

**T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ÇOK KAYNAK KISITLI PROJELERİN SEZGİSEL YÖNTEMLERLE
ÇİZELGELENMESİ**

UĞUR SATIÇ

**YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI
SİSTEM MÜHENDİSLİĞİ PROGRAMI**

**DANIŞMAN
YRD. DOÇ. DR. FAHRETTİN ELDEMİR**

İSTANBUL, 2014

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**ÇOK KAYNAK KISITLI PROJELERİN SEZGİSEL YÖNTEMLERLE
ÇİZELGELENMESİ**

Uğur SATIÇ tarafından hazırlanan tez çalışması 25.02.2014 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri Enstitüsü Endüstri Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Yrd. Doç. Dr. Fahrettin ELDEMİR
Yıldız Teknik Üniversitesi

Jüri Üyeleri

Yrd. Doç. Dr. Fahrettin ELDEMİR
Yıldız Teknik Üniversitesi

Prof.Dr. Coşkun ÖZKAN
Yıldız Teknik Üniversitesi

Prof. Dr. Osman KULAK
Pamukkale Üniversitesi

ÖNSÖZ

Bu çalışmada kaynak kısıtlı proje çizelgeleme probleminin çözümü için genetik algoritma metodolojisi kullanılarak hazırlanan Windows tabanlı bir bilgisayar uygulaması geliştirilmiştir. Geliştirilen uygulama PSPLIB ve MPSPLIB Kütüphanesi verileri ile sınanmış ve aynı kütüphaneleri kullanarak hazırlanan diğer çalışmalar ile kıyaslanmıştır.

Bu projeyi bana veren ve çalışmanın tamamlanmasında yakın ilgi ve desteğini esirgemeyen saygıdeğer öğretmenim ve danışmanım Yrd. Doç. Dr. Fahrettin ELDEMİR'e teşekkürlerimi sunmayı bir borç bilirim.

Şubat, 2014

Uğur SATIÇ

İÇİNDEKİLER

	Sayfa
SİMGE LİSTESİ.....	vii
KISALTMA LİSTESİ.....	viii
ŞEKİL LİSTESİ.....	ix
ÇİZELGE LİSTESİ	x
ÖZET	xi
ABSTRACT.....	xii
BÖLÜM 1	
GİRİŞ.....	1
1.1 Literatür Özeti	2
1.2 Tezin Amacı	2
1.3 Hipotez	3
BÖLÜM 2	
LİTERATÜR ARAŞTIRMASI	4
2.1 Kaynak Kısıtlı Proje Çizelgeleme Problemi.....	4
2.2 Kaynak Kısıtlı Proje Çizelgeleme Probleminin Alt Türleri.....	6
2.2.1 Çok modlu kaynak kısıtlı proje çizelgeleme problemi	6
2.2.2 Kaynak kısıtlı çok proje çizelgeleme problemi.....	8
2.2.3 Çok modlu kaynak kısıtlı çok projeli çizelgeleme problemi.....	9
2.2.4 Kaynak yatırım problemi	10
2.2.5 Diğer çizelgeleme problemleri.....	11
2.2.5.1 Üretim çizelgeleme problemi.....	11
2.2.5.2 İş çizelgeleme problemi (atölye çizelgeleme)	11

BÖLÜM 3

KAYNAK KISITLI PROJE ÇİZELGELEME PROBLEMİ.....	15
3.1 Kaynak Kısıtlı Proje Çizelgeleme Problemi.....	15
3.1.1 Kaynak kısıtlı proje çizelgeleme probleminin modeli.....	16
3.2 Çok Modlu Kaynak Kısıtlı Proje Çizelgeleme Problemi	18
3.2.1 Çok modlu kaynak kısıtlı proje çizelgeleme probleminin modeli.....	18
3.3 Çok Projeli Kaynak Kısıtlı Proje Çizelgeleme Problemi.....	20
3.3.1 Çok projeli kaynak kısıtlı proje çizelgeleme probleminin modeli.....	21

BÖLÜM 4

KAYNAK KISITLI PROJE ÇİZELGELEME PROBLEMLERİ İÇİN KULLANILAN ALGORİTMALAR	23
4.1 Deterministik Yöntemler.....	23
4.2 Stokastik Yöntemler	24
4.3 Sezgisel Yöntemler	24
4.3.1 Çizelge yaratma yapıları.....	24
4.3.1.1 Seri çizelge yaratma yapısı	25
4.3.1.2 Paralel çizelge yaratma yapısı	25
4.3.2 Öncelik tabanlı sezgisel algoritmalar	26
4.3.2.1 Tek geçişli algoritmalar	27
4.3.2.2 Çok geçişli algoritmalar	27
Öncelik kuralları	27
Örnekleme	27
Başlangıç-bitiş çıkışlı çizelgeleme.....	28
4.3.3 Metasezgisel yaklaşımlar	28
4.3.3.1 Benzetilmiş tavlama (simulated annealing)	28
4.3.3.2 Tabu arama	28
4.3.3.3 Genetik algoritmalar	29
4.3.3.4 Karınca kolonisi algoritması	29
4.3.3.5 Dağınık arama (scatter search) algoritması	30
4.3.3.6 Diğer sezgisel yöntemler	30
4.3.3.7 Meta-sezgisel algoritma sonuçlarının gösterilmesi	31

BÖLÜM 5

GENETİK ALGORİTMA	33
5.1 Genetik Algoritma	33
5.2 Genetik Algoritma Temel Kavramları.....	34
5.2.1 Kromozom	34
5.2.2 Gen.....	35
5.2.3 Popülasyon	35
5.2.4 Jenerasyon.....	35
5.2.5 Uygunluk fonksiyonu	36
5.2.6 Elitist seçim	36
5.2.7 Çaprazlama	36

5.2.8	Mutasyon.....	37
BÖLÜM 6		
KAYNAK KISITLI PROJE ÇİZELGELEME PROBLEMLERİ İÇİN GELİŞTİRİLEN GENETİK ALGORİTMA		39
6.1	Uygulamanın Varsayımları	39
6.1.1	Proje varsayımları	39
6.1.2	Genetik algoritma varsayımları	40
6.1.2.1	Başlangıç toplumu	40
6.1.2.2	Kromozom yapısı.....	41
6.1.2.3	İlk popülasyonun yaratılması	41
6.1.2.4	Uygunluk fonksiyonu.....	42
6.1.3	Projede kullanılan genetik işlemler	43
6.1.3.1	Çaprazlama operasyonu.....	44
6.1.3.2	Mutasyon operasyonu	46
6.2	Uygulamanın Veri Yapısı	47
6.3	Uygulamanın Veri Giriş Yapısı	48
6.3.1	Yeni bir proje bilginin girilmesi	48
6.3.2	Mevcut bir proje bilgi kaydının açılması	51
6.4	Uygulamanın Genetik Algoritma Ayarları	52
6.5	Uygulamanın Sonuç Rapor Bilgileri.....	53
6.6	Algoritma Performansının Değerlendirilmesi	58
6.6.1	Algoritma performansını değerlendirme yolları.....	58
6.6.2	Kaynak kısıtlı proje çizelgeleme problemleri kütüphanesi (PSPLİB)..	59
6.6.3	Algoritma performansının değerlendirilmesi	60
6.6.3.1	Algoritmanın J30 grubu problemlerle test edilmesi	61
6.6.3.2	Algoritmanın J60 grubu problemlerle test edilmesi	62
6.6.3.3	Algoritmanın J120 grubu problemlerle test edilmesi	63
6.6.3.4	Algoritmanın çok projeli problemler ile test edilmesi	64
BÖLÜM 7		
SONUÇ VE ÖNERİLER		69
KAYNAKLAR.....		71
ÖZGEÇMİŞ.....		77

SİMGE LİSTESİ

t	zaman indisi ($t=1,...,T$),
j	faaliyet indisi ($j=1,...,J$),
q	proje indisi ($q=1,...,Q$),
d_j	"j" faaliyetinin süresi
P_j	"j" faaliyetinin öncüllerinin kümesi,
EFT_j	"j" faaliyetinin en erken bitiş zamanı,
LFT_j	"j" faaliyetinin en geç başlangıç zamanı,
R	yenilenebilir kaynaklar kümesi,
k_{jr}	j faaliyetinin r kaynağından birim zamanda kullanım miktarı,
K_r	r yenilenebilir kaynağının birim zamanda kullanım üst sınırı,
M_j	"j" faaliyetinin mod adedi,
m	mod indisi ($m = 1,...,M_j$),
q	proje indisi ($q=1,...,Q$),
S_g	Çizelgelenmiş faaliyetler kümesi
D_g	Çizelgelemeye uygun faaliyetler kümesi
C_g	Çizelgelenmiş faaliyetler kümesi
A_g	Gerekli öncelikli faaliyeti tamamlanmış faaliyetler çizelgelemeye aday faaliyetler kümesi
t_g	Erken çizelgelenebilecek faaliyetin süresi
$F_{süre}$	Faaliyetin süre uygunluğu
t_{min}	kritik yol metodu ile bulunan süre
t	n. Kromozomum proje tamamlanma süresidir.
$P_{gecikme}$	Ortalama proje gecikmesi
N	Proje sayısı
$P_{i,bitiş}$	i'inci projenin bitiş
$P_{i,başlangıç}$	i'inci projenin başlangıcı
$F_{gecikme}$	Projenin ortalama gecikme uygunluğu
P_{min}	kritik yol metoduna göre gecikmeler

KISALTMA LİSTESİ

APD	Ortalama proje gecikmesi
CBR-SAR	Vaka tabanlı muhakeme ve geliştirilmiş adaptif arama
GA	Genetik algoritma
GRD	En büyük kaynak talebi
GRPW	En büyük konumsal ağırlık
LFT	En geç bitirme zamanı
LST	En geç başlama zamanı
MM	Multi-mod
MMKKPÇP	Çok modlu kaynak kısıtlı proje çizelgeme problemi
MILP	Tam sayılı lineer programlama
MSLK	En kısa boşluk süresi
MTS	En çok sayıda toplam ardıl
NP	Deterministik olmayan polinomsal zamanlı
PROGEN	Proje çizelgeme problemi yaratma programı
PSGS	Paralel çizelge yaratma yapısı
PSPLIB	Proje çizelgeme problemi kütüphanesi
KKPÇP	Kaynak kısıtlı Proje Çizelgeme Problemi
KKÇPÇP	Kaynak kısıtlı çok projeli çizelgeme problemi
PM	Feromon matrisi
RSM	Kaynak çizelgeme metodu
SA	Benzetilmiş tavlama(Simulated Annealing)
SGS	Çizelge yaratma yapıları
SPT	En kısa işlem süresi
SSGS	Seri çizelge yaratma yapısı
TMS	Toplam tamamlanma süresi
TS	Tabu arama
WCS	En kötü alternatif boşluk

ŞEKİL LİSTESİ

	Sayfa
Şekil 3.1 Bir kaynak kısıtlı proje çizelgeleme problemi	17
Şekil 3.2 Kaynak kısıtlı çok proje çizelgeleme problemi.....	22
Şekil 6.1 Genetik işlemler akış şeması	43
Şekil 6.2 Çaprazlama için seçilen iki ebeveyn kromozom	45
Şekil 6.3 Sıralı çaprazlama operasyonu örneği	45
Şekil 6.4 Basit bir mutasyon operasyonu örneği	46
Şekil 6.5 Projede kullanılan faaliyet ve kaynak yapısı	47
Şekil 6.6 Genetik algoritma proje çizelgeleyici ana ekranı	49
Şekil 6.7 Genetik algoritma proje çizelgeleyici kaynak bilgisi ekranı.....	49
Şekil 6.8 Genetik algoritma proje çizelgeleyici kaynak bilgisi girildi ekranı.....	50
Şekil 6.9 Genetik algoritma proje çizelgeleyici proje bilgisi ekranı.....	51
Şekil 6.10 Genetik algoritma proje çizelgeleyici faaliyet aç bölümü.....	52
Şekil 6.11 Çizelgelemeye hazır ana ekran	52
Şekil 6.12 Rapor ekranı	54
Şekil 6.13 Excel rapor sayfası	55
Şekil 5.14 Excel’de gant şeması	55
Şekil 6.15 Excel’de kaynak kullanımı grafiği.....	56
Şekil 6.16 Excel’de yerel en iyi değerleri.....	56
Şekil 6.17 Excel’de çoklu proje sekmesi.....	57
Şekil 6.18 Excel’de çoklu proje gant şemaları	58

ÇİZELGE LİSTESİ

	Sayfa
Çizelge 3.1 Çok modlu faaliyet bilgileri	20
Çizelge 6.1 Farklı çaprazlama ve mutasyon yöntemlerinin karşılaştırılması	44
Çizelge 6.2 Çeşitli mutasyon oranlarının denenmesi	46
Çizelge 6.3 Sezgisel yöntemlerin performanslarının karşılaştırılması (j30 kümesi)	62
Çizelge 6.4 Sezgisel yöntemlerin performanslarının karşılaştırılması (j60 kümesi)	63
Çizelge 6.5 Sezgisel yöntemlerin performanslarının karşılaştırılması (j120 kümesi)	64
Çizelge 6.6 Çok projeli problem örnekleri	65
Çizelge 6.7 Sezgisel yöntemlerin performansının kıyaslanması (çok projeli)	67
Çizelge 6.8 Sezgisel yöntemlerin performansının kıyaslanması (çok projeli)	68

ÇOK KAYNAK KISITLI PROJELERİN SEZGİSEL YÖNTEMLERLE ÇİZELGELENMESİ

Uğur SATIÇ

Endüstri Mühendisliği Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Yrd. Doç. Dr. Fahrettin ELDEMİR

Kaynak kısıtlı proje çizelgeleme problemleri için birçok algoritma geliştirilmiştir. Buna rağmen bu problemin NP-zor sınıfına girmesi ve çözüm bulmanın zorluğundan dolayı sadece çok küçük boyutlu örneklerde optimum sonuç bulunabilmektedir.

Bu çalışmada, bu kaynak kısıtlı proje çizelgeleme problemlerinin büyük boyutlularını çözebilmek ve optimale yakın sonuçlar aramak için evrimsel arama algoritması olan bir genetik algoritma önerilmiştir. Önerilen algoritma kaynak kısıtlı proje çizelgeleme problemlerinin hem çok projeli problemlerini hem de tek projeli problemlerini çözebilmektedir.

Önerilen algoritma standart kıyaslama kütüphanelerinden biri olan PSPLIB ve MPSPLIB Kütüphaneleri verileri ile denenmiştir. Denenen algoritma optimale oldukça yakın sonuçlar üretmiştir. Sonuçlar ayrıca başka çalışmaların sonuçları ile kıyaslanmıştır.

Anahtar Kelimeler: Genetik Algoritma, kaynak kısıtlı proje çizelgeleme problemleri, proje çizelgeleme, sezgisel algoritmalar, kaynak kısıtlı çok projeli çizelgeleme problemleri.

**SCHEDULING THE MULTI RESOURCE-CONSTRAINED PROJECTS BY
HEURISTIC ALGORITHMS**

Uğur SATIÇ

Department of Industrial Engineering
MSc. Thesis

Adviser: Assist. Prof. Dr. Fahrettin ELDEMİR

Many algorithms have been developed for resource constrained project scheduling problems. Despite of the fact that this problem is belong to the class of NP-Hard and difficult of searching the optimal results, optimal results can be found in only a very small sized samples.

In this study, for solving large size problems and search near optimal results, we suggest a genetic algorithm. The proposed algorithm can solve bolt single-project and multi-project resource constrained multi project scheduling problems.

The proposed algorithm has tested with standard benchmark data in algorithm libraries which name is PSPLIB library and algorithm's produced results are very close to optimal results. The results are also compared with other studies.

Keywords: Genetic algorithms, resource constrained project scheduling, project scheduling, heuristic algorithm, resource constrained multi project scheduling

**YILDIZ TECHNICAL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

BÖLÜM 1

GİRİŞ

Faaliyetlerini planlayamayan ve bu nedenle maliyetini hesaplayamayan firmalar rekabet edemez ve faaliyetlerini sürdüremezler. İşin teslim sürelerinin yanlış verilmesi büyük sorunlar yaratmaktadır. Yüksek teslim süreleri müşterinin rakip firmaları tercih etmesine, gereğinden az verilen teslim sürelerinin ise gecikme cezası ödemesi ve müşteri güveni kaybı gibi olumsuz sonuçları olmaktadır. Bu nedenle proje sürelerinin tam ve doğru şekilde hesaplanması firmalar için hayati bir öneme sahiptir.

Proje planlama ve çizelgeleme yapılırken kullanılan kaynakların bulunurluğunun az olması ve maddi değerlerinin yüksek olması nedeniyle kaynakların israfını önlemek proje çizelgeleme konusunun en önemli bölümü olmuştur. Mevcut kaynakların kapasitelerini aşmadan en kısa proje tamamlanma süresini bulma problemi kaynak kısıtlı proje çizelgeleme problemi (KKPÇP) olarak adlandırılmaktadır.

