlemi yapıldıktan sonra bu operasyon atanabilecek işler listesinden kaldırılır ve bu operasyon ile aynı makinede işlenebilen diğer operasyonların işlenme süreleri arttırılır.

Çizelge 4.8 Atama İşlemi - 2

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		17,2	14,9	0	5,2
2	9,7	6,6	12,6		5,1
3	12,3	6,3	23,3		17,4

Çizelge 4.9 Atama İşlemi - 3

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		17,2	14,9	0	0
2	14,8	11,7	17,7		0
3	12,3	6,3	23,3		0

Çizelge 4.10 Atama İşlemi - 4

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		0	14,9	0	0
2	14,8	0	17,7		0
3	18,6	0	29,6		0

Çizelge 4.11 Atama İşlemi-5

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		0	14,9	0	0
2	0	0	32,5		0
3	0	0	29,6		0

Son haliyle operasyonların makine atamaları aşağıdaki tabloda görüntülenmektedir.

Çizelge 4.12 Makine Atamaları

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		0	1	1	0
2	1	0	0		1
3	0	1	0		0

4.2.2.2 Önerilen Atama Yöntemi – Uzun İşlerin Önce Atanması

Atama işleminde kullanılan ikinci yöntem ise atamaların operasyon makine ikilisinden öte, ortalama işlenme süresi kullanılarak yapılmasıdır. Bu yöntemde tüm makinalardaki işlenme süresi, operasyonu işleyebilen makine sayısına bölünerek ortalama işlem süresi hesaplanır ve ortalama işlem sürelerine göre operasyonlar büyükten küçüğe sıralanır. Atama işlemi sırası gelen operasyonun farklı makinalarda yine toplam mevcut makine yükünü en az artıracak makinanın seçilmesi ile yapılmaktadır. Bu yaklaşım ile uzun işleme süresine sahip işlemlerin son anda

atamaya dahil olarak makine işi yükünü dengesiz şekilde artırmasının önüne geçilmesi amaçlanmaktadır.

Örnekte yukarıda bahsi geçen işleme süreleri kullanılacaktır.

En büyük ortalama işlenme süresine sahip operasyon seçilir ve makine yükü bakımından en kısa sürede işleyebilecek makineye bu operasyon atanır.

Çizelge 4.13 Atama İşlemi - 1

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		14,8	12,5	2,4	2,8
2	9,7	6,6	12,6		5,1
3	12,3	6,3	23,3		17,4
Ortalama	11	9,2	16,3	2,4	8,4

Atama yapılan operasyonun işlem süresi tüm makinelerde sıfıra çekilir ve listede bulunan ikinci en büyük ortalama işleme zamanına sahip operasyon, makine yükü bakımından en kısa sürede işlenebileceği makineye atanır.

Çizelge 4.14 Atama İşlemi - 2

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		27,3	0	14,9	15,3
2	9,7	6,6	0		5,1
3	12,3	6,3	0		17,4
Ortalama	11	9,2	0	2,4	8,4

Kalan atamalara en büyük ortalamaya sahip olan operasyon ile devam edilir.

Çizelge 4.15 Atama İşlemi - 3

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		27,3	0	14,9	15,3
2	0	16,3	0		14,8
3	0	6,3	0		17,4
<i>Ortalama</i>	0	9,2	0	2,4	8,4

Çizelge 4.16 Atama İşlemi - 4

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		0	0	14,9	15,3
2	0	0	0		14,8
3	0	0	0		23,7
<i>Ortalama</i>	0	0	0	2,4	8,4

Çizelge 4.17 Atama İşlemi - 5

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		0	0	14,9	0
2	0	0	0		0
3	0	0	0		0
<i>Ortalama</i>	0	0	0	2,4	0

Son haliyle operasyonların makine atamaları aşağıdaki tabloda görüntülenmektedir.

Çizelge 4.18 Makine Atamaları

Makine No	İş No				
	Operasyon No				
	1			2	
	1	2	3	1	2
1		0	1	1	0
2	1	0	0		1
3	0	1	0		0

4.2.3 Çizelgeleme Aşaması

Bir önceki adımda ataması yapılan operasyonların, üretime hangi sırada alınacağı bu aşamada belirlenir. Öncelik bazlı kurallar yardımıyla sıralama işlemi yapılır. Öncelik kuralları, operasyonun iş üzerindeki öncelik sırasını dikkate alarak belirlenen kural çerçevesinde en uygun adayın çizelgeye dahil edilmesi ile ilerler.

Başlangıç popülasyonu birçok bireyden meydana gelmektedir. Bu bireylerin farklılaşmasını sağlayan etkenlerden birisi atama işlemi iken diğeri farklı öncelik kurallarının uygulanmasıdır.

Çizelgeleme yöntemi olarak aşağıdaki dört öncelik kuralı çizelgeleme işlemi kullanılmaktadır. Bunlardan iki tanesi daha önce anlatılmıştır. Kalan ikisi bu bölümde anlatılacaktır. Bu iki kuralın, en son biten işin tamamlanma zamanı üzerindeki etkisi Şekil 4.3'te görüntülenmektedir.

- **En Kısa İşlem Süresi**
- **En Çok Kalan Operasyon Sayısı**
- **Kural1**
- **Kural2**

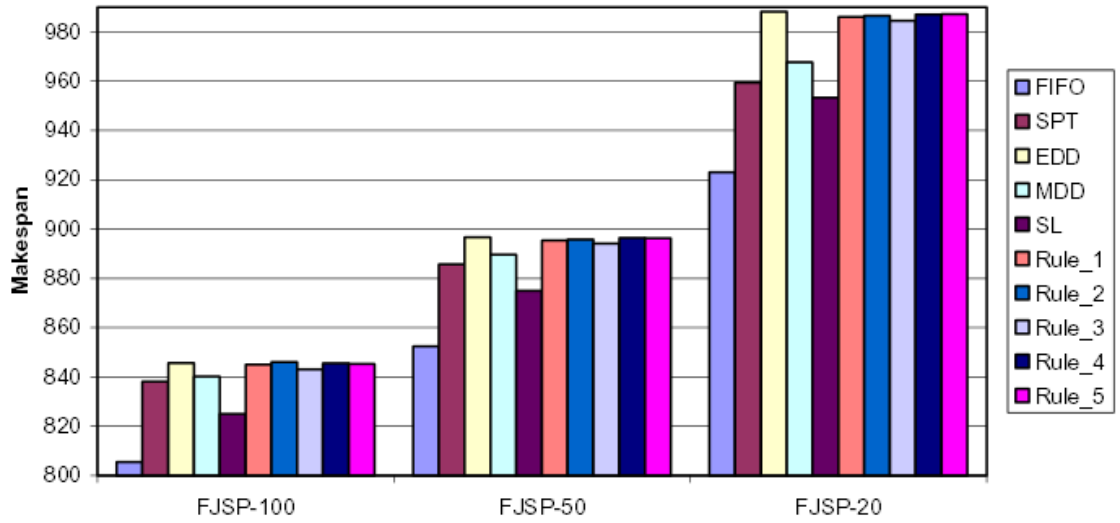
Kural1 ve Kural2 yine bir GA çalışması sonucu ortaya çıkan kompozit öncelik bazlı kuralları içermektedir. Bu kurallar farklı esnek atölye tipi çizelgeleme problem setlerinde ampirik olarak elde edilmiştir.[43] Aşağıda Kural1 ve Kural2 olarak gösterilen metotlar ile her bir işe ait sırada bekleyen operasyonun öncelik değerleri hesaplanır ve kurala göre öncelik değeri en büyük olan operasyon çizelgeye alınır.

Kompozit kurallar aşağıda gösterilen parametrelerden oluşmaktadır.

Kural1: Sipariş zamanı + Operasyonun makinede geçen işlem süresi + 2 *
işin ortalama toplam işlenme süresi + işin toplam operasyon sayısı

Kural2: Sipariş zamanı + Operasyonun makinede geçen işlem süresi + 2 *
işin ortalama toplam işlenme süresi

Kompozit kuralları etkileyen faktörlerden bir tanesi sipariş zamanı olarak gözükmektedir. Tüm işlerin sipariş zamanı sıfır olarak kabul edilerek öncelik bazlı çizelge değerleri hesaplanmıştır.



Şekil 4.3 Öncelik Bazlı Kuralların Performansı[43]

4.3 Çaprazlama

Çaprazlama ile iki bireyde bulunan farklı özelliklerinin birleşmesi sonucu yeni birey oluşması sağlanmaktadır. Kromozom haline çevrilmiş çözümler arasından iki birey seçilerek çaprazlama işlemini sokulurlar. Popülasyon havuzundan bireylerin seçilmesi ikili seçim yöntemi kullanılarak gerçekleştirilmektedir. Yapılan çalışmalar neticesinde eşleşme havuzuna seçilen bireylerin ikili seçim yöntemi ile gerçekleştirildiğinde daha iyi sonuçlar verdiği gözlemlenmiştir. [17]

Eşleşme havuzuna alınan bireylerden sıra bazlı çaprazlama yöntemi ile oğul bireyler oluşturulmaktadır. İlk olarak oğul bireyin kalıtım olarak esas alacağı ana birey seçilir. Rassal olarak belirlenen iki noktada arasındaki genler, çizelge parçaları yeni oluşacak bireye kopyalanır ve eksik kalan operasyonlar kopyalama işleminin yapılmadığı birey üzerindeki sıra göz önüne alınarak yeni oluşturulan bireye aktarılırlar. Yaptığımız çalışmada seçilen her iki bireyden yeni iki birey oluşturulmaktadır. Çaprazlama işlemi ana birey değiştirilerek bir kez daha tekrarlanır böylece iki yeni oğul birey oluşturulmuş olur.

Bundan sonraki işlemlerde aşağıdaki operasyon süreleri dikkate alınacaktır.

Çizelge 4.19 Operasyon İşlenme Süreleri

Makine No	İş No										
	Operasyon No										
	1			2			3		4		
	1	2	3	1	2	3	1	2	1	2	3
1	3	15	11		13	13	3	3	9	6	
2	18		24	11	8	11		5	27		7
3		12	9	11	7	23		14	32	2	4

Aşağıdaki örnekte bu işlemin nasıl yapıldığı detaylı olarak gösterilmektedir.

		Ebeveyn I										
Başlama-Bitiş		0-3	0-18	0-11	3-12	12-18	18-26	18-30	26-39	26-31	30-34	39-50
Operasyon Sırası		O31	O11	O21	O41	O42	O22	O12	O23	O32	O43	O13
Makine Ataması		1	2	3	1	1	2	3	1	2	3	1
		Ebeveyn II										
		O11	O31	O21	O41	O42	O22	O12	O23	O43	O13	O32
Oğul I		1	1	2	1	1	2	3	1	3	2	3
		Ebeveyn II										
Operasyon Sırası		O11	O41	O12	O31	O42	O21	O43	O13	O22	O23	O32
Makine Ataması		1	2	1	1	3	2	3	2	3	3	3
Başlama-Bitiş		0-3	0-27	3-18	18-21	27-29	27-38	29-33	38-62	38-45	45-68	68-82

Şekil 4.4 Kullanılan Çaprazlama İşlemi

4.4 Mutasyon

Mutasyon tek birey üzerinde değişiklik yapılarak popülasyondaki birey çeşitliliğinin artırılması için kullanılmaktadır. Arama uzayını lokal optimumlardan kurtarmanın yolu olarak değerlendirilir. Aşağıdaki mutasyon yöntemleri bu çalışmada kullanılmaktadır. Altı farklı mutasyon işlemi bulunmaktadır. İlk gruptaki mutasyon işlemleri her şeyin rassal olarak yapıldığı mutasyon işlemleri iken diğer grup en son biten operasyonun tamamlama süresini azaltmaya yönelik akıllı mutasyonlardır.

4.4.1 Rassal Makine Ataması

Kromozom üzerinde bulunan genler iş-operasyon-makine eşlemesi ile tutulmaktadır. Esnek atölye tipi üretim tipinde bir operasyon birden fazla makinede işlenebilmektedir. Bu mutasyonda rassal olarak bir operasyon seçilmekte. Eğer bu operasyonu birden farklı makinede işlenebiliyorsa yine rassal olarak bu makinelerden birisi seçilmekte ve operasyon yeni makinesine atanmaktadır. Yapılan yeni makine ataması sonrası kromozom aktif çizelgeye dönüştürülür ve tekrar popülasyon havuzuna bırakılır. Aşağıda bir örnek üzerinde bu işlem anlatılmıştır.

Popülasyon havuzundan rasgele bir birey ve bu birey üzerinde rasgele bir operasyon seçilir.

Çizelge 4.20 Rassal Makine Ataması - Birey ve Operasyon Seçilişi

Başlama-Bitiş	Ebeveyn I										
	0-3	0-18	0-11	3-12	12-18	18-26	18-30	26-39	26-31	30-34	39-50
Operasyon Sırası	O31	O11	O21	O41	O42	O22	O12	O23	O32	O43	O13
Makine Ataması	1	2	3	1	1	2	3	1	2	3	1

Aşağıda rassal olarak seçilebilecek makineler listelenmektedir.

Çizelge 4.21 Rassal Makine Ataması – Seçilen Operasyonu İşleyebilecek Makinelerde İşlenme Süreleri

Makine No	İş No	
	Operasyon No	
	3	
	1	2
1	3	3
2		5
3		14

Rassal olarak bir makine seçilir ve operasyon işlendiği makine değiştirilerek çizelge yeniden oluşturulur. Yeniden çizelgeleme esnasında O43 işlenmeye O13'ten daha geç başladığı için yerleri değiştirilmiştir.

Çizelge 4.22 Rassal Makine Ataması – İşlem Sonrası Kromozom

Başlama-Bitiş	0-3	0-18	0-11	3-12	12-18	18-26	18-30	26-39	30-44	39-50	44-48
Operasyon Sırası	O31	O11	O21	O41	O42	O22	O12	O23	O32	O13	O43
Makine Ataması	1	2	3	1	1	2	3	1	3	1	3

4.4.2 Operasyon Yer Değişikliği

Kromozom üzerinde rassal olarak seçilen bir operasyonun yeri değiştirilmektedir. Seçilen operasyonun yeni atanacağı yer yine rassal olarak belirlenmektedir. Ancak işe ait operasyonlar arasında önceliklendirme bulunduğundan, yer değişikliği seçilen operasyonun ile aynı işe ait kendisinden bir önceki operasyonun kromozomda bulunduğu yerden daha önce olamayacağı gibi, yine aynı işe ait kendisinden bir sonraki operasyonun kromozomda bulunduğu yerden sonra olamamaktadır. Yapılan değişiklik sonrası kromozom aktif çizelgeye dönüştürülür ve tekrar popülasyon havuzuna bırakılır.