Çalışmanın ikinci bölümünde literatürdeki proje çizelgeleme ile ilgili çalışmalar incelenecektir. Çalışmanın üçüncü bölümünde kaynak kısıtlı proje çizelgeleme problemi anlatılacaktır. Çalışmanın dördüncü bölümünde kaynak kısıtlı proje çizelgeleme problemini çözmek için kullanılan sezgisel metotlar incelenecek ve işlem aşamaları anlatılacaktır. Çalışmanın beşinci bölümünde genetik algoritma metodolojisi incelenecektir. Çalışmanın altıncı bölümünde kaynak kısıtlı proje çizelgeleme problemini çözmek için hazırlanan algoritma anlatılmış ve algoritmanın performansı değerlendirilecektir. Çalışmanın yedinci bölümü olan sonuç bölümünde yapılan çalışma değerlendirilecek ve literatüre yaptığı katkılar irdelenecektir.

1.1 Literatür Özeti

Proje yönetimine 1950'li ve 1960'lı yıllarda gösterilen ilgi daha sonra azalmış ancak yaklaşık son yirmi yılda yeniden hem araştırmacıların hem de uygulamacıların ilgisini çekmeye başlamıştır. Konu ile ilgili yayımlanan makale ve kitapların sayısındaki artıştan, bilgisayar yazılımlarının yaygınlaşmasından ve düzenlenen konferans ve seminer adedinin artışından proje yönetimine olan ilginin yeniden arttığını anlayabiliyoruz[1].

Çizelgelemenin tarihi oldukça eskidir. Henry Gantt'ın geliştirdiği Gantt Diyagramından bu güne çizelgeleme ve proje yönetimi işlemi daha karmaşık ve zor bir hale dönüşmüştür. Karmaşık ve büyük projelerin çizelgelenmesi için artık matematiksel yöntemler yeterli gelmemekte, işlem süreleri ve modelin kurulmasının zorluğu nedeniyle tercih edilmemektedir. Çözüm süreleri polinomsal zamanlı olarak adlandırılan bu zor problemlerin çözümü sezgisel yöntemler ile aranmaktadır.

Sezgisel yöntemlerden biri olan ve bu çalışmada kullanılan genetik algoritmanın genel ilkeleri John Holland tarafından 1975 [2] yılında "Adaptation in Natural and Artificial Systems" adlı kitapta ilk olarak ortaya konulmuştur[3]. Daha sonra kullanım alanı artmış ve algoritma teknikleri geliştirilmiştir.

Değişen ihtiyaçları karşılayabilmek için kaynak kısıtlı proje çizelgeleme problemi yeni alt türler eklenerek geliştirilmiş ve çeşitlenmiştir. Bu alt türlerden başlıcaları faaliyetlerin birden fazla kaynak kullanımına ve işlem zamanına sahip oldukları çok modlu kaynak kısıtlı proje çizelgeleme problemi ve birden fazla projenin aynı kaynaklar kullanılarak çizelgelendiği kaynak kısıtlı çok projeli çizelgeleme problemleridir.

1.2 Tezin Amacı

Bu çalışmada kaynak kısıtlı proje çizelgeleme problemini çözebilmek için sezgisel bir yöntem olan genetik algoritma kullanılarak visual studio programda c# programla dili ile bir bilgisayar uygulaması hazırlanacaktır. Hazırlanan algoritma PSPLIB ve MPSPLIB Kütüphanesi verileri ile intel core i5 2,4 GHz, 6 gb ramli bir bilgisayarda çalıştırılarak test edilecek ve aynı kütüphaneyi kullanan diğer araştırmacıların çalışmaları ile kıyaslanarak, önerilen algoritmanın performansı değerlendirilecektir. Böylece etkili bir

genetik algoritma uygulaması elde ederek kaynak kısıtlı çok projeli ve tek projeli problemler için daha iyi çözümler aranacaktır.

1.3 Hipotez

Bu tez çalışmasının başlangıç noktası, genetik algoritmanın seri çizelgeleme yapısı ve rastsal seçim kuralları ile tüm çözüm uzayına ulaşabileceği ve optimum çözümü bulabileceği düşüncesidir. Bu fikirle eğer algoritma yeteri kadar çalıştırırsa optimum çözümün bulunacağıdır. Doğru genetik parametreler kullanılarak optimum çözüm aranacak veya literatürdeki diğer çalışmaların buldukları en iyi çözüm geliştirilecektir.

LİTERATÜR ARAŞTIRMASI

Bu bölümde kaynak kısıtlı proje çizelgeleme problemi ve bu problemin çeşitli alt türleri ile ilgili literatürde bulunan çalışmalardan bazıları incelenmiştir.

2.1 Kaynak Kısıtlı Proje Çizelgeleme Problemi

Kaynak kısıtlı proje çizelgeleme problemleri literatürde sıklıkla çalışılmış olan bir problem türüdür. NP-hard sınıfında olduğu için çözümünde genellikle sezgisel algoritmalar tercih edilmiştir.

Kolisch vd. [4], çalışmalarında kaynak kısıtlı proje çizelgeleme problemleri üreten bir algoritma olan PROGEN'i geliştirmiştir. Bu algoritma sayesinde yaratılan problemleri öncelik kuralı kullanarak seri çizelgeleme metodu ile çözmüşlerdir.

Kolisch ve Sprecher [5], Kolisch v.d.'nin [4] çalışmasında kullanılan problemleri bir online problem kütüphanesine dönüştürerek diğer araştırmacıların algoritmalarını kıyaslayabilecekleri bir proje çizelgeleme problemleri kütüphanesi(PSPLIB) oluşturmuştur.

Kolisch ve Drexler [6], KKPÇP için, öncelik kuralı ve rastgele arama tekniklerinin birleşimi ile oluşturulmuş bir algoritma önermiştir. Önerilen algoritma uygulama olarak PSPLIB kütüphanesi problemlerini kullanmıştır. Algoritmanın sonuçları bölüm 6'daki kıyaslama bölümünde gösterilmiştir.

Kolisch [7], çalışmasında KKPÇP için, bir öncelik kuralı kullanarak bir tek çözüm üreten bir algoritma önermiştir. Önerilen algoritma uygulama olarak PSPLIB kütüphanesi

problemlerini kullanmıştır. Algoritmanın sonuçları bölüm 6'daki kıyaslama bölümünde gösterilmiştir.

Brucker v.d. [8], KKPÇP'ni çözmek için bir dal ve sınır algoritması kullanmışlar ve algoritma performansını, Kolisch v.d. [4], Kolish ve sprecher'ın [5] çalışmalarındaki problemler ile değerlendirilmiştir ve 1.2% optimumdan sapma ve 4.0% maksimumdan sapma göstermiştir.

Tsai ve Gemmill [9], çalışmasında KKPÇP için uyarlanmış bir tabu arama algoritması önermişlerdir. Yapılan çalışmada faaliyetlerin kaydırılmasının diğer faaliyetler üzerinde yaptığı etkiye göre sınıflandıran bir yapı vardır.

Schirmer [10], KKPÇP için vaka tabanlı değerlendirme yöntemi ile geliştirilmiş bir örnekleme yaklaşımı önermiştir. Önerilen algoritma PSPLIB kütüphanesi verileri ile sınanmıştır algoritma sonuçları bölüm 6'daki kıyaslama bölümünde gösterilmiştir.

Reyck ve Herroelen [11], KKPÇP'ne faaliyetler arasındaki minimal ve maksimal gecikmelerin eklenmesi ile oluşan güncellenmiş öncelik ilişkili kaynak kısıtlı proje çizelgeleme problemini (GRR-KKPÇP) çözmek için bir dal ve sınır algoritması önermişlerdir.

Hartmann [12], kaynak kısıtlı proje çizelgeleme problemi için bir genetik algoritma önermiştir. Önerilen algoritma PSPLIB kütüphanesi verileri ile sınanmıştır algoritma sonuçları bölüm 6'daki kıyaslama bölümünde gösterilmiştir.

Hartmann [13], çalışmasında KKPÇP için probleme göre kendini adapte eden bir genetik algoritma önermiştir. Önerilen algoritma problemin yapısına göre iki farklı kodlama yapısından(seri çizelgeleme ve paralel çizelgeleme) birini seçmektedir. önerilen algoritma PSPLIB kütüphanesi verileri ile sınanmıştır algoritma sonuçları bölüm 6'daki kıyaslama bölümünde gösterilmiştir.

Tormos ve Lova [14] KKPÇP için örnekleme ve ileri geri çizelgeleme metodunu birleştiren hibrit bir algoritma önermiştir. Önerilen algoritma PSPLIB kütüphanesi verileri ile sınanmıştır algoritma sonuçları bölüm 6'daki kıyaslama bölümünde gösterilmiştir.

Debels [15] KKPÇP için dağınık arama (Scatter search) ve genetik algoritmayı birleştiren hibrit bir algoritma önermiştir. Önerilen algoritma PSPLIB kütüphanesi verileri ile sınanmıştır algoritma sonuçları bölüm 6'daki kıyaslama bölümünde gösterilmiştir.

Paksoy [16], KKPÇP için bir genetik algoritma önerdi. Önerilen algoritma rastgele anahtar gösterimi ve öncelik kurallı yöntemleri ile çizelgeleme yapmaktadır. Önerilen algoritma PSPLIB kütüphanesi verileri ile sınanmıştır algoritma sonuçları bölüm 6'daki kıyaslama bölümünde gösterilmiştir.

Sengupta v.d. [17] KKPÇP için çalışma zamanı tüketim maliyeti ilişkisine dayalı bir genetik algoritma önermiştir. Algoritmada, her bir fonksiyonel ünitenin ayrı kromozomlarda gösterilmekte böylece çok katlı bir yapı oluşmaktadır.

2.2 Kaynak Kısıtlı Proje Çizelgeleme Probleminin Alt Türleri

Kaynak kısıtlı proje çizelgeleme problemi literatürde geçen oldukça eski bir problemdir. Zaman içerisinde kaynak kısıtlı proje çizelgeleme problemi modeli, gerçek hayattaki durumlara daha uygun hale getirmek için geliştirilmiştir. Böylece KKPÇP'nin alt türleri oluşmuştur.

2.2.1 Çok modlu kaynak kısıtlı proje çizelgeleme problemi

Çok modlu kaynak kısıtlı proje çizelgeleme problemi (MMKKPÇP), bir faaliyetin farklı kaynak kullanım oranları ile daha kısa veya daha uzun sürede yapılabileceği gerçeğine dayanılarak geliştirilmiştir. Bu problemde faaliyet çizelgesi hazırlanırken faaliyetlerin hangi modunun kullanılması gerektiği de seçilmelidir. Çözümü tek modlu problemlere göre daha zor ve işlem süreleri daha uzundur.

Boctor [18], çalışmasında MMKKPÇP için 7 faaliyet seçim kuralı ve 3 mod seçim kuralı kombinasyonları ile yaratılmış 21 sezgisel metot ile rastsal olarak yaratılmış 240 problem üzerinde çalışmıştır.

Mori ve Tseng [19], MMKKPÇP üzerinde, geriye doğru çizelgeleme yöntemiyle ve modları rastgele seçerek faaliyetleri sıralamaya çalışmıştır. Oluşturulan sıralama fizibil ise popülasyona eklenmiş ve istenilen popülasyon sayısına ulaşıncaya kadar devam

edilmiştir. Tek noktalı çaprazlama kullanılmış ve çaprazlama için seçilen genlerden biri en kısa proje süresine sahip olan elitler kümesinden seçilmiştir. Mutasyon işleminde bir kromozom seçilmekte ve birkaç faaliyetin modu rastsal olarak değiştirilmektedir. Yeni nesil ise çaprazlama, mutasyon, elitist seçim ve yeniden yaratma işlemleri ile gerçekleştirilmektedir.

Sprecher ve Drexl [20], MMKKPÇP'ni çözmek için bir dal ve sınır algoritması kullanmışlardır. Kullandıkları algorithmada 9 adet çizelgeleme kuralı vardır. Problem boyutuna göre bu 9 kuraldan biri seçilmektedir. Paralel çizelgeleme algoritması kullanılmış ve faaliyetin en kısa işlem süresine sahip kaynak kapasitesine uygun modu seçilerek çizelgeleme yapılmaktadır. Seçilen faaliyetin hiç bir modu uygun değilse bir önceki faaliyetin modu değiştirilerek tekrar çizelgeleme yapılır. Algoritmayı sınamak ve uygulama yapmak amacı ile ProGen programı ile üretilmiş 10.000 örnek proje üzerinde çalışılmıştır.

Ahn ve Erenguc [21], çalışmalarında MMKKPÇP için sadece mod seçiminin yanında ayrıca işlem süresi ve maliyetinde değiştirmeyi içeren sezgisel bir algoritma kullanmışlardır. Kullanılan algoritma iki aşamalıdır. Birinci aşamada fizibil bir çizelge yaratılmakta. İkinci aşamada bu çizelge 6 adet kural ile geliştirilmeye çalışılmaktadır. Daha fazla geliştirme yapılamayınca kadar ikinci aşama uygulanmakta daha sonra ilk aşamaya dönüp işlemler tekrar edilmektedir. Önerilen ProGen programı ile üretilmiş problemler üzerinde uygulanmıştır.

Afshar-Nadjafi v.d. [22], MMKKPÇP türlerinden biri olan mod belirleme ve kaynak kısıtlı proje çizelgeleme problem üzerinde çalışmıştır. Bu modelde alt faaliyetleri kaynak kullanımlarına göre ayırık kümelere bölerek ve aynı kümedeki elemanları aynı modu kullandırarak çözüm aranmaktadır.

Ranjbar ve Kianfar [23], bir kaynak sayısına sahip MMKKPÇP için bir genetik algoritma kullanmıştır. Random key gösterimi kullanılarak çizelgeyi kromozom olarak temsil etmişlerdir. Seri çizelgeleme metodu ile faaliyetler sıralamışlardır. Yen nesil yaratılırken çaprazlama yapılmış daha sonra verilen olasılığa göre çaprazlama ile yaratılanlar mutasyona uğratarak yeni nesle eklenmiştir.

Bouleimen ve Lecocq [24], tek ve çok modlu kaynak kısıtlı proje çizelgeleme problemlerini çözmek için bir Tavlama benzetimi (SA) algoritması önermiştir. Önerilen algorithma seri çizelgeleme yapısı (SSGS) kullanılmış ve PSPLIB kütüphanesi verileri ile algoritma uygulanmıştır. Hazırlanan algoritma belirli sayıda işlem den veya kritik yola ulaşmanın ardından işlem sonucunu vermektedir.

Shi-man v.d. [25] MMKKPÇP'ni çözmek için, çok aşamalı yapıda bir genetik algoritma önermiştir. İlk aşamada kromozomlar faaliyet sırasına göre dizilmiş. İkinci aşamada bu kromozom yapısında çeşitli faaliyet modları denenmiştir. Kromozomlar ikili (binary) sistem ile kodlanan faaliyetlerin kaynak kıyaslaması ilişkisi algoritması ile üretilmiştir. Daha sonra standart elitist seçim, çaprazlama ve mutasyon işlemleri yapılarak nesiller geliştirilmiştir.

2.2.2 Kaynak kısıtlı çok proje çizelgeleme problemi

Günümüzde firmalar artık birden çok projeyi aynı anda yürütmektedir. Bu nedenle klasik tek projeli kaynak kısıtlı proje çizelgeleme problemi modeli, yetersiz kalmaktadır. Bu eksiği gidermek adına Birden çok projenin belirli ve kısıtlı kaynakları kullanmak için yarıştığı kaynak kısıtlı çok proje çizelgeleme problemi (KKÇPÇP) modeli ortaya çıkmıştır.

Homberger [26]'ün çalışmasında KKÇPÇP için biri çoklu ajan sistemi, diğeri yeniden başlatma evrim stratejisi kullanan iki farklı algoritma önermiştir. Önerilen algoritmalar PSPLIB kütüphanesi problemlerinin birleştirilmesi ile oluşturulan çok projeli problemler ile sınanmıştır. Bu çalışmada kullanılan problemler daha sonra çok projeli çizelgeleme problemi kütüphanesi (MPSPLIB) olarak diğ er araştırmacıların kullanımına açılmıştır.

Gonçalves v.d. [27] çalışmalarında, KKÇPÇP için faaliyet önceliği, gecikme zamanları ve proje başlangıç tarihlerine dayanarak faaliyetleri çizelgeleyen genetik bir algoritma kullanmıştır. Gecikme zamanlarını faaliyet bazında değerlendirerek çözüm uzayının büyüklüğünün kontrol etmeye çalışmışlardır.

Le [28] çalışmasında, KKÇPÇP'ni kaynakların projeler arasında taşınması için harcanan süreyi de içerecek şekilde genişleterek incelemiştir. inşaat projelerinde önemli bir yer

tutan kaynak taşıma süresini minimum'a indirecek bir örnekleme algoritması kullanmıştır.

Ulusoy v.d. [29], gecikme maliyetli dinamik çok projeli çizelgeleme problemi için yerel arama tabanlı bir algoritma önermiştir.