Popülasyon havuzundan rasgele bir birey ve bu birey üzerinde rasgele bir operasyon seçilir. Operasyonlar arası öncelik şartı gereği seçilen operasyon kendisi ile aynı işe ait bir önceki ve bir sonraki operasyonlar arasında kalan bölgede yer değiştirebilmektedir.

Çizelge 4.23 Operasyon Yer Değişikliği - Birey ve Operasyon Seçilişi

Başlama-Bitiş	0-3	0-18	0-11	3-12	12-18	18-26	18-30	26-39	30-44	39-50	44-48
Operasyon Sırası	O31	O11	O21	O41	O42	O22	O12	O23	O32	O13	O43
Makine Ataması	1	2	3	1	1	2	3	1	3	1	3

Rassal olarak bahsedilen iki operasyon arasında bir nokta belirlenir. Operasyon buraya taşınır.

Çizelge 4.24 Operasyon Yer Değişikliği – İşlem Sonrası Kromozom

Başlama-Bitiş	0-3	0-18	0-11	3-12	12-26	12-18	18-26	26-38	26-39	39-50	39-43
Operasyon Sırası	O31	O11	O21	O41	O32	O42	O22	O12	O23	O13	O43
Makine Ataması	1	2	3	1	3	1	2	3	1	1	3

4.4.3 Son Operasyon Yerinin Değiştirilmesi

Bu çalışmada her çizelge için amaç fonksiyon değeri, üretimde en son biten operasyonun tamamlanma süresi olarak kabul edilmektedir. Amaç fonksiyonunu küçültebilmek için kritik yol üzerinde değişiklik yapılması gerekmektedir. Kritik yol son biten operasyonu etkileyen aynı ya da farklı makinedeki tüm operasyonları kapsamaktadır.

Bu mutasyonda en son biten operasyon seçilerek yer değişikliği yapılmaktadır. Operasyonun kromozom üzerinde yeni yerinin belirlenmesi rassal olarak yapılmaktadır. Yer seçimi aynı işe ait kendisinden bir önceki operasyon ile kromozomun uzunluğu arasında seçilmektedir. Yapılan değişiklik sonrası kromozom aktif çizelgeye dönüştürülerek tekrar popülasyon havuzuna bırakılır. Aşağıda bu işleme ait bir örnek görüntülenmektedir.

En son iş seçilir. Bu iş kendisinde bir önceki operasyondan (Kromozom üzerinde 4.sıra O31) daha önce yerleşmeyecek şekilde yeni bir yere atanır(8. sıra) .

Çizelge 4.25 Son Operasyon Yer Değişikliği - Birey ve Operasyon Seçilişi

Operasyon Sırası	O11	O41	O12	O31	O42	O21	O43	O13	O22	O23	O32
Makine Ataması	1	2	1	1	3	2	3	2	3	3	3
Başlama-Bitiş	0-3	0-27	3-18	18-21	27-29	27-38	29-33	38-62	38-45	45-68	68-82

Atama sonrası çizelge yeniden oluşturulur.

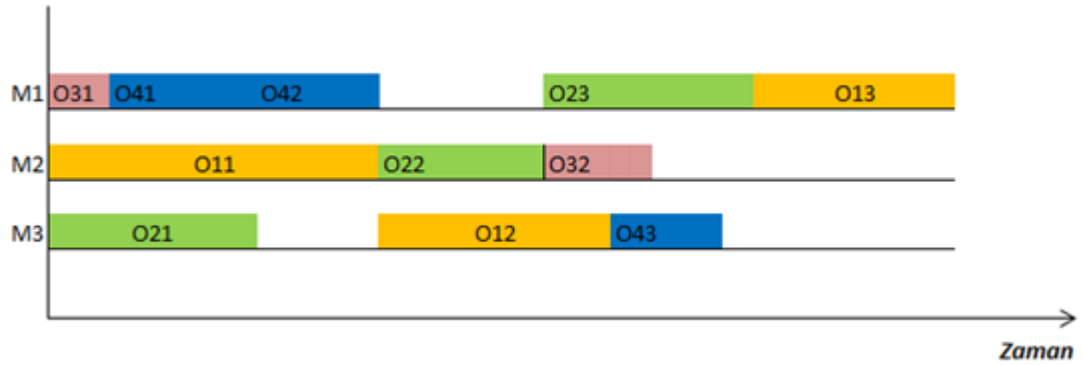
Çizelge 4.26 Son Operasyon Yer Değişikliği - İşlem Sonrası Kromozom

Operasyon Sırası	O11	O41	O12	O31	O42	O21	O43	O32	O13	O22	O23
Makine Ataması	1	2	1	1	3	2	3	3	2	3	3
Başlama-Bitiş	0-3	0-27	3-18	18-21	27-29	27-38	29-33	33-47	38-62	47-54	54-77

4.4.4 En Büyük Boşluğa Uygun Operasyon Yerleştirme

Çizelgeleme işlemi yapılırken aynı işe ait operasyonlardan sonraki operasyonun başlayabilmesi için, önceki operasyonun tamamlanmasının beklenmesi gerekmektedir. Bu, aynı makineye atanmış farklı işe ait birbirini takip eden iki operasyon arasındaki bekleme makine için ölü zamana sebep olabilmektedir. Bekleme süresini azaltmak için çizelge içindeki en uzun boşluk bulunarak, farklı bir makineye atanmış ve bu zaman aralığında başlayabilecek bir operasyon maksimum bekleme zamanının yer aldığı makineye atanır. Böylece makinenin boş beklemesi engellenerek en son işin tamamlanma süresinin kısaltılması öngörülmektedir.

Aşağıdaki örnekte en fazla ölü zaman makine1de O_{42} ile O_{23} numaralı operasyonlar arasında gözükmektedir.



Şekil 4.5 Kromozomun Gantt Şemasında Gösterimi

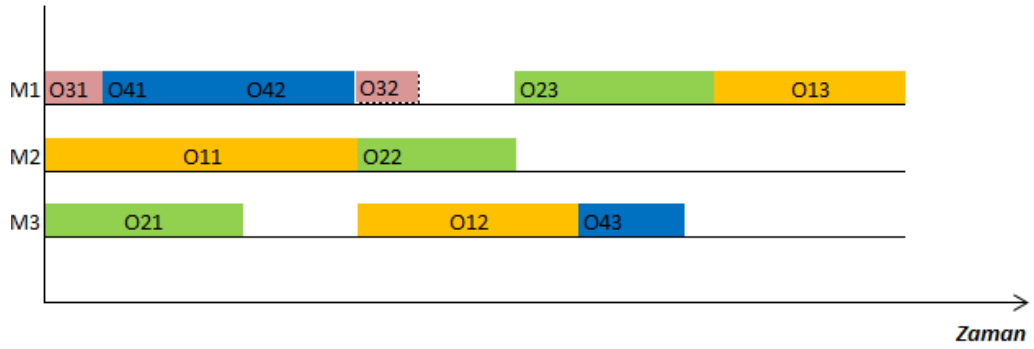
Çizelge 4.27 En Büyük Boşluğu Operasyon Yerleştirme - Seçilen Birey ve Operasyon

Operasyon Sırası	O31	O11	O21	O41	O42	O22	O12	O23	O32	O43	O13
Makine Ataması	1	2	3	1	1	2	3	1	2	3	1

Atama sonrası kromozom yapısı aşağıdaki şekilde olurken, gantt şemasında aşağıda görüntülenmektedir.