Homberger [30], evrimsel aramaya dayanan bir koordinasyon mekanizması kullanan çoklu ajan sistemi algoritması geliştirmiştir. Geliştirilen algoritma Homberger [26]'deki çalışma ile kıyaslanmıştır. Kıyaslama sonuçları 6. bölümdeki kıyaslama tablosu 6.7'dedir.

Browning ve Yassine [31] çalışmasında, KKÇPÇP'ni 20 öncelik kuralı ve paralel çizelgeleme yapısı kullanarak incelemiştir. Algoritmada çoklu proje yapısını kullanmış ve PSPLIB kütüphanesi ile uygulama yapmıştır.

Bakshi v.d. [32], KKPÇP türlerinden biri olan KKÇPÇP üzerinde çalışmış. Parçacık kümesi optimizasyonu PSO tabanlı bir algoritma geliştirmiştir.

Adhau v.d. [33] KKÇPÇP için, açık artırmaya dayalı müzakere kullanan bir dağıtılmış çoklu ajan algoritması önermiştir. Algoritma kaynak çakışmalarını önlemek ve problemin optimum sonucunu bulabilmek için en iyi metodu ihale yöntemi kullanarak aramaktadır.

Adhau v.d. [34] Projelerin nihai tamamlanma ve proje gecikmelerinin yanında kaynakların taşınması süresini ve maliyetini de en aza indirmeyi üçüncü bir amaç edinerek KKÇPÇP için, açık artırmaya dayalı müzakere kullanan bir çoklu ajan algoritması geliştirmiştir. Diğer algoritmalar ile kıyaslama yapabilmek adına kaynak taşıma süresi ve maliyeti ihmal edilerek hesaplamalarda yapılmıştır. Kıyaslama sonuçları 6. bölümdeki kıyaslama tablosu 6.7'dedir.

2.2.3 Çok modlu kaynak kısıtlı çok projeli çizelgeleme problemi

Beşikçi v.d. [35], çok modlu kaynak kısıtlı çok projeli çizelgeleme ve kaynak ayırma problemi üzerinde çalışmıştır. Kaynak ayırma probleminin farkı, bazı kaynakların çok projeli çalışmalarda taşınması ve paylaşılmasının yüksek maliyetli olacağı gerçeğine dayanarak kaynakların paylaşılması yerine projelerin tamamlanma süresine kadar

kaynakların projelere atanması gerektiğini esas almasıdır. Bu esasa göre iki aşamalı bir genetik algoritma geliştirilmiştir. Geliştirilen algoritmada birinci aşamada projelere kaynaklar paylaştırılmakta ikinci aşamada ise kaynak paylaşım çizelgesi genetik algoritma ile geliştirilmektedir. Hazırlanan algoritma PSPLIB kütüphanesindeki çok modlu örnekler üzerinde çalıştırılmış ve beşikçi ve arkadaşlarının daha önceki çalışmasındaki tek aşamalı algoritma ve deterministik yöntemlerle bulunan sonuçlarla karşılaştırılmıştır. Tek aşamalı algoritmaya göre iki aşamalı algoritma optimuma yaklaşımda daha başarılı olmuştur.

Ulusoy ve Can [36], Çok modlu kaynak kısıtlı çok proje çizelgeleme problemi için 2 aşamalı bir genetik algoritma önermiştir. Birinci aşamada projeler mega bir proje olarak birleştirilmekte ikinci aşamada bu tek proje çözülmektedir.

2.2.4 Kaynak yatırım problemi

Yeni yatırım problemleri ve kaynak kapasitesinin henüz belirli olmadığı durumlar için kaynak kısıtlı proje çizelgeleme probleminin, kaynak sınırının kaldırılması ile oluşmaktadır. Kaynak kullanımı bir maliyet fonksiyonu veya bir sınır fonksiyonu konularak en kısa proje süresi ile en uygun kaynak yatırım miktarı aranmaktadır.

Shadrokh ve Kianfar [37], kaynak kısıtlı proje çizelgeleme problemi türlerinden biri olan kaynak yatırım problemi üzerinde çalışmıştır. Bu problemde kaynak kapasitesi ve proje tamamlanma süresi maliyet atanarak en aza getirilmeye çalışılmaktadır.

Erdal [38], kaynak kapasitesi aşılmasını ceza puanı ile önlemeye çalışan, birincisi kaynak uygunluğu, diğeri süre uygunluğu olmak üzere iki uygunluk fonksiyonu çözümü kullanan bir genetik algoritma geliştirmiştir.

Ponz-Tienda v.d. [39], kaynak kısıtlı proje çizelgeleme problemi türlerinden kaynak yatırım problemi için bir genetik algoritma kurmuştur. Algoritmada yeni nesil yaratılırken tüm elemanlar çaprazlama ile yaratılmış, daha sonra belli bir olasılığa göre mutasyona tabi tutulmuşlar ve yeni nesle bu şekilde eklenmişlerdir. Kurulan algoritmayı diğerlerinden ayrılan tarafı mutasyon oranının üretilen neslin uygunluk oranına göre artık azalmasıdır. Algoritma PSPLIB kütüphanesi verileri ile uygulama yapmıştır ve algoritma sonuçları bölüm 6'daki kıyaslama tablosunda verilmiştir.

2.2.5 Diğer çizelgeleme problemleri

Kaynak kısıtlı proje çizelgeleme problemleri dışında diğer çizelgeleme problemi türleri üzerinde de ağırlıklı olarak çalışılmıştır.

2.2.5.1 Üretim çizelgeleme problemi

Li v.d. [40] genetik algoritma kullanarak bir kapasite atama ve üretim çizelgeleme problemi üzerinde çalışmıştır. Genetik algoritma sayesinde darboğazları önlenmeye ve tam zamanında üretim sistemindeki belirsizlikleri gidermeye çalışılmıştır.

ElMaraghy v.d. [41] çalışmalarında genetik algoritma kullanarak iş yükü ve makine olmak üzere iki kaynaklı üretim çizelgeleme problemi üzerinde çalışmışlardır. Algoritmalarında 6 farklı kural ve 8 farklı uygunluk fonksiyonu kullanmışlardır. Ayrıca tek kaynağı makine olan başka bir genetik algoritma hazırlayarak iki algoritma kullanarak hangi kuralların daha etkili olduğunu incelemiştir. Sonuç olarak, ortalama faaliyet süresini kısaltmak için en kısa faaliyeti seçme kuralı, maksimum faaliyet süresini kısaltmak için en az gecikmeli faaliyeti seçme kuralı gibi kuralların kullanılması gerektiğini önermişlerdir.

Delavar v.d. [42] çalışmalarında bir tedarik zincirinde hem üretimi hem de dağıtımı optimize etmek amacıyla iki farklı genetik algoritma kullanmışlardır. Birinci genetik algoritma hava taşımacılığı ile dağıtımı üretim çizelgesine göre optimize etmektedir. Kullanılan ikinci genetik algoritma ise üretim çizelgeleri yaratmaktadır. Kullanılan genetik algoritmalarından daha yüksek performans alabilmek için taguchi deneysel tasarım metodu ile en uygun genetik algoritma ayarları araştırılmıştır. Kurulan algoritmalar birçok problemle denenmiş ve algoritmanın yüksek boyutlu problemlerdeki performansı incelenmiş ve ikinci genetik algoritmanın daha iyi olduğu problem kümelerinde ortalama, %0.73'lük bir ortalama sapma ile %5.41'lik ortalama sapması olan birinci genetik algoritmadan daha iyi olduğu söylenmiştir.

2.2.5.2 İş çizelgeleme problemi (atölye çizelgeleme)

Orero ve Irving [43], bir iş atama problemi olarak enerji sistemlerinde için jeneratör çizelgeleme problemi üzerine bir genetik algoritma kullanarak çalışmıştır. Talebe ve sistem kısıtlarına göre termal jeneratörlerin çizelgelenmesini sağlayacak genetik bir

algoritma hazırlamışlar ve 26 jeneratörlü bir uygulama yapmışlardır. Yapılan uygulamada enerji talebi ve yakıt gideri verileri kullanılarak 1 saatlik çizelgeler hazırlanmıştır.

Wong [44], bir iş atama problemi olan kısa dönemli hidro-termal santral çizelgeleme üzerinde benzetilmiş tavlama (SA) ve genetik algoritmayı birleştiren bir algoritma ile çalışmış. Çok tesisli hidro-termal çizelgeleme üzerinde problemi üzerinde uygulama yapmıştır. karşılaştırılan 5 algoritma arasından önerilen algoritma %9.35'lik ortalama sapma ile ilk sırada yer almıştır.

Brucker [45], çalışmasında KKPÇP'nin bir alt türü olan olan makine çizelgeleme problemi için dal ve sınır algoritması ile yerel arama algoritmalarını birleştiren bir yaklaşım önermiştir.

Carter ve Ragsdale [46], genetik algoritma ile bir iş atama problemlerini olan gazetelerin reklam afişlerinin basımını problemini çizelgelemiştir. Önerilen algoritma üretim zamanını %25 azaltmaktadır.

Gupta v.d. [47], bir havaalanına gelen uçakların bakım ihtiyaçlarını karşılamak ve eldeki kısıtlı kaynakları (teknisyen gibi) dikkate alarak bakımları çizelgelemek için bir simülasyon programı kullanmışlardır. Sezgisel bir tekniğin seçilmesinin nedeni ise havacılık sektöründe rastsal olaylar, hava durumu, personel durumu gibi pek çok etkenin yarattığı belirsizlik yüzünden deterministik bir tekniğin uygulanmasının çok güç olmasıdır. Çalışmada kullanılan AUTO-MOD adındaki ticari program aslında bir atölye çizelgeleme programıdır ve genetik algoritma ile çalışmaktadır. Çalışmanın sonuçlarına göre personelin iş yükü dağılımı oldukça düzensizdir. Bazı günler çoğunlukla iş yükü yüzde 10 seviyesinde kalırken bazı günler %100'lük bir iş yükü oluşmakta ve darboğaz görülmektedir. Algoritmanın sonuçlarına göre mesai ve vardiya zamanları uygun şekilde ayarlanabilmektedir. Vardiya düzeltilmiş de ise iş yükü dengelenmekte ve dar boğazlar giderilmektedir.

Juang v.d. [48], personel atama problemi türlerinden biri olan personel eğitimi çizelgeleme problemi üzerinde genetik algoritma ile çalışmıştır. Bu algoritma ile optimum kurs çizelgesinin hazırlanmaya çalışılmıştır.

Chang v.d. [49], bir iş atama problemi olarak bilgisayar yazılımı projesi çizelgelemesi üzerinde genetik algoritma ile çalışmıştır. Bu problemde çalışanlar kaynak olarak görülmekte ve kaynak kapasitesi belirleme problemleri gibi projelerin ve çalışanların çizelgelenmesi yapılmaktadır.

Papakostas v.d. [50] uçağın uçuş rotasına atanmasından sonra en ekonomik şekilde hangi hava limanında bakım yapılabilceği sorusunu araştırmıştır gidilecek hava alanlarındaki; bakım personeli çizelgesini, bakım maliyetini, operasyonel riski, uçuş gecikmesini ve uçağın bakımdan önce kalan uçuş saatine göre bir uygunluk fonksiyonu belirlemiştir. Bu uygunluk fonksiyonuna göre senaryolar yaratarak en iyi bakım yerini belirlemeye çalışmışlardır. En uygun bakım yerlerinin ve zamanlarının belirlenmesi ile planlanmamış bakımların önlenmesi ve azaltılması sağlanmaktadır. Böylece uçağın işletme maliyetinin %10 ila %20'sini kapsayan bakım maliyeti azalmakta ayrıca uçuş güvenliği artmaktadır.

Belien v.d. [51] çalışmalarında bir havayolu bakım şirketi için uygun personel vardiya çizelgesini hazırlamak için uçak bakım alanı problemi üzerinde çalışmıştır. Personel çizelgeleme problemi karışık tamsayı lineer programlama (MILP) ile çözülebilmektedir. Belien ve arkadaşları çalışmalarında çalışma takımı sayıları üzerinde değişiklikler yaparak bakım alanı problemini non-lineer halden lineer hale dönüştürmüşlerdir. Problem üzerinde yaptıkları bu yeni formülasyon ile problemi ticari MILP çözücülerin problem biçimine uygun hale getirmişlerdir. MILP algoritması ile bulunan sonuçları değerlendire bilmek için ikinci bir algoritma kullanılmıştır. Kullanılan ikinci algoritma sezgisel bir algoritma olan tabu arama algoritmasıdır. Her iki algoritmada da aynı 40 örnek çalıştırılmış ve sonuçlar kıyaslanmıştır. İki algoritma kıyaslığında MILP algoritması genelde optima en yakın sonuçları (optimumdan sapma $\frac{1}{2}$ 9.84) üreterek tabu arama algoritmasında (optimumdan sapma %12.98) daha iyi bir performans göstermiştir.

Remenyi ve Staudacher [52] çalışmalarında, uçak motor bakımı için bakımın ve bakım operasyonlarını planlamadaki zorluklardan dolayı bu bakımın atölye operasyonları için uygun bir çizelgeleme kuralları bularak bakım operasyonunu geliştirmeye amaçlamışlardır. Çizelgeleme kurallarının kullanılmasının bakım operasyonunu geliştireceği öne sürülmüştür. Makalede araştırılan temel soru; dağıtılmış çizelge

kurallarının kullanılması ile bakım sisteminin lojistik hedeflerinde bir gelişme olup olmayacağıdır. Hipotezi denemek için sistematik simülasyon tabanlı bir yaklaşım geliştirilmiş ve bu yaklaşım ile çizelgeleme kuralları belirlenmiştir. Simülasyon sonuçları her takım tezgâhından önce dağıtılmış çizelgelemede bir belirsizlik olduğunu göstermiştir.

KAYNAK KISITLI PROJE ÇİZELGELEME PROBLEMİ

3.1 Kaynak Kısıtlı Proje Çizelgeleme Problemi

Kaynak kısıtlı proje çizelgeleme probleminin (KKPÇP) amacı, proje tamamlanma süresini en aza indirebilmek için projedeki her bir faaliyetin fizibil başlangıç tarihlerini belirlemektir. Problemi tarif etmenin bu kadar kolay olmasına rağmen, problemi çözmek oldukça zordur[53].

Günümüzün modern ekonomik ortamında, proje yönetiminde başarılı olabilmek için projenin tamamlanma süresi ve maliyetinin hesaplanılabilir olması ve hedeflenen proje tamamlanma süresi içerisinde gerçekleştirilebilmesi gereklidir[54]. Proje planlamaya gösterilen ilgi 1950'li yıllarda gösterilen yoğun ilgiden sonra günümüzün rekabet şartları nedeniyle son 10 yılda ilgi yeniden artmış birçok araştırmacı bu konuya yönelmiştir[1]. Bir çok araştırmacı bu problemi kullanmak için deterministik ve sezgisel algoritma kullanmıştır[53]. Sezgisel metodolojilerin gelişmesi ile literatürde son yıllarda yapılan proje çizelge çalışmalarda çoğunlukla sezgisel metodolojiler kullanılmıştır[1].

Proje çizelgelemesi yapılırken her projede var olan kaynak kısıtları dikkate alınarak çizelgeleme yapılması gereklidir. Zaman ve maliyet gibi kaynak kısıtlarının kullanılması çizelgelemenin gerçek hayat problemlerine uygun çözümler bulması için önemli bir etkenidir[55]. Kaynak kısıtlı, proje çizelgeleme problemini kombinasyonel bir probleme dönüştürür bu nedenle sadece küçük boyutlu projeler deterministik metotlar ile çözülebilir[56].

Kaynak kısıtlı proje çizelgeleme problemi (KKPÇP) üzerinde yapılan çalışmalar elde edilecek hedefler açısından incelendiğinde ikiye ayrılmaktadırlar. Birinci grup çalışmalar

hedef olarak proje süresini minimize etmeyi amaçlayan çalışmalardır. İkinci grup çalışmalar maliyet hedefleri üzerinde yoğunlaşmış ve projenin toplam maliyetinin en aza indirilmesi veya projenin net şimdiki değerinin maksimize edilmesi hedefleri üzerinde durmuştur. Proje yönetiminde gerçek hayatta sıklıkla karşılaşılan durumda, projenin belirli bir proje bitim tarihinden önce tamamlanması ve aynı zamanda proje toplam maliyetinin en aza indirilmesi gerekmektedir[54].