Çizelge 4.28 En Büyük Boşluğu Operasyon Yerleştirme - İşlem Sonrası Kromozom

Operasyon Sırası	O31	O11	O21	O41	O42	O32	O22	O12	O23	O43	O13
Makine Ataması	1	2	3	1	1	1	2	3	1	3	1



Şekil 4.6 İşlem Sonrası Kromozomun Gantt Şemasında Gösterimi

4.4.5 En Fazla Üretim Zaman Kazancı

İki operasyonun seçilip atanmış oldukları makineleri değiştirilerek kromozom üzerine mutasyon işlemi yapılır. Aslen en son tamamlanan operasyona ait makine bulunarak bu makine üzerinden herhangi bir operasyon seçilir. Aynı işe ait operasyonlar arasında önceliklendirme bulunmaktadır. Öncelik kurallarının çiğnenmemesi için kromozomda aynı işe ait bir sonraki operasyonun yeri belirlenir. Bu iki operasyon arasındaki kalan operasyonlar ile seçilmiş olan operasyonun makine numaraları değiştirilerek operasyon işleme sürelerindeki farka bakılır. İki operasyondaki değişikliğin işleme zamanı bakımından maksimum getiri sağladığı noktada, makine değişikliği gerçekleştirilir. Yapılan değişiklik sonrası kromozom aktif çizelgeye dönüştürülerek tekrar popülasyon havuzuna bırakılır.

Çizelge 4.29 En Fazla Üretim Zaman Kazancı Mutasyonu için Sözde Kod

```
En son tamamlanan operasyonun atandığı makine  $M_s$  bulunur.
Rassal olarak  $M_s$ 'ye atanmış operasyonlar arasında bir tanesi  $O_{is}$  seçilir.
// i operasyonun ait olduğu işi belirtmektedir
Kromozom üzerinde  $O_{is}$  bulunduğu yer belirlenir  $poz_{is}$ 
 $O_{is+1}$ 'in kromozom üzerindeki yeri bulunur  $poz_{is+1}$ 
FOR i =  $poz_{is+1}$  to  $poz_{is+1}$ 
    Kromozomda pozisyon i'de bulunan operasyon  $O_x$  bulunur.
     $O_x$  atanmış olduğu makine  $M_x$  ve işleme süresi  $P_x M_x$  bulunur
    IF  $M_s = M_x$ 
         $i \leftarrow i + 1$ 
    ELSEIF  $O_x$  işleyebilen makineler arasında  $M_s$  varsa &&  $O_{is}$ 
    işleyebildiği makineler arasında  $M_x$  varsa
        Kazanç  $\leftarrow (P_{is} M_x - P_{is} M_s) + (P_x M_s - P_x M_x)$ 
    IF Kazanç > MaxKazanç
        MaxKazanç  $\leftarrow$  Kazanç
         $O_c \leftarrow O_x$ 
    ELSE
         $i \leftarrow i + 1$ 
    ENDIF.
ENDFOR
 $O_c$  ile  $O_{is}$  ye ait makine atamaları değiştirilir.
```

4.4.6 En İyi Makineye Atama

Herhangi bir operasyon seçilerek, operasyonun işleme süresi bakımından en kısa olduğu makineye ataması yapılır. Yapılan değişiklik sonrası kromozom aktif çizelgeye dönüştürülerek tekrar popülasyon havuzuna bırakılır.

Çizelge 4.30 En İyi Makine Atama Mutasyonu için Sözde Kod

```
Kromozomdan rassal olarak bir operasyon  $O_{is}$  seçilir
 $O_{is}$  'nin atandığı makine bulunur  $M_s$ 
 $O_{is}$  'yi en kısa sürede işleme süresine sahip makine bulunur  $M_b$ 
IF  $M_b \neq M_s$ 
     $O_{is}$   $M_b$  makinası atanır.
    Kromozomom tekrar çizelgelenir.
ENDIF
```

4.5 Lokal Arama

GA işleminin ardından çıkan son nesil lokal bir arama algoritmasını sokularak daha iyi sonuçların elde edilmeye çalışılmaktadır. Çok fazla zaman kaybettirmeyen ve uygulaması kolay olan bir arama algoritması arama sürecinin sonuna eklenmiştir.

Lokal arama işlemi aynı makinede bulunan operasyonların yerleri değiştirilerek gerçekleştirilir. Kromozomda operasyonlar başlama zamanına göre sıralanmaktadır. Sıralamadaki son operasyondan başlanarak aynı işe ait bir önceki operasyon bulununcaya kadar aynı makinede işlenen operasyonların yerleri değiştirilir ve değişiklik sonrası kromozom aktif çizelge algoritmasına sokulur. Eğer daha iyi bir sonuç bulduysa çizelge yeni kromozom yapısına çevrilir ve arama işlemine en baştan başlanır.

Çizelge 4.31 Lokal Arama için Sözdde Kod

Kromozomdaki(krmz1) son operasyon “*krmz1.lastOp*” kromozomdaki sırası “*krmz1.lastOpSeq*” bağlı bulunduğu iş nosu “*lastOp.jobNo*” ve atandığı makine no “*lastOp.machNo*” belirlenir

opr ← *krmz1.lastOp*

Son operasyondan bir önceki operasyon seçilir “*krmz1.(lastOp-1)*”

prevOp ← *krmz1.(lastOp-1)*

prevOp bağlı bulunduğu iş nosu “*prevOp.jobNo*” belirlenir.

WHILE *opr* != *krmz1.firstOp*

FOR *i* = *krmz1.oprSeq* **TO** *opr* ile aynı işe ait bir önceki operasyona kadar

IF *opr.machNo* == *prevOp.machNo*

opr ↔ *prevOp* ile kromozom(*krmz2*) üzerinde yer değiştirir

CALL “AktifÇizelge”

IF *krmz2.uygunluk değeri* > *krmz1.uygunluk değeri*

krmz1 ← *krmz2*

opr ← *krmz1.lastOp*

prevOp ← *krmz1.(lastOp -1)*

ELSE

prevOp ← *krmz1.(prevOp-1)*

ENDIF.

ELSE

prevOp ← *krmz1.(prevOp-1)*

ENDIF.

opr ← *krmz1.(lastOp-1)*

ENDWHILE

5. ÖNERİLEN SEZGİSELİN UYGULANMASI VE PERFORMANS TESTİ

Geliştirilen algoritma ile en son işin tamamlanma zamanını (C_{max}) en küçüklemeyi amaçlamaktadır. Yukardaki anlatılan işlemler farklı parametreler verilerek çalıştırılmıştır. Brandimarte[12] ait problem seti kullanılmıştır. Bu sette 10 farklı problem bulunmaktadır, problemlerin büyüklükleri aşağıdaki tabloda verilmiştir. Kimi problem tam esnek bir yapıya sahipken kimisi kısmi esnektir. Tam esnek problemlerde operasyonların tamamı atölyede bulunan bütün makineler tarafından işlenebilmektedir. Kısmi esnek problem ise makinelerin işleyebildiği operasyonlar belirlidir. Operasyonları işleyebilen makineler değişiklik göstermektedir. Her problem belirlediğimiz parametreler ile beşer defa çalıştırılmış ve aşağıdaki sonuçlar elde edilmiştir. Nesil sonuna ulaşıldığında mevcut tüm bireyler üzerinde lokal arama işlemi gerçekleştirilmektedir.