3.1.1 Kaynak kısıtlı proje çizelgeleme probleminin modeli

Tek projeli kaynak kısıtlı proje çizelgeleme probleminde problem modeli şu şekildedir[5,1]:

$$MinZ = \sum_{t=EFT_j}^{LFT_j} tX_{jt} \quad (3.1)$$

$$\sum_{t=EFT_j}^{LFT_j} X_{jt} = 1; \quad j = 0, 1, \dots, n + 1 \quad (3.2)$$

$$\sum_{t=EFT_i}^{LFT_i} tX_{it} \leq \sum_{t=EFT_j}^{LFT_j} (t - d_j)X_{jt} \quad j = 1, \dots, n + 1 \text{ ve } i \in P_j \quad (3.3)$$

$$\sum_{j=1}^J k_{jr} \sum_{r=t}^{t+d_j-1} X_{jt} \leq K_r \quad r \in R, \quad t = 1, \dots, r \quad (3.4)$$

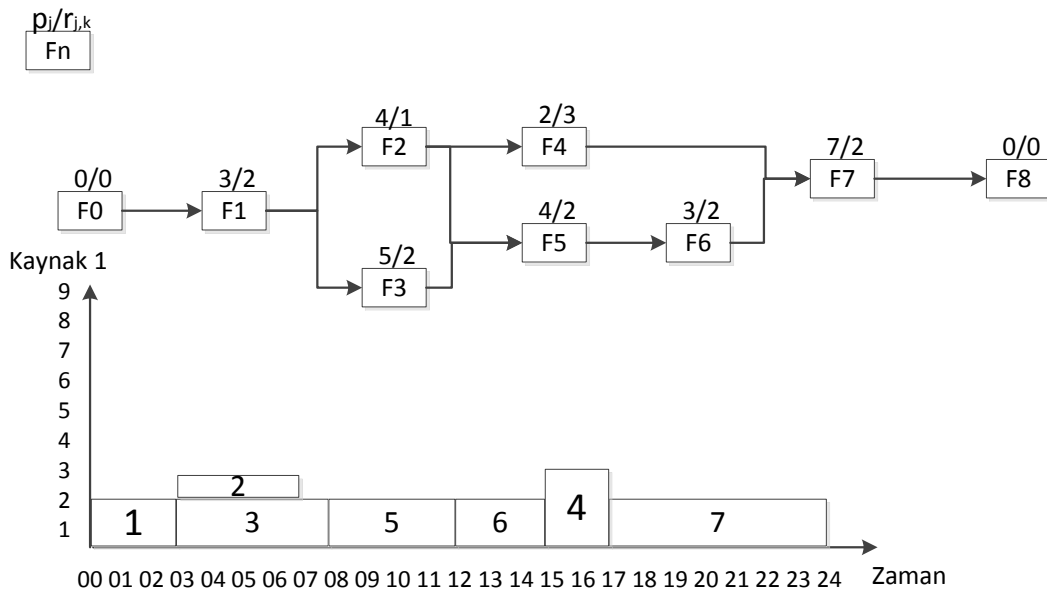
$$X_{jt} = \begin{cases} 1, & \text{eğer } j \text{ faaliyeti } t \text{ dönemi sonunda bitiyorsa} \\ 0, & \text{diğer durumlarda} \end{cases} \quad (3.5)$$

Kullanılan sembollerin tanımları aşağıda açıklanmıştır.

- t : zaman indisi (t=1,...,T),
- j : faaliyet indisi (j=1,...,J),
- d_j : "j" faaliyetinin süresi
- P_j : "j" faaliyetinin öncüllerinin kümesi,
- EFT_j : "j" faaliyetinin en erken bitiş zamanı,
- LFT_j : "j" faaliyetinin en geç başlangıç zamanı,
- R : yenilenebilir kaynaklar kümesi,
- k_{jr} : j faaliyetinin r kaynağından birim zamanda kullanım miktarı,
- K_r : r yenilenebilir kaynağının birim zamanda kullanım üst sınırı,

Burada; KKPÇP probleminin amacı formül 3.1'de gösterildiği gibi projenin bitiş tarihini en aza indirmektir. Formül 3.2'de her bir j faaliyetinin çizelgelenmesi sağlanmaktadır. Bir projenin $P = n+1$ adet faaliyet (veya iş yada görev) oluştuğu kabul edilir. Faaliyetler arasındaki bazı teknik gereksinimlerden dolayı, formül 3.3'te gösterildiği gibi faaliyet j , başlama bilmesi için bağlı olduğu faaliyetlerin X_{it} , $i \in P_j$, tamamlanmış olması gereklidir. 0. ve $n+1$. faaliyetin projenin başlangıç ve bitişini temsil eden temsili faaliyetler olduğu ve bu faaliyetlerin işlem süresi ve kaynak kullanımının olmadığı kabul edilir[57]. İkinci olarak, bir faaliyeti gerçekleştirmek için limitli kapasiteli kaynakları kullanmaya ihtiyaç duyar. $K = \{1, \dots, K\}$ olmak üzere K çeşit kaynak türü vardır. kaynak türü yenilenebilir ise her periyot başlangıcında kaynak türü K 'nın kapasitesinin K_r olduğu kabul edilir. Faaliyet j 'yi gerçekleştirebilmek için $k_{j,r}$ birim kaynak $K \in R$ ya gereksinim vardır[57]. Faaliyet j 'yi gerçekleştirmek için yeterli kaynak bulunmadığı durumda, kaynak seviyesi yeterli olana kadar faaliyete başlanılamaz.

Şekil 3.1'de $n = 7$ faaliyetli, kaynak kapasitesi 3 olan tek kaynaklı basit bir kaynak kısıtlı proje çizelgeleme problemi verilmiştir.



Şekil 3.1 Bir kaynak kısıtlı proje çizelgeleme problemi

3.2 Çok Modlu Kaynak Kısıtlı Proje Çizelgeleme Problemi

Kaynak kısıtlı proje çizelgeleme problemi çözmesi zor, polinomsal zamanda (NP) kolayca doğrulanabilen bir problemdir. Deterministik algoritmalar bu problemleri çözemez veya bilgisayardaki çözüm zamanları kabul edilebilenin çok üstündedir. Kaynak kısıtlı proje çizelgeleme problemine (KKPÇP) göre çok modlu kaynak kısıtlı proje çizelgeleme problemi (ÇMKKPÇP) çok daha karmaşıktır. Bu nedenle araştırmacılar bu problemi çözebilmek için çoğunlukla sezgisel metotlar kullanmışlardır[25]. Bu problem türü faaliyetlerin tek şekilde veya birden fazla şekilde yapılmasına bağlı olarak tek-modlu veya çok-modlu ismini alabilmektedir[19].

Klasik kaynak kısıtlı proje çizelgeleme probleminde olduğu gibi, çok modlu kaynak kısıtlı proje çizelgeleme probleminde de her faaliyet birbirine başlangıç, bitiş zamanı bağı ile bağlıdır. Özel olarak ÇMKKPÇP'nde her faaliyet, farklı işlem zamanı ve kaynak kullanımına sahip birden fazla mod ile gerçekleştirilebilir[23]. Bir faaliyet bir mod seçilerek başlatıldıktan sonra faaliyetin modu değiştirilemez[19].

ÇMKKPÇP proje çizelgeleme konusundaki en önemli alanlardan biridir ve bu problem gerçek hayattaki durumlara daha yakındır. Fakat bu problemin zorluğu bizi meta-sezgisel algoritmalar kullanmaya zorlamaktadır[22].

3.2.1 Çok modlu kaynak kısıtlı proje çizelgeleme probleminin modeli

ÇMKKPÇP'nin modeli KKPÇP'ine benzese de arada çok modlu yapıda kaynaklanan küçük farklılıklar vardır[19][55];

$$MinZ = \sum_{t=EFT_j}^{LFT_j} tX_{jmt} \quad (3.6)$$

$$\sum_{m=1}^{M_j} \sum_{t=EFT_j}^{LFT_j} X_{jmt} = 1; \quad j = 0, 1, \dots, n+1 \quad (3.7)$$

$$\sum_{m=1}^{M_j} \sum_{t=EFT_i}^{LFT_i} tX_{imt} \leq \sum_{m=1}^{M_j} \sum_{t=EFT_j}^{LFT_j} (t - d_{jm})X_{jmt} \quad j = 1, \dots, n+1 \text{ ve } i \in P_j \quad (3.8)$$

$$\sum_{m=1}^{M_j} \sum_{j=1}^J k_{jmr} \sum_{r=t}^{t+d_j-1} X_{jmt} \leq K_r \quad r \in R, \quad t = 1, \dots, r \quad (3.9)$$

$$X_{jt} = \begin{cases} 1, & \text{eğer } j \text{ m. modda faaliyeti } t \text{ dönemi sonunda bitiyorsa} \\ 0, & \text{diğer durumlarda} \end{cases} \quad (3.10)$$

Kullanılan sembollerin tanımları aşağıda açıklanmıştır.

- t : zaman indisi (t=1,...,T),
- j : faaliyet indisi (j=1,...,J),
- d_j : "j" faaliyetinin süresi
- P_j : "j" faaliyetinin öncüllerinin kümesi,
- M_j : "j" faaliyetinin mod adedi,
- m : mod indisi (m = 1,...,M_j),
- EFT_j : "j" faaliyetinin en erken bitiş zamanı,
- LFT_j : "j" faaliyetinin en geç başlangıç zamanı,
- R : yenilenebilir kaynaklar kümesi,
- k_{jr} : j faaliyetinin r kaynağından birim zamanda kullanım miktarı,
- K_r : r yenilenebilir kaynağının birim zamanda kullanım üst sınırı,

Proje N+1 faaliyetten oluşmaktadır. 0. ve N+1. faaliyetler kukla faaliyettir, 1 modları vardır ve işlem süreleri yoktur (d_{j1}=0)[23]. j. faaliyet herhangi bir modda (i = 1,...,M_i) başlayabilir, başladıktan sonra modu değiştirilemez. i. mod ile gerçekleştirilen faaliyetin işlem süresi d_{jm}'dir ve kaynak kullanımı k_{jmk} olarak gösterir. K tipindeki kaynak her periyot K_r kapasitesine sahiptir[19]. Hedef fonksiyonu formül 3.6'da gösterildiği gibi proje tamamlanma süresini en kısa yapacak faaliyet sırasını ve faaliyet modlarını belirlemektir[23][25]. Formül 3.7 kullanılan çizelgeleme periyodu içerisinde tüm faaliyetlerin bir mod ile çizelgelenmesini sağlar. Formül 3.8 faaliyetlerin birbirleri ile ilişkini belirten faaliyet öncüllük ilişkilerinin yerine getirilmesini sağlar. Formül 3.9 ise kaynak kullanımının üst sınırının hiç bir zaman aşılamayacağını ifade eder[55].

Çizelge 3.1'de görüldüğü gibi mod sayısına göre faaliyetin birden fazla tamamlanma süresi ve kaynak kullanımı vardır. Tek modlu problemlerde olduğu gibi ilk ve son faaliyet ise kukla faaliyettir ve bunların faaliyet süresi ve kaynak gereksinimi olmayan tek modu vardır.

Çizelge 3.1 Çok modlu faaliyet bilgileri

Faaliyet Numarası	Mod	Faaliyet Süresi	Kaynak Gereksinimi
1	1	0	0
2	1	2	5
	2	3	3
3	1	5	7
	2	4	10
4	1	5	6
	2	3	10
5	1	4	5
	2	3	8
6	1	6	3
	2	3	9
7	1	7	3
	2	5	7
8	1	6	3
	2	5	8
9	1	0	0

3.3 Çok Projeli Kaynak Kısıtlı Proje Çizelgeleme Problemi

Günümüzde üretim ve servis firmaları, aynı anda birden fazla proje ile uğraşmak zorundadırlar[35]. Günümüzde yapılan çalışmalarının %90'ından fazlası kısıtlı kaynakları paylaşan çok projeli yapıdadır[58]. Çizelgeleme kısıtlı kaynakların faaliyetler arasında paylaştırılması ve faaliyetlerin başlangıç ve bitiş zamanlarının belirlenmesidir. Aynı kısıtlı kaynaklar için yarışan birden fazla projenin çizelgelenmesi gerektiği durumda çözüm süreci oldukça karmaşık bir hal alır[27]. Piyasa rekabeti arttıkça firmaların eş zamanlı olarak yürüttüğü işlerin önem dereceleri artmakta, kaynakların farklı projelere paylaştırılması kararları başarısızlıkların önlenmesi için çok önemli hale gelmiştir[36].

Tek projeli kaynak kısıtlı proje çizelgeleme probleminin (RCPSP) aksine, çok projeli kaynak kısıtlı proje çizelgeleme probleminde (RCMPSP), toplam tamamlanma süresi en aza indirilmeye çalışılırken aynı zamanda her bir proje içinde gecikme süresi azaltılmaya çalışılmaktadır[31].

Çok projeli kaynak kısıtlı proje çizelgeleme problemi ile çalışılırken iki farklı yaklaşım kullanılmaktadır: Birinci yaklaşımda tüm projeler hayali bir başlangıç ve bitiş düğümü ile birleştirilerek çok projeli yapının tek projeli yapıya dönüştürülmesi ve aynı kaynak

kısıtlı proje çizelgeleme probleminde olduğu gibi (RCPS) çözülmesi amaçlanmaktadır[31]. İkinci yaklaşımda her projeyi ayrı ele alan ve farklı kritik yollar kullanan, daha gerçekçi bir proje çizelgeleme yaklaşımıdır[59][31]. Çok projeli problemlerde tüm projeler aynı anda başlamak zorunda değildir, mevcut projeler devam ederken yeni projelerin geliş zamanı değişken olabilir[29].

3.3.1 Çok projeli kaynak kısıtlı proje çizelgeleme probleminin modeli

Çok projeli kaynak kısıtlı proje çizelgeleme problemi şu şekildedir;

$$\text{Min } Z = \frac{\sum_{t=EFT_j}^{LFT_j} tX_{qt}}{Q} \quad (3.11)$$

$$\sum_{t=EFT_j}^{LFT_j} X_{jt} = 1; \quad j = 0, 1, \dots, n + 1 \quad (3.12)$$

$$\sum_{t=EFT_i}^{LFT_i} tX_{it} \leq \sum_{t=EFT_j}^{LFT_j} (t - d_j)X_{jt} \quad j = 1, \dots, n + 1 \text{ ve } i \in P_j \quad (3.13)$$

$$\sum_{j=1}^J k_{jr} \sum_{r=t}^{t+d_j-1} X_{jt} \leq K_r \quad r \in R, \quad t = 1, \dots, r \quad (3.14)$$

$$X_{jt} = \begin{cases} 1, & \text{eğer } j \text{ faaliyeti } t \text{ dönemi sonunda bitiyorsa} \\ 0, & \text{diğer durumlarda} \end{cases} \quad (3.15)$$

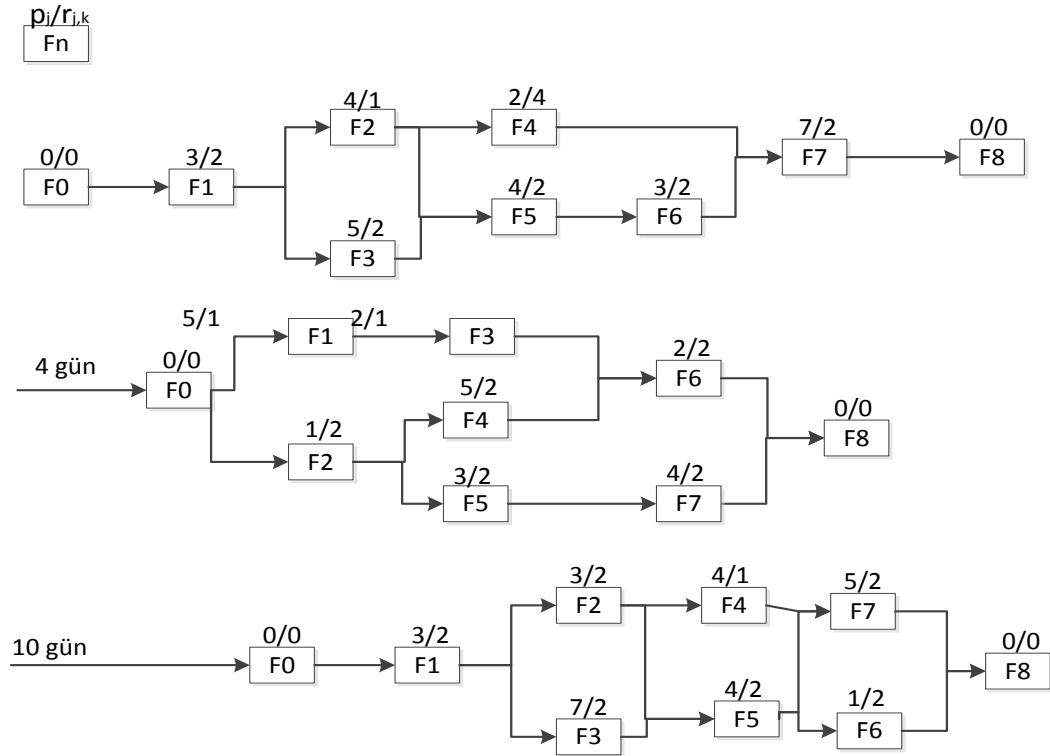
Kullanılan sembollerin tanımları aşağıda açıklanmıştır.