Çizelge 5.1 Brandimarte Problem Seti Özellikleri[12]

Problem	İş X Makine Sayısı	Toplam Operasyon Sayısı	Bir Operasyonu İşleyebilen Ortalama Makine Sayısı
mk01	10 × 6	55	2,09
mk02	10 × 6	58	4,01
mk03	15 × 8	150	3,01
mk04	15 × 8	90	1,91
mk05	15 × 4	106	1,71
mk06	10 × 15	150	3,27
mk07	20 × 5	100	2,83
mk08	20 × 10	225	1,43
mk09	20 × 10	240	2,53
mk10	20 × 15	240	2,98

Parametreler

Tüm problemler için aşağıdaki parametreler sabit tutulmuştur.

Rassal Bozma(Perturbasyon) Oranı(PR): % 30

Seçim Kuralı: İkili Seçim yöntemi

Şeçkinlik: %2

Mutasyon Oranları:

Rassal makine ataması: 0,07

Operasyon yer deęişikliği: 0,03

Son operasyon yer deęiştirme: 0,01

En fazla üretim zamanı kazancı 0,04

En iyi makineye atama 0,05

En büyük boşluęa uygun operasyon yerleřtirme: 0,04

Ařaęıda, elde ettięimiz sonuçlar literatürde yapılmıř dięer çalıřmalar ile kıyaslanmaktadır. Kullanılan popülasyon büyüklüęü, nesil sayısı ve bařlangıç popülasyonu oluřturma sırasında kullanılan atama yöntemine göre farklı senaryolar oluřturulmuřtur

Toplamda üç farklı popülasyon-nesil sayısı çifti kullanılmıřtır. Bunlar (500,350), (750,500), (300,50) řeklinindedir. Bu üç çiftin her biri için makine atamaları iki farklı řekilde yapılmıřtır. Dolayısıyla $3 \times 2 = 6$ senaryo altında kořturmalar yapılmıřtır. Her senaryo için 5 kořturma yapıp en iyi deęerler raporlanmıřtır.

Aldıęımız en iyi sonuçlar literatürdeki dięer GA tabanlı meta-sezgisel algoritmalar ile kıyaslandıęında iyi çözümlerin elde edildięi gözlenmektedir. Her tabloda “S” bařlıklı kolonlar yapılan çalıřma sonucu elde edilmiř en iyi deęerleri göstermektedir. Ekte tüm kořturmalara ait sonuçlar bulunmaktadır.

Senaryo 1 (S1)

Popülasyon Sayısı: 500

Nesil Sayısı: 350

Başlangıç Popülasyonu Makine Ataması: “Approach by Localization”

Senaryo 2 (S2)

Popülasyon Sayısı: 500

Nesil Sayısı: 350

Başlangıç Popülasyonu Makine Ataması: : Önerilen Atama Yöntemi

Çizelge 5.2 – Senaryo1 ve Senaryo2 ile Literatür Sonuç Karşılaştırma

	Pezella vd. [17]	Tang vd. [19]	Al-Hinai [21]	S1	Fark %	S2	Fark %	En iyi sonuçlar	Alt Sınır
MK01	40	40	40	40	-	40	-	40	36
MK02	26	26	26	27	3,8	27	3,8	26	24
MK03	204	204	204	204	-	204	-	204	204
MK04	60	60	61	63	5	62	3,3	60	48
MK05	173	173	173	173	-	173	-	173	168
MK06	63	60	62	60	-	61	1,6	60	33
MK07	139	140	141	143	2,8	142	2,1	139	133
MK08	523	523	523	523	-	523	-	523	523
MK09	311	307	307	311	1,3	311	1,3	307	299
MK10	212	205	214	208	1,4	216	5,3	205	165
Pop. Sayısı	5000	500	1200	500		500			
Nesil sayısı	1000	1000	1000	350		350			

Senaryo 3 (S3)

Popülasyon Sayısı: 750

Nesil Sayısı: 500

Başlangıç Popülasyonu Makine Ataması: “Approach by Localization”

Senaryo 4(S4)

Popülasyon Sayısı: 750

Nesil Sayısı:500

Başlangıç Popülasyonu Makine Ataması: : Önerilen Atama Yöntemi

Çizelge 5.3 – Senaryo3 ve Senaryo4 ile Literatür Sonuç Karşılaştırma

	Pezella vd. [17]	Tang vd. [19]	Al-Hinai [21]	S3	Fark %	S4	Fark %	En iyi sonuçlar	Alt Sınır
MK01	40	40	40	40	-	40	-	40	36
MK02	26	26	26	27	3,8	26	-	26	24
MK03	204	204	204	204	-	204	-	204	204
MK04	60	60	61	65	8,3	62	3,3	60	48
MK05	173	173	173	173	-	173	-	173	168
MK06	63	60	62	61	1,6	61	1,6	60	33
MK07	139	140	141	141	1,4	142	2,1	139	133
MK08	523	523	523	523	-	523	-	523	523
MK09	311	307	307	311	1,3	307	-	307	299
MK10	212	205	214	209	1,9	212	3,4	205	165
Pop. Sayısı	5000	500	1200	750		750			
Nesil sayısı	1000	1000	1000	500		500			

Senaryo 5 (S5)

Popülasyon Sayısı: 300

Nesil Sayısı: 50

Başlangıç Popülasyonu Makine Ataması: “Approach by Localization”

Senaryo 6 (S6)

Popülasyon Sayısı: 300

Nesil Sayısı: 50

Başlangıç Popülasyonu Makine Ataması: : Önerilen Atama Yöntemi

Çizelge 5.4 – Senaryo5 ve Senaryo6 ile Literatür Sonuç Karşılaştırma

	Pezella vd. [17]	Tang vd. [19]	Al-Hinai [21]	S5	Fark %	S6	Fark %	En iyi sonuçlar	Alt Sınır
MK01	40	40	40	42	5	41	2,5	40	36
MK02	26	26	26	27	3,8	27	3,8	26	24
MK03	204	204	204	204	-	204	-	204	204
MK04	60	60	61	66	10	67	11,7	60	48
MK05	173	173	173	174	0,06	174	0,06	173	168
MK06	63	60	62	66	10	66	10	60	33
MK07	139	140	141	145	4,3	145	4,3	139	133
MK08	523	523	523	523	-	523	-	523	523
MK09	311	307	307	322	4,8	324	5,5	307	299
MK10	212	205	214	241	17,5	235	14,6	205	165
Pop. Sayısı	5000	500	1200	300		300			
Nesil sayısı	1000	1000	1000	50		50			

Elde ettiğimiz tüm farklı senaryoların en iyi sonuçlarının birbiri ile kıyaslaması aşağıdaki şekildedir.

Çizelge 5.5 –Tüm Senaryoları Birbiri ile Kıyaslama

	S1	S2	S3	S4	S5	S6	Alt Sınır
MK01	40	40	40	40	42	41	36
MK02	27	27	27	26	27	27	24
MK03	204	204	204	204	204	204	204
MK04	63	62	65	62	66	67	48
MK05	173	173	173	173	174	174	168
MK06	60	61	61	61	66	66	33
MK07	143	142	141	142	145	145	133
MK08	523	523	523	523	523	523	523
MK09	311	311	311	307	322	324	299
MK10	208	216	209	212	241	235	165
Pop. Sayısı	500	500	750	750	300	300	
Nesil sayısı	350	350	500	500	50	50	

Popülasyon sayısı ve nesil sayısı arama eforunun büyüklüğünü belirleyen iki ana faktördür. Yapılan çalışma literatürdeki çalışmalar ile kıyaslandığında daha küçük bir arama eforuyla arama yapılmış olmasına rağmen, yapılan çalışmalara benzer sonuçlar elde edilmiştir. Bunun sebebi iyi bir başlangıç popülasyonu ile başlanması ve farklı GA işlemleri kullanılarak arama işleminin gerçekleştirilmesidir.

Farklı senaryolar değerlendirildiğinde daha küçük arama eforuna sahip 500X350'lik senaryonun kimi problem tipleri için daha iyi sonuçlar verdiği gözlemlenmiştir. Bu, geliştirilen algorithmada kullanılan rassallıktan kaynaklanabileceği gibi, daha küçük bir popülasyonda iyi bireylerin çaprazlama şansının daha yüksek olmasından da kaynaklanıyor olabilir.

Çok daha küçük arama uzayına sahip 300X50'lik senaryoda sonuçlar diğer senaryolara göre daha kötü olmasına rağmen, çok daha hızlı zamanda sonuç elde

edilmiştir. Bu durum, çabuk çözümler elde edilmesi gereken gerçek dünya ortamında algoritmanın kullanılabilir bir hal almasını sağlamaktadır.

Önerilen atama yöntemi ile Kacem'in[13] önerdiği yöntem, kalan parametrelerin aynı tutulduğu durumda karşılaştırıldığında iki yöntemin birbirine benzer sonuçlar ürettiği görülmüştür, bu da önerilen yöntemin iyi performans sergilediğini göstermektedir.

Tüm senaryolar değerlendirildiğinde popülasyon, nesil sayısı (750,500) ve önerilen atama yöntemi kullanıldığı durumda en iyi sonuçlar elde edilmiştir.

Lokal Arama işlemi nesil sayısı küçüldüğünde amaç fonksiyonunda daha fazla etki göstermektedir. Küçük nesil sayısı, popülasyon havuzundaki bireylerin tam olarak olgunlaşmadan sonlanmasına sebep olabilmekte, bu da lokal arama yardımıyla daha fazla iyileşmenin gerçekleşmesine sebep olabilmektedir. 300X50'lik senaryoda çok sayıda ve büyük miktarda iyileşme sağlandığı görülmüştür. (Bkz Ekler)

6. SONUÇ

Bu çalışmada GA tabanlı hibrid bir algoritma geliştirilmiştir. GA kısmında farklı mutasyonlar operatörleri geliştirilerek, uygulanmıştır. GA çalışmasını tamamlandıktan sonra son üretilen nesil, kromozom üzerindeki operasyonların yer değiştirilerek daha iyi sonuçların arandığı lokal bir aramaya sokulmaktadır. Ayrıca GA için oluşturulan başlangıç popülasyonun üretilmesinde yeni bir yaklaşım sunulmuştur.

Başlangıç popülasyonu oluşturma aşamasında ilk iş makine atamaları yapılması ve farklı öncelik kuralına göre çizelgelenmeler oluşturulmasıdır, bu duruma literatürde sıkça rastlanılmaktadır ancak alınan işlere ait operasyonların işlenme zamanlarında oynamalar yapılarak başlangıç atamaları farklılaştırılmaktadır. Bu yöntemle elde edilen sonuçlar literatür ile değerlendirildiğinde iyi bir başlangıç kümesi oluşturulduğu görülmüştür. Ek olarak literatürde sıkça kullanılan makine yükünü dengeleyecek şekilde yapılan atamalar yönteminden esinlenerek uzun işlenme süresine sahip operasyonların sona kalmasını engellemek amacıyla makine ataması yapılırken bu tip işlere öncelik verilmesini sağlayan bir atama algoritması geliştirilmiştir.

Geliştirilen algoritmanın amaç fonksiyonu en son biten işin tamamlanma süresini en küçükmektir. Literatürde bulunan çalışmalar ile geliştirilen algoritma karşılaştırılmış ve yapılan çalışmanın literatürdeki birçok sonuçtan iyi sonuçlar elde ettiği gözlemlenmiştir. Brandimarte[12] ait çözülen tüm problemler için elde ettiğimiz sonuçların tamamı için daha iyi sonuçlar elde etmiş bir çalışma bulunmamaktadır. Bazı problemlerde geliştirdiğimiz çözüm literatürde bulunan çözümlerden daha iyi sonuçlar verirken, kimi problem için aynı sonuçlar elde edilmiş ve bir kısmında daha kötü sonuçlar bulunmuştur. Ancak geliştirilen algoritma literatürdeki çalışmalardan çok daha az hesaplama eforu gerektirmektedir.

İlerleyen çalışmalarda esnek atölye tipi çizelgeme probleminin varsayımları kaldırılarak gerçek dünyaya daha yakın problem için çalışma genişletilebilir. Gerçek dünyada üretim devam eden bir süreçtir ve makinelerin tamamının üzerinde iş

ataması yapılmadan bekliyor olması mümkün değildir. Ayrıca taleplerin tamamı başlangıç zamanında oluşmamakta, zaman içerisinde talepler alınmakta ve aralarında yüksek öneme sahip işler bulunmaktadır.

Üretim sadece makinelerin durumu ile alakalı değildir. Hammadde, iş gücü gibi üretimi etkileyen farklı ana etkenlerde bulunmaktadır. Planlama yapılırken bunlarında hesaplanması çözümün gerçek dünyaya daha uygun hale gelmesini sağlayacaktır.

Bu çalışmada tek bir amaç fonksiyonunu ele alınmıştır, en son biten işin tamamlanma süresi. Birçok üretimde ana amaç son işin bitme süresinin yanında işlerin zamanında teslim edilmesini de içermektedir. Bunun dışında üretim şekline göre farklılık gösteren amaçlar bulunmaktadır. Bunları dikkate alan farklı çalışmalar gerçekleştirilebilir.

KAYNAKLAR

- [1] Pinedo, M. L., Scheduling: Theory, Algorithms, and Systems, *Springer*, 2008.
- [2] “The Legacy of Taylor, Gantt, and Johnson: How to Improve Production Scheduling” erişim adresi: http://drum.lib.umd.edu/bitstream/1903/7488/4/25813_cov.pdf, erişim tarihi 20 Mart 2013.
- [3]“Production Scheduling, Definition and History” erişim adresi: <http://www.scmfocus.com/productionplanningandscheduling/2010/12/05/production-scheduling-background/>, erişim tarihi: 27 Mart 2013.
- [4] Garey, M. R., Johnson D. S., Sethi ,R., The Complexity of Flowshop and Jobshop Scheduling, *Mathematics of Operations Research*, 1(2), 117-129, 1976.
- [5] Muth, J.F., Thompson, G.L., *Industrial Scheduling*, *Prentice-Hall*, Engle-wood Cliffs, N.J., 1963.
- [6] Yahyaoui, A., Fnaiech, N., Fnaiech, F., A Suitable Initialization Procedure for Speeding a Neural Network Job-Shop Scheduling , *IEEE Transactions On Industrial Electronics*,. 58(3), 1052 – 1060, 2011.
- [7] Metaxiotis, K. S., Ergazakis K., Psarras J. E., An Elaborate Analysis of Production Systems in Industry: What A Consultant Should Know, *Industrial Management & Data Systems*, 101(4), 185 – 193, 2001.
- [8] Gao, J., Gen, M., Sun, L .,Zhao X. , A Hybrid of Genetic Algorithm and Bottleneck Shifting For Multiobjective Flexible Job Shop Scheduling Problems, *Computers & Industrial Engineering*, 53(1), 149-162, 2007.