- t : zaman indisi (t=1,...,T),
- j : faaliyet indisi (j=1,...,J),
- q : proje indisi (q=1,...,Q),
- d_j : "j" faaliyetinin süresi
- P_j : "j" faaliyetinin öncüllerinin kümesi,
- EFT_j : "j" faaliyetinin en erken bitiş zamanı,
- LFT_j : "j" faaliyetinin en geç başlangıç zamanı,
- R : yenilenebilir kaynaklar kümesi,
- k_{jr} : j faaliyetinin r kaynağından birim zamanda kullanım miktarı,
- K_r : r yenilenebilir kaynağının birim zamanda kullanım üst sınırı,

Burada; KKMPÇP J sayıda (J=0,1,2,...,N,N+1) faaliyet içeren Q sayıda projenin kümesinden oluşur. KKMPÇP probleminin amacı formül 3.11'de gösterildiği gibi ortalama proje gecikmesini en aza indirmektir. Formül 3.12'de her bir j faaliyetinin

çizelgelenmesi sağlanmaktadır. Bir projenin $P = n+1$ adet faaliyet (veya iş yada görev) oluştuğu kabul edilir. Faaliyetler arasındaki bazı teknik gereksinimlerden dolayı, formül 3.13'te gösterildiği gibi faaliyet j , başlama bilmesi için bağlı olduğu faaliyetlerin X_{it} , $i \in P_j$, tamamlanmış olması gereklidir. 0. ve $n+1$. faaliyetin projenin başlangıç ve bitişini temsil eden temsili faaliyetler olduğu ve bu faaliyetlerin işlem süresi ve kaynak kullanımının olmadığı kabul edilir[57]. İkinci olarak, bir faaliyeti gerçekleştirmek için limitli kapasiteli kaynakları kullanmaya ihtiyaç duyar. $K = \{1, \dots, K\}$ olmak üzere K çeşit kaynak türü vardır. kaynak türü yenilenebilir ise her periyot başlangıcında kaynak türü K 'nın kapasitesinin K_r olduğu kabul edilir. Faaliyet j 'yi gerçekleştirebilmek için $k_{j,r}$ birim kaynak $K \in R$ ya gereksinim vardır[57]. Faaliyet j 'yi gerçekleştirmek için yeterli kaynak bulunmadığı durumda, kaynak seviyesi yeterli olana kadar faaliyete başlanılamaz[27].

Şekil 3.2'de her biri $n = 7$ faaliyetli, 3 projeli bir kaynak kısıtlı çok projeli bir çizelgeleme problemi verilmiştir. Şekilde de görüldüğü gibi bazı projeler başlangıç anında mevcut değildir ve başlangıcın ardından belirli bir süre sonra gelmektedir.



Şekil 3.2 Kaynak kısıtlı çok proje çizelgeleme problemi

KAYNAK KISITLI PROJE ÇİZELGELEME PROBLEMLERİ İÇİN KULLANILAN ALGORİTMALAR

Kaynak kısıtlı proje çizelgeleme problemi, araştırmacılar tarafından sıkça ele alınan ve çeşitli yöntemler kullanılarak çözülmeye çalışılmış bir problem türüdür[55]. Kaynak kısıtlı proje çizelgeleme probleminde faaliyet öncelikleri ve kaynak kısıdı olmak üzere iki ayrı kısıt olduğu için çözümü diğer programlama problemlerinden daha zordur[60].

Bu çalışmada, kaynak kısıtlı proje çizelgeleme problemine dair çözümler 3 ana başlık altında toplanabilir, bunlar deterministik yöntemler, stokastik yöntemler ve sezgisel yöntemlerdir.

4.1 Deterministik Yöntemler

Deterministik yöntemler matematiksel yöntemler olarak da adlandırılırlar. Bu yöntemlerde matematiksel ve sayısal işlemler ile en uygun çözümün aranması esasına dayanmaktadır[61][38].

Deterministik yöntemler kesin sonuçlar üretirler. Bu yöntemler kullanıldığında algoritma aynı problem için her seferinde daima aynı sonucu üretecektir. Bu özelliği nedeni ile bu yöntem türü, kısıt sayısı fazla olan karmaşık problemlerde işlem süreleri nedeni ile pratik bir yöntem olmamaktadır. Deterministik algoritmalar optimum sonucu üretemez ise alternatif çözüm üretmedikleri için başka bir çözüm üretme alternatifi olmayacaktır[16].

Kaynak kısıtlı proje çizelgelemede kullanılan deterministik yöntemlere; lineer-tamsayılı programlama, dal ve sınır algoritmaları ve sayılama teknikleri örnek verilebilir[62]

4.2 Stokastik Yöntemler

Stokastik çizelgeleme yöntemlerinde; çizelgeleme faaliyet ve aşama seçimleri olasılıklara bağlı veya rastsal yollarla yapılmaktadır. Proje çizelgeleme aşamasında her hangi bir işlemde; faaliyet öncelikleri belirlenirken veya çizelgeleme aşamasında uygun faaliyetler arasında birinin seçilmesi durumunda, olasılık kurallarına göre bir işlem yapılırsa "stokastik çizelgeleme yöntemi" olarak adlandırılmaktadır[16].

4.3 Sezgisel Yöntemler

Kaynak kısıtlı proje çizelgeleme problemi kombinasyonel işlemli problemler kümesine aittir. Bu tür problemlerin optimum çözümünü aramamıza rağmen, çoğu çizelgeleme probleminde her zaman optimum sonucu bulabilecek bir algoritma yapılamamaktadır. Bu nedenle çoğu araştırmacı bu problemi çözmek için optimal sonucu bulmayı garanti edemeyen sezgisel algoritmaları kullanmayı tercih etmektedir[63].

Kaynak kısıtlı proje çizelgeleme problemi kombinasyonel bir yapıda olduğu için orta ölçekli problemlerden itibaren en bir problemin çözümü, gelişmiş bilgisayarlar yardımıyla bile çok uzun zaman sürmektedir. Bu nedenle basitleştirilmiş modeller üzerinde sezgisel algoritmalar kullanarak optimalin altında, ancak tatmin edici sonuçlar elde edilmektedir[64].

Bu çalışmada ilk önce sezgisel yöntemlerde kullanılan çizelge yaratma yapıları incelenmiş daha sonra sezgisel yöntemler iki alt başlık altında incelenmiştir. Birinci olarak öncelik kuralı kullanarak çizelge oluşturan algoritmalar, ikinci olarak meta sezgisel algoritmalar incelenmiştir.

4.3.1 Çizelge yaratma yapıları

Çizelge yaratma yapıları (SGS) KKPÇP için birçok sezgisel algoritmanın temel elemanıdır. SGS fizibil olmak şartı ile gelişmiş güzel hazırlanmış bir başlangıç çizelgesi ile başlar. İki tür SGS yapısı vardır. Bunlar faaliyet ve zaman ataması konusunda ayrılırlar. Bunlardan biri seri SGS diye adlandırılır. Bu yapıda faaliyete zaman atanır. Diğer yapı ise zamana faaliyetin atandığı paralel SGS'dir[57].

4.3.1.1 Seri çizelge yaratma yapısı

Faaliyetlere zaman atandığı bu yapıda ilk önce faaliyetler kümelerine ayrılır. Çizelgelenen faaliyetler çizelgelenmiş faaliyetler kümesine (S_g), çizelgelenmemiş ama çizelgelenmek için gerekli öncelikli faaliyeti tamamlanmış faaliyetler çizelgelemeye uygun faaliyetler kümesine (D_g) atanırlar. Burada genellikle S_g ve D_g 'nin toplamı tüm faaliyetlere eşit olamaz çünkü çoğunlukla tüm faaliyetler çizelgelenme şartını henüz sağlamıyor durumda olabilirler. Faaliyet sayısı (n) kadar aşamadan ($g=1,2,...,n$) çizelge hazırlanır[57].

Çizelgelemeye uygun faaliyetler kümesinden bir faaliyet seçilir ve faaliyet kendisine uygun olan öncelikli faaliyetin tamamlanmış olduğu ve kaynak kapasitesinin aşılmadığı en erken zamana atanır. Bu çizelgelenen faaliyet çizelgelenmiş faaliyetler kümesine atanırken çizelgelemeye uygun faaliyetler kümesi de güncellenir. Her adımda bir faaliyet çizelgelendiği için faaliyet sayısı kadar adıma sahiptir. Bu işlem çizelgelemeye uygun faaliyetler kümesinde eleman kalmayana kadar devam eder[57].

Paralel çizelgelemenin erteleme-siz çizelgeler ürettiği, seri çizelgelemenin ise aktif çizelgeler ürettiği tespit edilmiştir. Bu nedenle aktif çözümler kümesi düzgün performans ölçütleri ile mutlaka optimum çözümü içermektedir buna karşı erteleme-siz çözümler kümesi optimum çözümü içermeye bilir. Bu sebeple seri çizelgeleme yöntemi optimum çözümü bulmaya paralel çizelgeleme yönteminden daha yakın durmaktadır[1].

4.3.1.2 Paralel çizelge yaratma yapısı

Zamana faaliyet atandığı bu yapıda da ilk önce faaliyetler kümelerine ayrılır. Çizelgelenen faaliyetler çizelgelenmiş faaliyetler kümesine (C_g) ye atanır. Çizelgelenmemiş ama çizelgelenmek için gerekli öncelikli faaliyeti tamamlanmış faaliyetler çizelgelemeye aday faaliyetler kümesine (A_g) ye atanırlar. A_g kümesi içerisinde kaynak kısıdına bakılmaksızın en erken çizelgelenebilecek faaliyetin süresi t_g' ye atanır. Kaynak kısıdına bakılarak t_g' de başlayacak faaliyetler çizelgelemeye uygun faaliyetler kümesine (D_g) ye atanırlar. D_g kümesi boş küme değilse bu kümeden bir faaliyet seçilir ve t_g zamanına atanır. D_g kümesi boş küme ise t_g bir arttırılır. $t_g = t_g + 1$ ve yeni t_g zamanı için D_g

güncellenir ve yeni kümenin boş küme olup olmadığına bakılır. D_g kümesi boş küme olmayana kadar bu işlem devam eder. Bir sonraki t_g zamanı için bu işlemler tekrarlanır, çizelgelenebilecek faaliyet kalmayana kadar bu işlemler devam eder[57].

4.3.2 Öncelik tabanlı sezgisel algoritmalar

Bir öncelik kuralı çizelgelemeye uygun faaliyetler kümesinden hangi faaliyetin seçileceğinin belirlenmesinde kullanılır. Faaliyetin seçilmesinde kullanılan karar fonksiyonunun ve en küçük veya en büyük karar değişkeninin mi seçileceğinin belirlenmesini sağlar. Bir seçim için bir veya daha çok kural birlikte kullanılabilir. En kolay seçim kuralı en kısa işlem süreli faaliyetin seçilmesidir[57]. Browning ve Yassine [31], ile Kurtulus ve Davis [59] çalışmalarında öncelik kurallı bir sezgisel yöntem kullanmışlardır.

Öncelik kuralları farklı kriterlere göre düzenlenebilir. Hesaplanacak bilginin türüne göre bağlantı, zaman, kaynak kullanımı tabanlı kurallar kullanabileceğimiz gibi alt ve üst sınır kuralları da kullanabiliriz. Alt sınır kuralında her bir faaliyet için amaç fonksiyonuna göre bir alt sınır, üst sınır kuralında her bir faaliyet için amaç fonksiyonuna göre bir üst sınır hesaplanır. Çalışılan konunun türüne göre yerel veya genel kurallar kullanılabilir. Yerel kurallarda örneğin faaliyetin işlem süresi gibi kısıtlı alanlardaki veriler incelenirken, genel kurallarda daha geniş çaplı veriler incelenir. Kurallar sabit veya değişkende olabilirler. Tüm işlemlerde sabit bir uygunluk fonksiyonu kullanılabileceği gibi her iterasyonda kendini güncelleyen bir uygunluk fonksiyonu da kullanılabilir[57].

Öncelik tabanlı algoritmalar faaliyetler çizelgelemeye uygun faaliyetler kümesi kümesinden hangi faaliyetin seçileceğinin belirlenmesinde kullanılır. Eğer öncelik tabanlı sezgisel algoritmalar sadece bir farklı çizelge üretebiliyorsa bunlara tek geçişli metotlar. Eğer algoritmalar çeşitli farklı çizelgeler üretebiliyorsa bunlara da çok geçişli metotlar denir[57].

4.3.2.1 Tek geişli algoritmalar

Tek geişli algoritmalar izelge yaratma yapısında kullanılan en eski sezgisel algoritmalardır. Bu algoritmalar sadece bir fizibil sonu retebilirler[57]. Literatrde ok sayıda ncelik kuralı nerilmiřtir. Bu kurallın en ok kullanılanları řu řekilde sıralanabilir[1]:

- i. En kısa iřlem suresi (shortest processing time - SPT).
- ii. En kısa bořluk suresi (minimum slack - MSLK).
- iii. En ok sayıda toplam ardıl (most total successor - MTS).
- iv. En byk kaynak talebi (greatest resource demand - GRD).
- v. En ge bitirme zamanı (latest finish time - LFT).
- vi. En ge bařlama zamanı (latest start time - LST).
- vii. Kaynak izelgeleme metodu (resource scheduling method - RSM).
- viii. En byk konumsal ağırlık (greatest rank positional weight - GRPW).
- ix. En kotu alternatif bořluk (worst case slack – WCS)

4.3.2.2 ok geişli algoritmalar

ok geişli algoritmalarda izelge yaratma yapısını ve ncelik kurallarını birleřtirmek iin birok farlı yol vardır[57]. ok zm treten sezgisel yntemleri ana baėlıėa ayrılabilir: rnekleme, ncelik kuralları ve Bařlangı-bitiř ıkıřlı iteratif izelgeleme. ok zm reten sezgisel yntemlerde, belirlenen sayıda izelge retilerek aralarından en iyisi seilir. Bu zm ynteminin en byk avantajı zm sresinin kısalıėıdır[1].

ncelik kuralları

En ok kullanılanı ok ncelik kurallı metodudur. Birok ncelik kuralına sahip bu yapılarda izelge yaratma yapısı defalarca ele alınır. Her seferinde ise farklı bir ncelik kuralı ile iřlem yapılır. Bu yapıda M farklı ncelik kuralının kullanılması ile M farklı izelge retilebileceėi gibi M farklı ncelik kuralından N farklı izelgede retilbilir[57].

rnekleme

En ok kullanılan bir diėer algoritma tr de rnekleme metotlarıdır. Bu metotlar yapılarında rastsal seim saėlayan bir mekanizma bulunur. Her faaliyetin seilme olasılıėı eēit olabileceėi gibi bir karar deėiřkenine gre seim olasılıkları belirlenebilir.

Karar deęiřkeninin deęeri olasılık iliřkisini direk etkileyebileceęi gibi dolaylı yoldan da etkileyebilir[57].

Bařlangıř-bitiř ıkıřlı izelgeleme

ok geiřli algoritmaların bir dięer tr bařlangıř-bitiř ıkıřlı izelgeleme metotlarıdır. Bu metotlarda izelgelenecek faaliyetler ilk faaliyetten son faaliyete veya son faaliyetten ilk faaliyete olarak seilerek izelge hazırlanabilir[57].

4.3.3 Metasezgisel yaklařımlar

Zor optimizasyon problemlerini zmek iin birok meta-sezgisel yaklařım geliřtirilmiřtir[57].

4.3.3.1 Benzetilmiř tavlama (simulated annealing)

Benzetilmiř tavlama(SA) algoritması Kirkpatrick [65] tarafından geliřtirilmiřtir. Algoritma eritilmiř katı cisimlerin soęutularak dřk enerjili formlara dnřtrlmesi sreci olan fiziksel tavlama srecinden esinlenilerek geliřtirilmiřtir. Bu algoritmada ilk sonular bulunduktan sonra, mevcut zmler biraz deęiřtirilerek yeni zmler elde edilir. Yeni zmler ncekilerden daha iyi ise kabul edilir ve yeni arařtırma sreci yeni zmler zerinden devam eder. Eęer yeni zmler ncekilerden iyi deęilse yeni zmler sıcaklık diye adlandırılan bir parametreye gre belirlenen bir olasılıkla kabul edilirler[57].

Bu sayede hedefler yerel en iyilerden kurtulmuř olur ve genel en iyiyi arama sreci devam eder. Kolay uygulanabilirlięi ve kombinatoriyal optimizasyon problemleri iin kaliteli zmler retmesine raęmen, tavlama benzetimi yntemi daha uzun bilgisayar iřlem zamanına ihtiya duyması ve parametre seimi iin ok deneme gerektirdięi iin dezavantajlıdır[66].

4.3.3.2 Tabu arama

Tabu arama(TS) algoritması Glover[67] tarafından 1989 yılında geliřtirilmiřtir. Bu algoritmada bulunan bir sonucun tm komřu sonuları incelenir ve aralarındaki en iyi sonu seilerek bu iřlem tekrarlanmaya devam edilir. Bu metotta bir yerel optimuma

takılıp sonuçların bir döngüye düşme olasılığı çok yüksektir. Tabu arama algoritması bu döngüyü kırmak için arama sürecinin kaydını bir listede tutarak aynı komşuların tekrar seçilmesini önlemeye çalışır. Teorik olarak tabu arama algoritması yeterince uzun süre çalıştırılırsa tüm ilişkileri deneyerek optimum sonucu bulacaktır. Tabu aramada uzun süre zarfında bulunan sonuçlarda iyileştirme olmuyorsa algoritma durdurulur[57].