- [9] Fattahi, P., Mehrabad, M. S., Jolai, F., Mathematical Modeling and Heuristic Approaches to Flexible Job Shop Scheduling Problems, *Journal of Intelligent Manufacturing*, 18, 331-342, 2007.
- [10] Gen M., Cheng R., Genetic Algorithms and Engineering Design, *Wiley-Interscience*, 1997.
- [11] Baker, K.R., Introduction to Sequencing and Scheduling, *John Wiley*, NY, 1974.
- [12] Brandimarte, P. , Routing and Scheduling in a Flexible Job Shop by Tabu Search, *Annals of Operations Research*, 41(3) , 157-183,1993.
- [13] Kacem, I., Hammadi, S., Borne, P., Approach by Localization and Multiobjective Evolutionary Optimization For Flexible Job-Shop Scheduling Problems, *Systems, Man, and Cybernetics, Part C: Applications and Reviews*, IEEE Transactions on, 32, 1-13, 2002.
- [14] Ho, N.B. , Tay, J.C., GENACE: an Efficient Cultural Algorithm For Solving The Flexible Job-Shop Problem, Congress On Evolutionary Computation CEC2004, Portland, Oregon, USA, Haziran 2004.
- [15] Ho, N.B., Tay, J. C., Edmund, M. ,Lai, K., An Effective Architecture For Learning and Evolving Flexible Job-Shop Schedules, *European Journal of Operational Research*, 179(2), 316-333, 2007.
- [16] Gao, J., Sun, L ., Gen, M., A Hybrid Genetic and Variable Neighborhood Descent Algorithm For Flexible Job Shop Scheduling Problems, *Computers & Operations Research*, 35, 2892–2907, 2008.
- [17] Pezzella, F., Morganti, G., Ciaschetti G., A Genetic Algorithm For The Flexible Job-shop Scheduling Problem, *Computers & Operations Research*, 35(10), 3202–3212, 2008.

- [18] Zhang, G., Gao L., Shi, Y., A Genetic Algorithm and Tabu Search for Multi Objective Flexible Job Shop Scheduling Problems, International Conference on Computing, Control and Industrial Engineering CCIE2010, 251-254 Wuhan, China, Haziran 2010.
- [19] Tang, J., Zhang, G., Lin, B., Zhang B., A Hybrid Algorithm For Flexible Job-Shop Scheduling Problem, *Procedia Engineering*, 15, 3678–3683 , 2011.
- [20] Zhang, G. H., Shi, Y., Gao, L., A Genetic Algorithm and Tabu Search for Solving Flexible Job Shop Schedules, *Proc. IEEE Symp. Computational Intelligence and Design, IEEE Press*, 369-372, 2008.
- [21] Nasr, A. H., ElMekkawy, T., An Efficient Hybridized Genetic Algorithm Architecture For The Flexible Job Shop Scheduling Problem, *Flexible Services & Manufacturing Journal*, 23(1), 64-85, 2011.
- [22] Zhang G., Gao,L., Shi Y., An Effective Genetic Algorithm For The Flexible Job-Shop Scheduling Problem, *Expert Systems with Applications*, 38(4), 3563-3573, 2011.
- [23] Girish, B.S., Jawahar, N.,A Particle Swarm Optimization Algorithm For Flexible Job Shop Scheduling Problem, *IEEE International Conference on Automation Science and Engineering, CASE 2009, Bangalore, India, Ağustos 2009*.
- [24] Xia, W., Wu, Z., An Effective Hybrid Optimization Approach For Multi-Objective Flexible Job-Shop Scheduling Problems, *Computers and Industrial Engineering Journal*, 48, 409-425, 2005.
- [25] Aydemir, E., 2009, Atölye Tipi Çizelgeleme Problemlerinin Öncelik Kuralı Tabanlı Genetik Algoritma Yaklaşımıyla Simülasyon Destekli Optimizasyonu, *Yüksek Lisans Tezi, Süleyman Demirel Üniversitesi, Fen Bilimleri Enstitüsü, Isparta*.

- [26] Xhafa, F., Abraham, A., *Metaheuristics for Scheduling in Industrial and Manufacturing Applications*, Springer, 2008.
- [27] Beasley, D., Bull, D. R., Martin, R. R., An Overview of Genetic Algorithms: Part 1, Fundamentals, *University Computing*, 15(2):58-69, 1993.
- [28] Holland, J. H., *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975.
- [29] Sivanandam, S.N. , Deepa, S.N., *Introduction to Genetic Algorithms*, Springer, 2008.
- [30] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley Longman Publishing Co., Boston, MA, 1989.
- [31] Goldberg, D., Sastry, K., *Genetic Algorithms: The Design of Innovation*, Springer, 2010.
- [32] Goldberg, D. E., *The design of innovation: Lessons From and For Competent Genetic Algorithms*, Kluwer Academic Publishers, MA, Boston, 2002.
- [33] Syswera, G., *Uniform Crossover in Genetic Algorithms*, Morgan Kaufmann, San Francisco, CA, 1989.
- [34] Spears, W. M., De Jong, K. A., On The Virtues of Parameterized Uni-Form Crossover, Proc. 4th Int. Conf. on Genetic Algorithms, Morgan Kaufmann, San Mateo, CA, 1994.
- [35] Davis, L., Applying Algorithms to Epistatic Domains, Proc. Int. Joint Conf. on Artificial Intelligence, 162–164, 1985.

- [36] Goldberg, D. E., Lingle, R., Alleles, loci, and the TSP, Proc. 1st Int. Conf. on Genetic Algorithms, 154–159, 1985.
- [37] Oliver, J. M., Smith, D. J., Holland, J. R. C., A Study of Permutation Crossover Operators On The Travelling Salesman Problem, Proc. 2nd Int. Conf. on Genetic Algorithms, 224–230, 1987.
- [38] Espinoza, F.P., Minsker, B.S., Goldberg, D.E., Adaptive Hybrid Genetic Algorithm For Groundwater Remediation Design. Journal of Water Resources Planning and Management, 121(1), 14-24, 2005.
- [39] Kim, J. L., Permutation-Based Elitist Genetic Algorithm Using Serial Scheme For Large-Sized Resource-Constrained Project Scheduling, Proceedings of the 2007 Winter Simulation Conference, 2112-2118, 2007.
- [40] Negnevitsky, M., Artificial Intelligence: A Guide to Intelligent Systems, *Pearson Education Limited*, 2006.
- [41] Storer, R. H., Wu, S. D., Vaccari, R., New Search Spaces for Sequencing Problems with Application to Job Shop Scheduling Management Science, 38, 1495-1509, 1992.
- [42] Mesghouni, K., Hammadi, S., Borne, P., Evolution Programs For Job-Shop Scheduling, Proceedings of the IEEE International Conference on Computational Cybernetics and Simulation 1, 720–725, 1997.
- [43] Tay, J.C., Ho, N.B., Evolving Dispatching Rules Using Genetic Programming For Solving Multi-Objective Flexible Job-Shop Problems, Computers & Industrial Engineering, 54(3), 453-473, Nisan 2008.

EKLER

Senaryo 1 (S1)

Popülasyon Sayısı: 500

Nesil Sayısı: 350

Başlangıç Popülasyonu Makine Ataması : “Approach by Localization”

Çizelge A.1 – Senaryo 1 ile Alınan Sonuçlar

	R1	R2	R3	R4	R5	Ortalama
MK01	40	40	40	40	40	40
MK02	27	27	27	27	28	27,2
MK03	204	204	204	204	204	204
MK04	63	65	65	65	65	64,6
MK05	173	173	173	173	173	173
MK06	60	62	62	65	66	63
MK07	143	144	144	144	144	143,8
MK08	523	523	523	523	523	523
MK09	311	311	311	311	311	311
MK10	208	210	215	215	218	213,2

Lokal arama sonrası en iyi çözümlerde aşağıdaki şekilde iyileşmeler gerçekleşmiştir.