Tabu arama algoritmasındaki listeye tabu listesi adı verilir ve bu listenin en önemli özelliklerinden birisi, bulunan en iyi çözümler tabu listesi ile karşılaştırılarak tabu listesinin güncelleştirilmesinin sağlanmasıdır. Eğer komşu çözüm adaylarından biri tabu listesindeki çözümlerden biri ile aynı ise bu çözüm değerlendirilmez. Tabu listesi oluşturulurken her döngüdeki en iyi çözüm listeye alınmakta, liste dolduğunda listedeki eski kayıtlar atılarak yeni kayıtlar listeye alınmaktadır. Tabu listesi oluşturulurken sadece elde edilen çözüm tabu listesindekilerden daha iyi ise listeye eklenir. Böylece kötü bir çözümün listeye eklenmesi önlenmiş olur[68].

4.3.3.3 Genetik algoritmalar

Holland [2] tarafından geliştirilen genetik algoritmalar bölüm 5'te detaylı olarak anlatılmıştır.

4.3.3.4 Karınca kolonisi algoritması

Gerçek karıncalar, birbirleri ile iletişim yolu olarak kullandıkları feromon olarak adlandırılan kimyasal maddeyi koklama ve salgılama yeteneğine sahiptirler. Karıncalar yuvalarından yiyecek toplamak için çıktıklarında rastsal olarak gezerler, fakat feromon izi bulurlarsa takip etmeye yada etmemeye karar verirler. Eğer takip etmeyi seçerlerse izin üstüne kendi feromonlarını salgırlar. Karıncanın diğer yollardan birini seçme olasılığı yoldaki feromon izinin gücüne bağlıdır. En güçlü feromon kokan yol genellikle karıncanın seçeceği yoldur. Zaman geçtikçe yoldaki feromonun bir kısmı buharlaşır. Koloni yuva ile yiyecek arasındaki en kısa yolu bulmadan önce tüm potansiyel yolları eşit oranda kullanırlar ve geçtikleri yollara feromon salgırlar. En kısa yoldaki karınca yuvaya yiyecek ile ilk dönen olacaktır. Aynı zamanda En kısa yol en güçlü feromon kokusuna sahip olacaktır çünkü taze feromon daha buharlaşmamış olacak ve diğer karıncalara yiyeceğe gitmek için daha çok etkileyecektir[69].

Hatta dolaşan ve tüm tepelere yapay feromon salgılayan yapay karınca kolonileriler yaratılmıştır. Yuva ilk mevki/tepe ile ve yiyecek son mevki/tepe ile temsil edilmektedir. Karıncalar ileriye gitmek için bir tepe ve komşu tepelerindeki salgılanmış feromonun gücünün bir fonksiyonu olan olasılıklı seçim kuralına göre bir tepe seçer. Karıncanın feromonu kokladığı ve salgıladığı tüm tepelerinde feromon bulunduran yol feromon matrisi olarak adlandırılır. Çözülmesi gereken sorun amaç fonksiyonlarının ve sınırlamaların bir kümesi olarak gösterilir. Yollar gezildikçe tepelerdeki salgılanmış feromonun miktarı artar. Bu feromon miktarı artışı tanımlanan yolun performansına bağlıdır. Gerçek karıncalarda olduğu gibi, tüm tepelerdeki feromonun miktarı zaman geçtikçe artar, fakat düşük performanslı yollar zamanla unutulur. Sonuç yolların fizibil olduğunu garanti etmek için, problemin kısıtları karıncaların bir sonraki tepelerden hangisini seçtiğine göre belirler[69].

4.3.3.5 Dağınık arama (scatter search) algoritması

Dağınık arama algoritması ilk kez 1970'lerde Fred Glover tarafından ortaya atılmıştır. Dağınık arama metodunun en önemli özelliği tabu arama algoritmasına bağlı olması ve arama prosedürünü geliştirebilmesidir. Dağınık arama metodunda daha güçlü bir hafıza mevcuttur. Bu sayede arama sırasında bulduğu iyi çözümleri saklar ve yeni çözümler üretilirken saklanan çözümlerden faydalanılır. Dağınık arama prosedüründe, ilk popülasyon oluşturulur. Popülasyon referans kümesi üretilir. Referans kümesinde bir altküme seçilir. Altkümeye kombinasyon prosedürü uygulanır. Kombinasyonlar iyileştirilir. Referans kümesi güncellenir. Sonlandırma gerekinceye kadar bu adımlar tekrarlanır[70].

4.3.3.6 Diğer sezgisel yöntemler

Yerel kısıt analizi; Bu yöntemde her çizelgeleme anında faaliyetlerin kısıtları yeniden değerlendirilir. Bu kısıtlar kaynak kısıtları ve faaliyet önceliği ilişkisi kısıtlarıdır. Her çizelgelemede kısıtlar güncellenir. Bu kısıtlar çizelgelenebilen faaliyetler kümesini oluşturur. Çizelgelenebilir faaliyetler kümesindeki faaliyetler, atama önceliklerine göre bir dizi önemli kural oluşturmuşlardır. Bu önemli kurallara kümenin elemanları arasında en yüksek atama önceliğine sahip faaliyetleri (Kritik Faaliyetleri) belirler. Kritik

faaliyetler çizelgelendikten sonra hala yeterli kaynak kapasitesi varsa diğer faaliyetlerde çizelgelenir[1].

Yerel arama; Bu algoritmada eldeki çözümün öncüllük ilişkileri açısından geçerli fakat kaynak kapasitesi bakımından geçerli olmayabilir. Kaynak kapasitesi ihlali durumunda bir ceza yöntemi ile ihlalin en aza indirilmesine çalışılır[1].

Budanmış dal-sınır algoritması; Dallanma sezgisel kurallar ile daraltılarak çözüm ağacının çok daha az sayıda çözüm içermesi ve kısa sürede bir sonuca ulaşması sağlanır[1].

4.3.3.7 Meta-sezgisel algoritma sonuçlarının gösterilmesi

Meta-sezgisel algoritma seçildikten sonra sonuçların gösterilmesi için uygun tasarımın seçilmesi gereklidir. Kaynak kısıtlı proje çizelgeleme problemleri için çoğunlukla çizelgenin kendisi sonuçların gösterilmesinde kullanılmaktadır. Literatürde kaynak kısıtlı proje çizelgeleme problemlerinin sonuçlarının gösterilmesi için beş farklı tasarım bulunmaktadır[57].

Faaliyet listesi tasarımı; Faaliyet listesi tasarımında faaliyetlerin sıraları sonuçların gösterilmesinde kullanılmaktadır. Bu yapıda bir faaliyet mutlaka öncelikli faaliyetinden daha sonra gelmelidir. Seri ve paralel çizelgeleme yapılarında genellikle faaliyet listesi tasarımı kullanılmaktadır. Faaliyetlerin sırası değiştirilerek yeni sonuçlar üretilmektedir. Bu tasarım genetik algoritmada çoğunlukla kullanılmıştır. Faaliyet listesi tasarımına aşağıdaki örnek verilebilir[57].

Faaliyet Sırası = {1 3 2 4 6 5 };

Rastgele anahtar tasarımı; Rastgele anahtar yaklaşımında faaliyet sırası yerine faaliyetlerin yerlerine gerçek sayı değerler yazılır. Faaliyetlerin sırası ise probleme göre küçükten büyüğe veya büyükten küçüğe olarak faaliyetler dizilerek bulunur. Faaliyetlere atanan bu değerler faaliyetin öncelik şartını sağlayacak şekilde ayarlanmıştır. Rastgele anahtar tasarımına aşağıdaki örnek verilebilir[57].

Faaliyet öncelik değerleri = { 0.08, 0.19, 0.11, 0.24, 0.40, 0.35};

Buradan faaliyet değerler küçükten büyüğe sıralanarak faaliyet sırası elde edilir.

Faaliyet sırası = {1 3 2 4 6 5 };

Öncelik kuralı tasarımı; Bu yaklaşımda çözümler kullanılan öncelik kurallarının sırası ile gösterilmektedir. Örneğin yedi elemanlı bir tasarımda yedi kural kullanılacaktır ve bu kuralların sırası sonucu belirleyecektir. Bu yaklaşım ile her faaliyet farklı bir kural ile seçilecek kural değiştikçe seçilen faaliyette beraber değişecektir. Öncelik Kuralı Tasarımına aşağıdaki örnek verilebilir[57].

Faaliyet seçim kuralı sırası = {LST, GRPW, MTS, LST, MSLK, LFT}

Kaydırma vektörü tasarımı; Kaydırma vektörü tasarımında sonuçlar kaydırma vektörleri ile temsil edilirler. Hesaplama yapılırken sonucun öncelikli faaliyetlerinden en geç biten faaliyetin süresinin sonuna kaydırma vektörü eklenerek sonuç bulunur[57].

Çizelge tasarımı gösterimi; Faaliyetler ayrık kümelerle ayrılarak gösterilirler. Öncelikli tamamlanması gerekenler, geciktirilemeyecek faaliyetler, paralel çalışabilecek faaliyetler ve esnek faaliyetler olmak üzere faaliyetler 4 farklı kümeye ayrılırlar[57].

GENETİK ALGORİTMA

5.1 Genetik Algoritma

Sezgisel tekniklerden biri olan genetik algoritma teknikleri 19. yüzyılda Charles Darwin tarafından ileri sürülen evrim teorisinden esinlenilerek Holland[2] tarafından geliştirilmiş ve Goldberg[71] tarafından daha detaylı olarak yayınlanmıştır. İsminden de anlaşıldığı gibi doğal seçilim ve genetik konseptlerini içermektedir[19]. Kalıtsal bilginin yeni nesillere aktarılması sırasında gerçekleşen çaprazlama ve mutasyon olaylarının bilgisayar ortamında modellenmesi ile genetik algoritma teknikleri doğmuştur. Çözümü güç olan rotalama, çizelgeleme, sınıflandırma, vb. gibi çok boyutlu optimizasyon problemleri genetik algoritma teknikleri sayesinde daha kolay ve hızlı bir şekilde çözülebilmektedir[16].

Genetik algoritmalar karmaşık çözüm uzayı içerisinde global optimumu arayan arama algoritmalarıdır. Önceki neslin arama sonuçlarını geliştirerek yeni nesillerin daha iyi sonuçlar bulması beklenmektedir[19]. Genetik algoritma birçok farklı problem yapısına uyarlanmasına rağmen çizelgeleme problemlerine nispeten daha az yaklaşım uygulanmıştır ve bunların birçoğu iş çizelgeleme, atölye çizelgeleme problemleri ([43],[44],[45],[46],[47],[48],[49],[50],[51],[52]) veya üretim çizelgeleme ([40][41],[42]) problemleridir[19].

Ortam koşullarına daha iyi uyum sağlayan bireylerin yeni jenerasyona daha fazla katkı yapması sayesinde jenerasyon sayısı ilerledikçe popülasyonun önceki kuşaklara göre ortama uyum sağlama becerisi artar. Ortam koşullarına uyum sağlayamayan bireyler ise yeni jenerasyonlara daha az katkı yaparlar. Bunların etkisi ise jenerasyonların sayısı

ilerledikçe azalır böylece zayıf bireyler jenerasyon sayısı ilerledikçe elenmiş olur. Bu olaya doğal seleksiyon denir. Genetik algoritmada da nesil sayısı ilerledikçe çözümler optimale yaklaşır[16].

Genetik bilginin yeni nesillere aktarılması üreme olarak adlandırılır ve doğada 2 türlü üreme vardır. Bunlar eşeyli üreme ve eşeysiz üremedir.

Eşeyli üreme, eşeyli üremede seçilen iki bireyin genetik bilgileri birbiri ile karışarak yeni bireyler oluşturur. Bireylerin genetik bilgileriyle yaptıkları bu değiş tokuşa **çaprazlama** denilir. Oluşan yeni bireylerde önceki iki bireyinde genetik bilgisinden kısımlar bulunur[16].

Eşeysiz üreme, eşeysiz üremede seçilen bir bireyin genetik bilgisi değişerek veya değişmeden yeni nesillere aktarılmaktadır. Doğada birçok tek hücreli canlıda ve bazı bitkilerde bu üreme şekli görülür. Tek bir bireyden farklı bireyler oluşması **mutasyon** sonucu olmaktadır[16].

Genetik algoritma belirlenen sayıda ilk popülasyonun yaratılması ile başlar. popülasyondaki her bir birey kromozom ile temsil edilmiştir. kopyalama, çaprazlama ve mutasyon operatörleri ile yeni nesil oluşturulur[22].

5.2 Genetik Algoritma Temel Kavramları

Bu bölümde genetik algoritma ile ilgili temel kavramlar açıklanacaktır.

5.2.1 Kromozom

Gerçek hayatta bireylerin genetik bilgisini taşıyan kromozom terimi genetik algoritma içerisinde çözümlerin temsil edilme şekli olarak biçim almıştır. Bir kromozom bir adet çözümün kodlanmış halidir. Kromozomdaki kodlama problemin türüne ve algoritmanın çalışma sistemine göre ikilik tabanda, onluk tabanda, reel sayı olarak veya harf olarak hazırlanmış olabilir[16].

Kaynak kısıtlı proje çizelgeleme probleminde, Kromozomlar birer çizelgeyi temsil ettikleri için çizelge yaratma yapıları ile oluşturulurlar ve kromozomları çizelgeyi temsil etme şekilleri Kolisch'in [57] anlattığı rastgele anahtar, faaliyet listesi, gibi yöntemler ile yapılır[37].

5.2.2 Gen

Kromozomlar gen adı verilen kısımlardan oluşmaktadır. Gen gerçek hayatta genetik bilginin bir bölümünü temsil eden parçalar demektir. Genetik algoritmada da çözümün bir bölümünü temsil etmektedir. Kaynak kısıtlı proje çizelgeleme probleminde ise gen bir adet faaliyeti temsil eder[16].

5.2.3 Popülasyon

Popülasyon gerçek hayatta bir yerdeki topluluktaki bireylerin tümünü ifade etmektedir. Genetik algoritmada popülasyon kavramı bir seferde aranılan çözüm sayısını temsil etmektedir. Bir seferde ele alınan çözüm sayısı arttıkça bulunan çözümlerin genel en iyi sonucu bulma şansı artmaktadır[16].

Popülasyon ve jenerasyon sayıları çözüm uzayında yapılacak arama sayısını göstermektedir. Büyük popülasyon ve jenerasyon sayıları optimal bulma şansını artırır fakat çözüm süresini uzattığı için bu değerlerin en uygununu bulmak gerekir. Goldberg[71], kromozom uzunluğuna bağlı olarak popülasyon değerinin ayarlanması gerektiğini önermiştir. Küçük miktarlardaki popülasyon değerleri yerel optimumlara takılmaya sebep olabilir[38].

5.2.4 Jenerasyon

Gerçek hayatta farklı zaman aralıklarında yaratılan bireyler topluluktaki jenerasyonları oluştururlar. Genetik algoritma içerisinde ise jenerasyon çözümlerin geliştirilmesini ifade etmektedir. Jenerasyon sayısı arttıkça çözümler üstünde uygulanan iyileştirme işlemleri de artmaktadır[16].

Yeni jenerasyon 4 farklı yol ile üretilebilir. Bunlar[19]:

- Elitist seçim stratejisi olarak bilinen, En iyi çizelgeler kümesinden bireylerin (P_1) değiştirilmeden yeni popülasyona taşınması,
- Çaprazlama operatörü ile yeni bireylerin (P_2) oluşturulması,
- Mutasyon operatörü ile yeni bireylerin (P_3),
- Rastsal olarak yeni bireylerin (P_4) yaratılması ile,

$P_1 + P_2 + P_3 + P_4 = P$, burada P oluşturulan yeni nesildeki çizelge sayısıdır[19].

5.2.5 Uygunluk fonksiyonu

Genetik algoritmada aranılan çözümün tanına bilmesi için bulunan çözümlerin değerlendirilmesi gereklidir. Bunun için uygunluk fonksiyonu popülasyondaki bireyleri değerlendirmek için kullanılır. Uygunluk fonksiyonu sonunda bireyler bir uygunluk puanı alırlar[16].

Uygunluk fonksiyonu, problem türüne göre değişebilir. Birden fazla hedef varsa uygunluk fonksiyonu bu hedeflerin birleşimidir. Doğrusal ölçekleme, üstel ölçekleme gibi yöntemler ile probleme en uygun uygunluk fonksiyonu seçilir[72][38].

5.2.6 Elitist seçim

Elitist seçim en iyi olanların seçilmesi anlamına gelmektedir. Uygunluk değerleri hesaplandıktan sonra popülasyonun en iyileri yeni jenerasyona hiç bir değişikliğe uğramadan kopyalanır. Bu stratejiye elitist seçim stratejisi denir. Bu yaklaşımın en önemli özelliği bulunan en iyi çözümleri kaybedilmemesi ve yeni jenerasyonların daha iyi çözümler bulmasına katkı sağlamasıdır.[22].

5.2.7 Çaprazlama

Gerçek hayatta eşeyli üreme sırasında baba ve annenin genetik bilgisi karışarak iki bireyinde bazı özelliklerini taşıyan yeni bir birey oluştururlar. Genetik algoritmada da mevcut çözüm bilgileri birleştirilerek daha gelişmiş çözümler aranmaktadır. Genetik algoritmada birçok çaprazlama türü mevcuttur[16].

Problemin tipine göre kullanılması gereken dört farklı çaprazlama operatörü bulunmaktadır[73]:

- Tek noktalı çaprazlama
- İki noktalı çaprazlama
- Çok noktalı çaprazlama
- Tekdüze (Uniform) çaprazlama

Tek noktalı çaprazlamada, rastgele seçilen kromozomlar, çaprazlama yapılacak bölge rastgele seçilerek çaprazlama yapılmaktadır[16].

İki noktalı çaprazlamada, kromozomlar iki farklı yerden kesilerek üçer parçaya ayrılmaktadırlar. Parçalar karşılıklı olarak yer değiştirilerek çaprazlama yapılmaktadır. Çaprazlama tek bir parçaya uygulanırsa iki yeni kromozom elde edilmektedir. Rastgele seçilen iki parçaya uygulanırsa, dört yeni kromozom elde edilmektedir. Çaprazlama üç parçaya da uygulanırsa altı yeni kromozom elde edilmektedir[16].

Çok noktalı çaprazlama yöntemi ise, iki nokta çaprazlama mantığına dayanmaktadır ve daha fazla noktalardan dolayı kromozomlar daha fazla parçalara ayrılmaktadır. Parçalar, çiftler arasında karşılıklı değiştirilerek yeni kromozomlar elde edilmektedir. Bu çaprazlama yönteminde de çok sayıda yeni kromozom elde edilebilmektedir. Eğer parça sayısı n ise, n_2 kadar yeni kromozom elde edilebilmektedir[74].

Çaprazlama operatörü iki ebeveyn çizelgenin birleştirilmesi ile yavru çizelgenin oluşturulmasını sağlar[19]. Proje çizelgeleme problemlerinde çaprazlama sırasında aynı faaliyetin iki kere yazılmasını önlemek ve tüm faaliyetlerin çaprazlama sonrasındaki yeni bireyde bulunmasını sağlamak için çaprazlama yöntemleri mevcuttur. Bunlar: Sıralı çaprazla ve doğrusal sıralı çaprazlama yöntemidir[3].

Bazı araştırmacılar (örn;[19][22]) çaprazlamada daha iyi çözümler bulabilmek için ebeveyn çizelgelerden birinin en kısa çözüm süresine sahip çizelgeler kümesinden seçilmesini, sadece diğer ebeveynin rastsal olarak seçilmesini tavsiye etmektedir. Bu yaklaşım daha iyi sonuçları daha kısa sürede bulabilmesine karşın rastsallığı sınırladığı için yerel optimumlara takılma ve optimum sonucu asla bulamama ihtimali oluşturur.

5.2.8 Mutasyon

Gerçek hayatta mutasyon genetik bilginin bir etken yüzünden bozulması ve farklılaşmasıdır. Mutasyon oranı gerçek hayatta çok düşüktür ancak bakteriler ve sinekler gibi çok fazla üreyen küçük canlılarda mutasyon etkilerini görmek mümkündür. Genetik algoritmada mutasyon çözüm bilgisinin bir miktar değişmesi ile yeni bir çözüme dönüşmesidir[16].

Mutasyon operatörü çaprazlama operatöründen sonra yavru kromozomlara uygulanabileceği gibi (örn. [19][22]) çaprazlama operatöründen bağımsız olarakta uygulanabilir.

Genetik algoritmada, problemin yapısına bağlı olarak aşağıdaki mutasyon operatörlerinden biri seçilebilir[73]:

- Ters çevirme
- Yer değişikliği
- Ekleme
- Karşılıklı değişim

KAYNAK KISITLI PROJE ÇİZELGELEME PROBLEMLERİ İÇİN GELİŞTİRİLEN GENETİK ALGORİTMA

Bu bölümde kaynak kısıtlı proje çizelgeleme problemlerinin (KKPÇP) çözümüne yönelik c# ve .net 3.5 kullanılarak geliştirilen genetik algoritma proje çizelgeleme programının performansının ölçülmesinde kullanılan kaynak kısıtlı proje çizelgeleme problemleri kütüphanesinin özelliklerine, genetik algoritmaların proje çizelgeleme programında faaliyetlerin temsil edilmesinde kullanılan veri yapısına, uygulamanın varsayımlarına, algoritmada veri girişinin nasıl yapıldığına, genetik algoritma programının nasıl çalıştırılacağına ve raporların nasıl inceleneceğine değinilmiştir.

6.1 Uygulamanın Varsayımları

6.1.1 Proje varsayımları

Programda kullanılan çizelgeleme algoritması Braz'ın [75] çalışmasındaki kritik yol algoritmasının sınıf yapısı temel alınarak kodlanmaya başlanmıştır.

Deneme aşamalarında bir çok farklı genetik algoritma tekniği ve yapısı denenerek en hızlı çalışan ve daha iyi sonuçlar veren bir yapı aranmıştır. Algoritmada seri çizelgeleme yapısının kullanılmasına, faaliyet listesi tasarımı ile kromozomların ifade edilmesine, çaprazlamanın sıralı çaprazlama ve kaydırma yöntemi ile yapılmasına, mutasyonun kaydırma yöntemi ile yapılmasına, yeni nesil'e aktarılabacak kromozomların elitist seçimle aktarılanlar dışında sadece çaprazlama operatörü ile aktarılmasına, mutasyon operatörünün belirli bir mutasyon şansı ile çaprazlama sonucu oluşan yeni bireylere

uygulanmasına, uygunluk fonksiyonu olarak nihai proje bitiş zamanının veya çok projeli problemler için ortalama proje gecikmelerinin alınmasına karar verilmiştir.

Kaynak kısıtlı proje çizelgeleme problemlerinin, geliştirilen programın genetik algoritma ile çizelge oluşturma esnasında kabul edilen varsayımlar;

1. En az bir kaynak türü tanımlanmalıdır.
2. En fazla 10 kaynak çeşidi tanımlanabilmektedir.
3. Her faaliyetin ismi farklı olmalıdır.
4. Projede en az iki faaliyet bulunmalıdır. Bir üst sınır yoktur. (Excel formatında sonuç alabilmek için üst sınır 32000'dir)
5. Faaliyetleri tanımlanmadan önce kaynaklar tanımlanmalıdır.
6. İlk ve son faaliyetler kukla faaliyet olarak tanımlanmalıdır.
7. Kullanılan kaynaklar yenilenebilir olmalıdır.
8. Kaynaklar proje başlama tarihinde hazır olmalıdır.
9. Kaynak kullanımı ve faaliyet süreleri tam sayı olmalıdır.
10. Bir kaynak, o kaynağı en fazla kullanan faaliyetin ihtiyacından az olmamalıdır.

Algoritmada kullanılan yenilenebilir kaynaklar: faaliyetlerin kullanım süresi ile kısıtlı olup, faaliyetlerin bitiminde tüm kapasitesi ile serbest kalan kaynaklardır. Yenilenebilir kaynak kullanımı varsayımı nedeniyle: projeye bir kaynak tanıtımı yapıldığında, kaynak türü ne olursa olsun yenilenebilir kaynak olarak kabul edilecektir.

6.1.2 Genetik algoritma varsayımları

6.1.2.1 Başlangıç toplumu

Faaliyetlerin başlangıç tarihleri kaynak kapasitesi kısıtları göz önüne alınmadan sadece öncelik kısıtları göz önünde bulundurularak, rastsal olarak belirli bir periyot içerisinde atanmaktadır. Kullanılan çizelgeleme periyodu tüm faaliyetlerin toplam süresi olan projenin en uzun bitme süresidir. Faaliyet kısıtları göz önüne alınarak rastsal olarak verilen bu başlangıç tarihleri büyükten küçüğe sıralanarak faaliyet sırasını oluşturur. Bu sayede ilk nesil oluşturulmuş olur ve faaliyet sırası kromozom olarak saklanır. Proje çizelgesi faaliyet listesi yapısı kullanılarak gösterilmektedir. Bir kromozom faaliyet sırasını temsil eder. Faaliyetler çizelgelenirken bölüm 4.3.1.1'de anlatılan seri

çizelgeleme yapısı kullanılmıştır. Genetik algoritma ayarlarında seçilen popülasyon büyüklüğü kadar başlangıç toplumu üretilir.

Çok projeli problemlerde projelerin geliş zamanı dikkate alınarak çizelgeleme yapılmaktadır.

6.1.2.2 Kromozom yapısı

Kromozom yapısı olarak faaliyet listesi tasarımı metodu kullanılmıştır. Başlangıç popülasyonundaki bir bireyin faaliyetlerin başlangıç sırası kromozom yapısındaki genler olarak temsil edilmiştir. Faaliyetlerin başlangıç ve bitiş tarihleri faaliyet sırasının kaynak kapasitelerine göre yerleştirilmesi ile bulunmaktadır.

Çok projeli problemlerde tüm projeler birleştirilerek faaliyetleri bir kromozom içinde gösterilmektedir.

6.1.2.3 İlk popülasyonun yaratılması

Genetik algoritma yapısında ilk olarak başlangıç toplumunun üretilmesi ile başlanır. Başlangıç toplumu faaliyet önceliği kurallarına uyularak rastsal olarak hazırlanmıştır. Faaliyetleri sıralamak için seri çizelgeleme yapısı kullanılmıştır. Seri çizelge yapısı aktif çizelgeler üretmektedir. Aktif çizelgeler ile yeterli arama yapıldığında optimum çözümün bulunacağı varsayılmaktadır. Bu nedenle seri çizelge yapısı optimum çözümü aramak için seçilmiştir.

Faaliyet sayısı ($g=1,2,...,n$) kadar aşama ile Çizelgelenmeye hazır faaliyetler kümesi D_g den bir faaliyet rastsal olarak seçilerek çizelgelenen faaliyetler kümesine S_g eklenir. Her işlemten sonra D_g kümesi güncellenir ve çizelgelenebilecek faaliyet kalmadığında işlem durur.

$$D_g = \begin{cases} x_j; & \text{kümeden rastsal olarak seçilen bir faaliyet çizelgeye eklenir} \\ & \text{ve küme güncellenir.} \\ \emptyset; & \text{kümede eleman kalmayınca çizelgeleme tamamlanır.} \end{cases} \quad (6.1)$$

6.1.2.4 Uygunluk fonksiyonu

Kullanılan uygunluk fonksiyonu değeri yaratılan kromozomların değerlendirilmesi ve genetik algorithmada kullanılabilmesi için derecelendirilmesinde kullanılmaktadır. Kullanılan uygunluk fonksiyonu türüne göre minimizasyon (en küçükleme) veya maksimizasyon (en büyükleme) olabilir. Bireylerin uygunluk değerleri, proje sürelerine göre hesaplanmıştır. Algorithmada maksimizasyon yapısında ki iki uygunluk fonksiyonu kullanılmıştır. Bunlar süre uygunluğu ve ortalama proje gecikme süreleridir.

Süre uygunluğu:

Süre uygunluk fonksiyonu aşağıda belirtildiği gibi hesaplanmaktadır.

$$F_{süre} = \frac{t - t_{min}}{t_{min}} \quad (6.2)$$

Burada;

$F_{süre}$: Faaliyetin süre uygunluğu

t_{min} : projenin kaynak kapasitesi dikkate alınmadan kritik yoldaki tamamlanma süresi

t : n. kromozomum proje tamamlanma süresidir.

$$P_{gecikme} = \sum_{i=1}^N (P_{i,bitiş} - P_{i,başlangıç}) / N \quad (6.3)$$

$$F_{gecikme} = \frac{P_{gecikme} - P_{min}}{P_{min}} \quad (6.4)$$

Burada;

$P_{gecikme}$: Ortalama proje gecikmesi,

N : Proje sayısı,

$P_{i,bitiş}$: i'inci projenin bitişi,

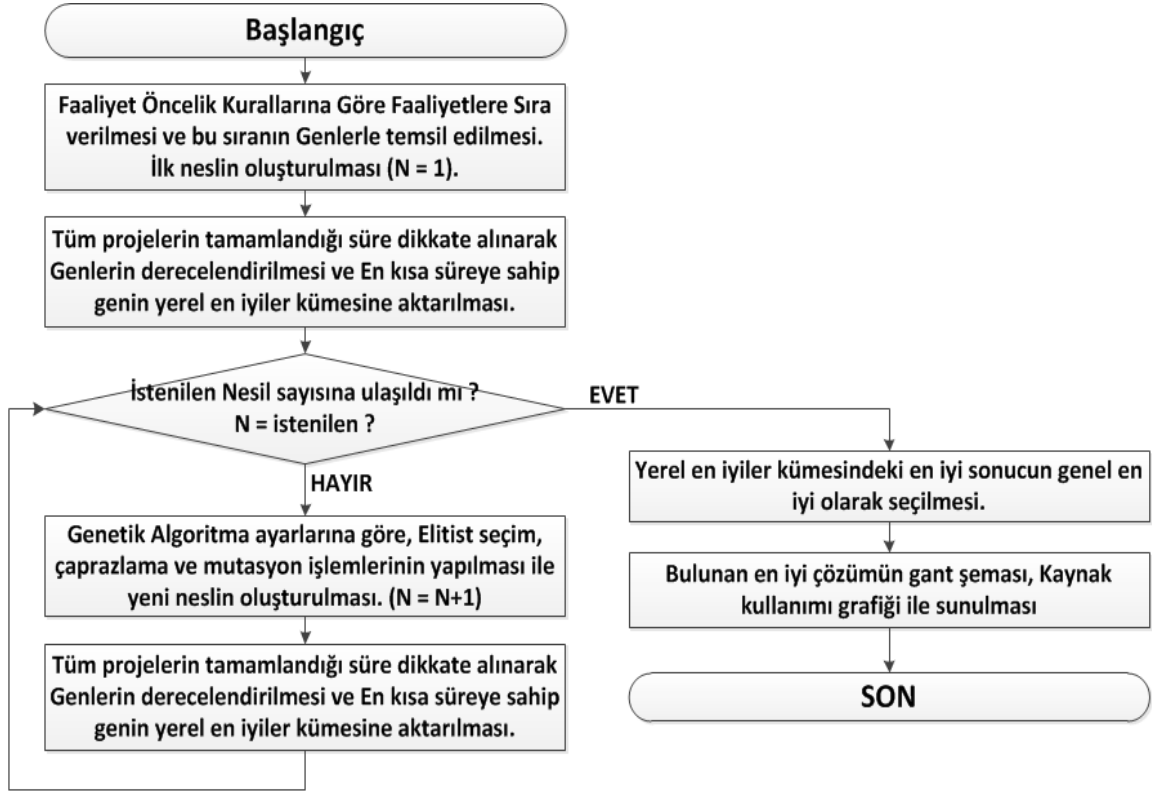
$P_{i,başlangıç}$: i'inci projenin başlangıcı,

$F_{gecikme}$: Projenin ortalama gecikme uygunluğu,

P_{min} : Kapasite dikkate alınmadan kritik yoldaki ortalama proje gecikme süresi.

6.1.3 Projede kullanılan genetik işlemler

Şekil 6.1’de görüldüğü gibi yeni popülasyonun oluşturulurken bir miktar birey önceki popülasyonun en iyileri doğrudan dahil olur. Kalan bireyler ise rastsal seçim ile seçilip tek noktalı çaprazlama metodu ile çaprazlanırlar, daha sonra belirli bir olasılıkla tek noktalı mutasyona uğrayarak popülasyona katılır.



Şekil 6.1 Genetik işlemler akış şeması

Hazırlanan algoritmada ilk önce çaprazlama ve mutasyon operatörleri birbirlerinden bağımsızdı. Yeni neslin bir bölümü çaprazlama ile bir bölümü ise doğrudan mutasyon ile oluşturulmaktaydı. Daha iyi çözümler bulmak daha kısa işlem süresinde istenilen sonuçları bulabilmek adına bu yöntem ile başka bir teknik olan mutasyon operatörünün belirli bir olasılıkla çaprazlama operatöründen oluşan bireylere uygulanması yöntemi karşılaştırılmıştır. Tablo 6.1’de görüldüğü gibi ile mutasyon operatörünün çaprazlama operatöründen sonra kullanılmasının daha iyi sonuçlar verdiği daha kısa işlem süresi sürdüğü ve daha erken nesillerde en iyi sonucu bulduğu görülmüştür. Bu nedenle mutasyon operatörü çaprazlama operatörüne belirli bir olasılık ile bağlanmıştır.