Çizelge A.2 – Senaryo 1’de Lokal Arama ile Sağlanan İyileşmeler

	LA Öncesi	LA sonrası
MK07	144	143
MK09	312	311
MK10	221	218
	209	208
	216	215

Senaryo 2 (S2)

Popülasyon Sayısı: 500

Nesil Sayısı: 350

Başlangıç Popülasyonu Makine Ataması: Önerilen Yöntem

Çizelge A.3 – Senaryo 2 ile Alınan Sonuçlar

	R1	R2	R3	R4	R5	Ortalama
MK01	40	40	40	40	40	40
MK02	27	27	27	27	28	27,2
MK03	204	204	204	204	204	204
MK04	62	63	65	65	67	64,4
MK05	173	173	173	173	173	173
MK06	61	62	64	66	66	63,8
MK07	142	142	143	144	144	143
MK08	523	523	523	523	523	523
MK09	311	312	312	312	313	312
MK10	216	219	220	221	221	219,4

Lokal arama sonrası en iyi çözümlerde aşağıdaki şekilde iyileşmeler gerçekleşmiştir.

Çizelge A.4 – Senaryo 2’de Lokal Arama ile Sağlanan İyileşmeler

	LA Öncesi	LA sonrası
MK04	65	63
MK10	220	219
	224	219

Senaryo 3 (S3)

Popülasyon Sayısı: 750

Nesil Sayısı:500

Başlangıç Popülasyonu Makine Ataması : “Approach by Localization”

Çizelge A.5 – Senaryo 3 ile Alınan Sonuçlar

	R1	R2	R3	R4	R5	Ortalama
MK01	40	40	40	40	40	40
MK02	27	27	28	29	29	28
MK03	204	204	204	204	204	204
MK04	65	65	65	65	65	65
MK05	173	173	173	173	173	173
MK06	61	61	62	62	63	61,8
MK07	141	141	144	144	144	142,8
MK08	523	523	523	523	523	523
MK09	311	311	311	311	312	311,2
MK10	209	210	211	213	215	211,6

Lokal arama sonrası en iyi çözümlerde aşağıdaki şekilde iyileşmeler gerçekleşmiştir.

Çizelge A.6 – Senaryo 3’de Lokal Arama ile Sağlanan İyileşmeler

	LA Öncesi	LA sonrası
MK07	142	141
MK10	216	215

Senaryo 4 (S4)

Popülasyon Sayısı: 750

Nesil Sayısı: 500

Başlangıç Popülasyonu Makine Ataması: Önerilen Yöntem

Çizelge A.7 – Senaryo 4 ile Alınan Sonuçlar

	R1	R2	R3	R4	R5	Ortalama	Alt Sınır
MK01	40	40	40	40	40	40	36
MK02	26	27	27	28	28	27,2	24
MK03	204	204	204	204	204	204	204
MK04	62	65	65	65	66	64,6	48
MK05	173	173	173	173	173	173	168
MK06	61	61	62	63	64	62,2	33
MK07	142	144	144	144	145	143,8	133
MK08	523	523	523	523	523	523	523
MK09	307	311	313	314	315	312	299
MK10	212	213	214	215	218	214,4	165

Lokal arama sonrası en iyi çözümlerde aşağıdaki şekilde iyileşmeler gerçekleşmiştir.

Çizelge A.8 – Senaryo 4’de Lokal Arama ile Sağlanan İyileşmeler

	LA Öncesi	LA sonrası
MK02	28	27
MK06	62	61
MK07	143	142
MK09	315	313
MK10	219	218

Senaryo 5 (S5)

Popülasyon Sayısı: 300

Nesil Sayısı: 50

Başlangıç Popülasyonu Makine Ataması: “Approach by Localization”

Çizelge A.9 – Senaryo 5 ile Alınan Sonuçlar

	R1	R2	R3	R4	R5	Ortalama
MK01	42	42	42	42	42	42
MK02	27	28	28	28	28	27,8
MK03	204	204	204	204	204	204
MK04	66	67	68	68	68	67,4
MK05	174	174	176	176	177	175,4
MK06	66	67	67	69	71	68
MK07	145	145	148	148	149	147
MK08	523	523	523	523	523	523
MK09	322	324	327	328	329	326
MK10	241	241	242	243	244	242,4

Lokal arama sonrası en iyi çözümlerde aşağıdaki şekilde iyileşmeler gerçekleşmiştir

Çizelge A.10 – Senaryo 5’de Lokal Arama ile Sağlanan İyileşmeler

	LA Öncesi	LA sonrası
MK02	29	28
	28	27
	29	28
MK04	68	67
	69	68
	68	67
MK05	176	174
	177	174
	178	176
MK06	71	67
	71	69
	68	66
	69	67

	LA Öncesi	LA sonrası
MK07	148	145
	153	148
	151	148
	148	145
	152	149
MK09	332	324
	337	328
	340	327
	335	329
MK10	331	322
	253	241
	253	243
	250	241
	250	242
	252	244

Senaryo 6 (S6)

Popülasyon Sayısı: 300

Nesil Sayısı: 50

Başlangıç Popülasyonu Makine Ataması: Önerilen Yöntem

Çizelge A.11 – Senaryo 6 ile Alınan Sonuçlar

	R1	R2	R3	R4	R5	Ortalama
MK01	41	41	41	42	42	41,4
MK02	27	28	28	28	29	28
MK03	204	204	204	204	204	204
MK04	67	67	67	67	68	67,2
MK05	174	174	176	177	177	175,6
MK06	66	66	69	70	70	68,2
MK07	145	145	146	147	149	146,4
MK08	523	523	523	523	523	523
MK09	324	326	328	329	331	327,6
MK10	235	246	248	248	249	245,2

Lokal arama sonrası en iyi çözümlerde aşağıdaki şekilde iyileşmeler gerçekleşmiştir

Çizelge A.12 – Senaryo 6’da Lokal Arama ile Sağlanan İyileşmeler

	LA Öncesi	LA sonrası
MK02	29	28
	29	28
	28	27
MK04	68	67
MK05	178	177
	176	174
	175	174
MK06	68	66
	70	69
	69	66
MK07	148	146
	147	145
	151	149
	150	147
	146	145

	LA Öncesi	LA sonrası
MK09	337	331
	343	329
	336	326
	338	324
	339	328
MK10	254	246
	239	235
	259	248
	257	248
	261	249

ÖZGEÇMİŞ

Kişisel Bilgiler

Soyadı, adı : Meto, Bilal
Uyruğu : T.C.
Doğum tarihi ve yeri : 24.02.1984 Antakya
Medeni hali : Bekar
Telefon : 0 (312) 555 86 06
Faks : 0 (312) 292 40 91
e-mail : bmeto@etu.edu.tr

Eğitim

Derece	Eğitim Birimi	Mezuniyet Tarihi
Lisans	ODTÜ/Endüstri Müh. Bölümü	2007

İş Deneyimi

Yıl	Yer	Görev
2007-2010	TOBB ETÜ	Araştırma Görevlisi
2010-	Türk Telekom	SAP Danışmanı

Yabancı Dil

İngilizce