Çizelge 6.1 Farklı çaprazlama ve mutasyon yöntemlerinin karşılaştırılması

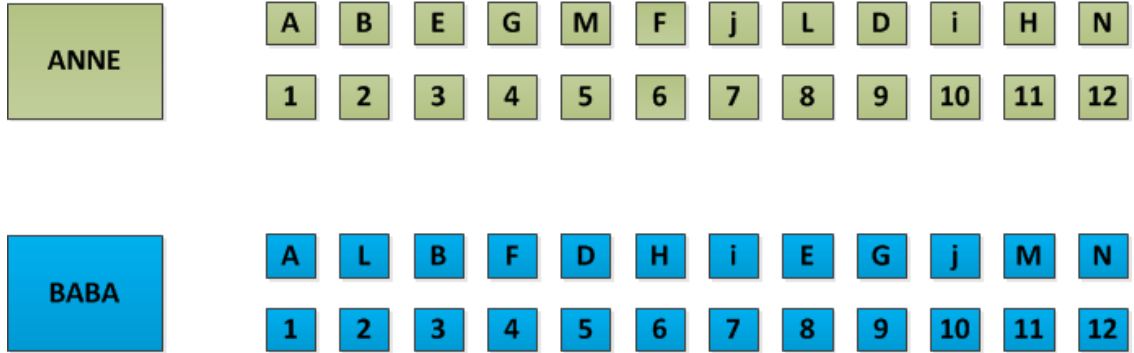
Eski Yöntem	Örnek No	1	2	3	4	5	6	7	8	9	10	Ort
	İşlem Süresi	1219	1103,5	720,5	1198,5	317,5	614,5	1091,5	854,5	578,5	482,5	818,05
	Proje süresi	182	315,5	161	346	67	182	327	166,5	347,5	94,5	218,9
	Sonucun bulunduğu Nesil											
		585,5	780,5	329	866	1	459,5	869	501	833,5	214	543,9
Yeni Yöntem	Örnek No	1	2	3	4	5	6	7	8	9	10	Ort
	İşlem Süresi	679	545	628,5	640,5	495	769,5	523	489,5	544,5	474	578,85
	Proje süresi	181,5	296	167	343,5	67	181,5	316	170,5	345	98,5	216,65
	Sonucun bulunduğu Nesil											
		299	506	403,5	686,5	1	198,5	398	195,5	956,5	268	391,25

Gotshall ve Rylander [76], çalışmalarında proje boyutuna göre ideal popülasyon oranını araştırmıştır. Proje boyutu ile popülasyon büyüklüğü arasında $y = 267.43 \ln(x) - 293.21$ oranını önermişlerdir. Goldberg [71], kromozom uzunluğuna bağlı olarak popülasyon değerinin ayarlanması gerektiğini önermiş ve birçok araştırmacıda popülasyon büyüklüğü olarak kromozom uzunluğunu yada kromozom uzunluğunun iki katını alarak kıyaslamalarını yapmıştır. Bu nedenle çalışmada kromozom sayısı kadar veya kıyaslama yaptığımız çalışmalardaki kadar popülasyon büyüklüğü alınmıştır.

6.1.3.1 Çaprazlama operasyonu

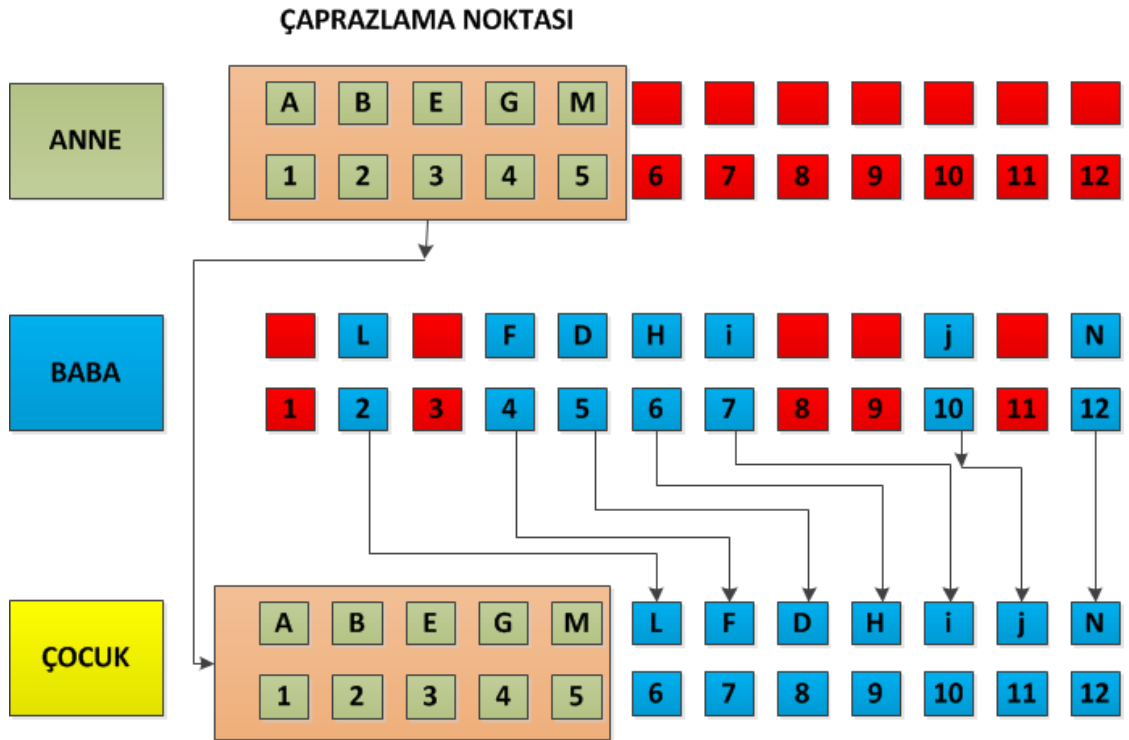
Algoritmada tek noktalı çaprazlama operatörü kullanıldı. Davis[77]'in önerdiği sıralı çaprazlama yöntemi ile çaprazlama yapıldı. Çaprazlama için seçilen tüm ebeveynler rastsal olarak seçildi böylece yerel optimumlara takılma önlenecektir. Çaprazlama sonrası oluşan yavru belirli bir olasılıkla mutasyon operatörüne girmektedir. Mutasyon operatörüne girmeyen yavru bireyler doğrudan yeni nesil'e aktarılır.

Faaliyet Adı	A
Faaliyet Numarası	1



Şekil 6.2 Çaprazlama için seçilen iki ebeveyn kromozom

Çaprazlamada bir ebeveyn çizelgenin belli bir bölümü seçilmekte ve yavru çizelgeye aktarılmaktadır. Eksik kalan faaliyetler diğer ebeveyn çizelgedeki sırası ile yavru çizelgeye eklenir. Bu yöntem sayesinde yeni çizelge, faaliyet öncelik ilişkisi korunarak oluşturulur[19][3]. Şekil 6.2 ve şekil 6.3'de Bu çaprazlamaya örnek verilmiştir.



Şekil 6.3 Sıralı çaprazlama operasyonu örneği

6.1.3.2 Mutasyon operasyonu

Algoritma'da tek noktalı mutasyon operatörü kullanılmıştır. Mutasyon operatörünü belirlenen olasılığa göre çaprazlamadan oluşan yavruyu seçmektedir. Mutasyon için seçilen faaliyet kendi başlangıç zamanı ile kendisinin bağlı olduğu en geç tamamlanan faaliyetin bitiş zamanı arasındaki bir zamana rastsal olarak atanarak faaliyet ilişkisi bozulmadan mutasyon yapılmaktadır. Mutasyon sonrası oluşan birey yeni popülasyona katılır. Algoritmada kullanılan mutasyon örneği şekil 6.4'te gösterilmiştir.



Şekil 6.4 Basit bir mutasyon operasyonu örneği

Hazırlanan algorithmadan en yüksek performansı alabilmek için en uygun genetik algoritma ayarlarının bulunması gereklidir. Bunun için farklı mutasyon oranlarını denemiş ve en iyi oranın %50 mutasyon şansı kullanılması olduğu görülmüştür. Tablo 6.2'de farklı mutasyon oranlarına göre elde edilen sonuçlar gösterilmektedir.

Çizelge 6.2 Çeşitli mutasyon oranlarının denenmesi

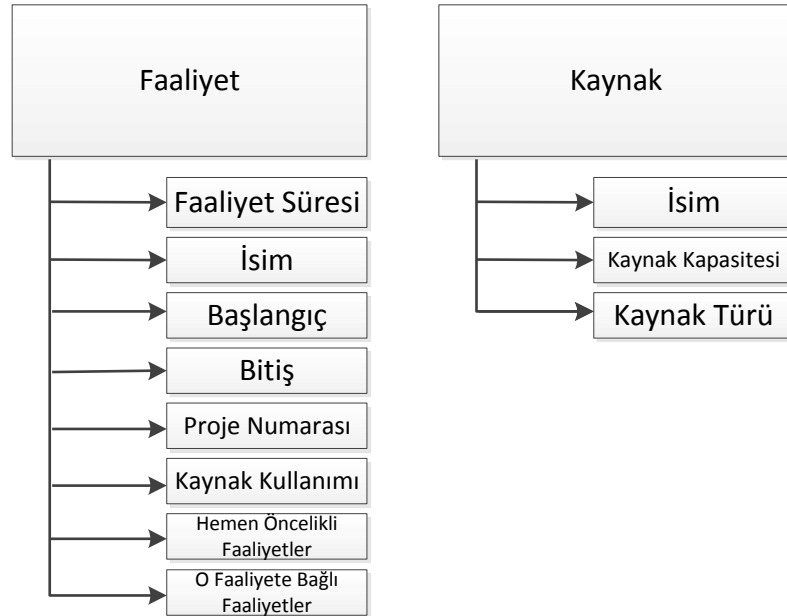
Mutasyon oranı	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Ort. işlem süresi	486,5	527	532,3	579,1	575,9	582,5	591,4	576,3	562	537
Ort. proje süreleri	217,9	222,4	218,9	215,2	213,1	218,6	217,2	215,8	220	217
Ort. Gerekli nesil	345,8	357,8	409,6	444,5	461	534,7	317,5	279,6	453	403

6.2 Uygulamanın Veri Yapısı

Önerilen algoritma'da proje ve kaynakları sınıf yapısı ile temsil edilmiştir. Hazırlanan program öncelikle kaynak sınıfında kaynak türlerinin yaratılmasını isteyerek, bu kaynak türlerini ve çeşitlerini proje sınıfına eklenir. Böylece kaynakları önceden tanımlayarak projenin olmayan bir kaynağı kullanması önlenir.

Hazırlanan projenin saklanmasını sağlamak için kayıt özelliği eklenmiştir. Böylece hazırlanan proje bilgisi ayrı bir ortamda saklanmasına olanak verilmektedir. Ayrıca proje bilgisi birçok kaynakta depolanabileceği için sınırsız bir saklama ortamı sağlanmış olur.

Kaynak sınıfının içerisinde isim, kaynak kapasitesi, kaynak türü tanımlanmalıdır. Faaliyet sınıfının içerisinde ise, faaliyet süresi, isim, başlangıcı, bitişi, kaçınıcı proje olduğu, kaynak kullanımı, hemen öncelikli faaliyetleri ve o faaliyette bağlı faaliyetler tanımlanmaktadır. Şekil 6.5'te faaliyet ve kaynak sınıflarının elemanları gösterilmektedir. Kaynak ve faaliyet bilgileri Bu sınıflarda saklanarak gerektiğinde kullanılabilir veya değiştirilebilmektedir.



Şekil 6.5 Projede kullanılan faaliyet ve kaynak yapısı

6.3 Uygulamanın Veri Giriş Yapısı

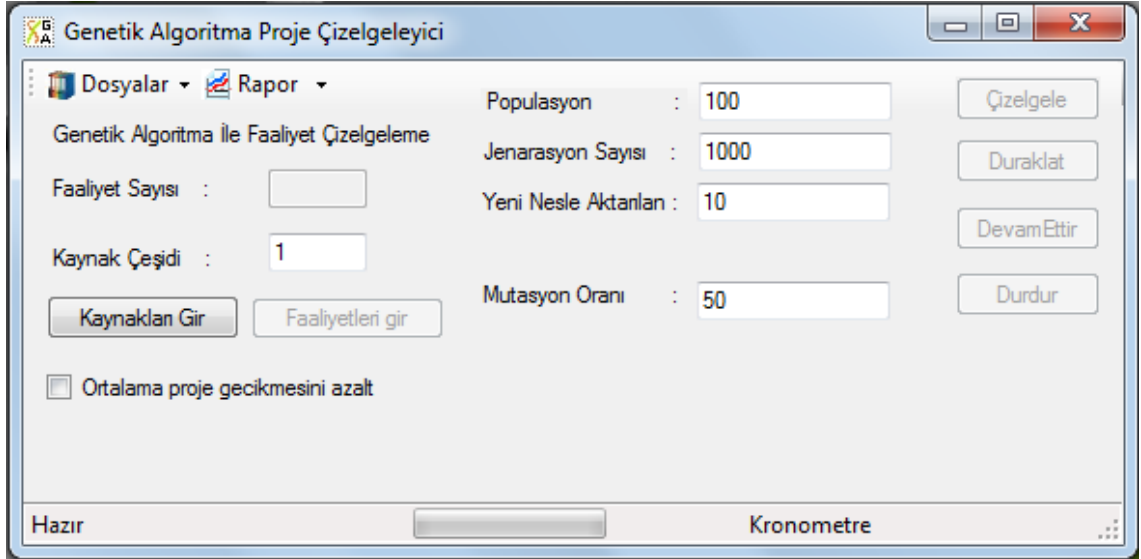
Yeni bir proje bilgisi girmek için ilk önce kaynak bilgileri tanımlanmalıdır. Böylece faaliyet bilgileri girilirken kaynak isimleri ekrana gelecek ve kaynak kullanım miktarlarının girilmesi istenecektir. Algoritmada yapı gereği faaliyetler öncülden ardıla şeklinde girilmelidir. Öncül faaliyetlerin önce girilmesi sayesinde ardıl faaliyete sıra geldiğinde öncüller bölümünden sadece işaret koyarak öncül-ardıl ilişkisi kolayca kurulabilecektir. Bu özelliğin en büyük avantajı faaliyetin öncül ilişkisinin faaliyet tanımlanırken yapılmasıdır. Bu sayede öncül ardıl ilişkisini kurabilmek için tüm faaliyetlerin girilmesi ihtiyacını ortadan kalkmaktadır.

Literatürdeki problemlerde problemin tanıtımı ardıl ilişkisi verilerek yapılmıştır. Kütüphane problemlerinde proje bilgisini girmek için karmaşık dönüştürme işlemi yapmaya gerek yoktur. Programda karmaşık dönüştürme işlemlere girilmeden literatür problemi yüklene bilmektedir.

Çok Projeli bir problemi çözmek için bir tüm proje dosyalarının içinde bulunduğu ".zip" uzantılı dosyası programa yüklenmelidir. Bu sayede bir farlı projelerin bilgilerini içeren dosyalar aynı dosya altında saklanabilmektedir.

6.3.1 Yeni bir proje bilginin girilmesi

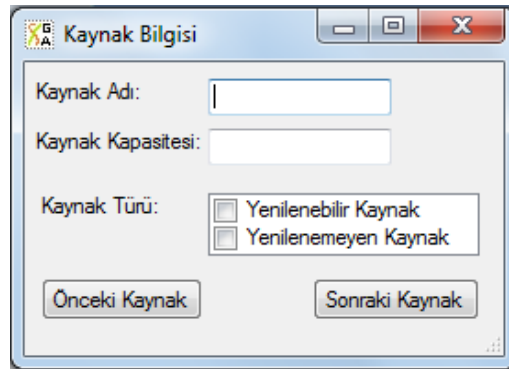
Yeni bir proje girmek için ilk önce kaynak bilgileri tanımlanmalıdır. Böylece “Faaliyet gir” düğmesi aktifleşir. Aynı zamanda Faaliyet Gir formuna girilen kaynakların bilgileri gelecek ve kullanım miktarlarının girilmesi istenecektir. Şekil 6.6’da görüldüğü gibi yeni projenin kaynak bilgilerini girmek için “Kaynak Çeşidi Gir:” bölümüne projede bulunan kaynak çeşidi sayısı girilmelidir. Daha sonra “Kaynak Gir” düğmesine basılarak kaynak bilgilerin ve kapasitelerinin girilmesine geçilir.



Şekil 6.6 Genetik algoritma proje çizelgeleyici ana ekranı

Bu ekran üzerinde Proje süresi boyunca kullanılan tüm kaynakların kaynak türü "Kaynak Kapasitesi"nin altında bulunan "Kaynak Türü" bölümünden "Yenilenebilir Kaynak" ve "Yenilenemeyen Kaynak" türlerinden birinin yanına işaret konularak seçilmelidir. İki tür kaynak türü tanımlanmıştır. Bunlardan birincisi, her gün, gün sonunda yenilenen her gün tekrar kullanılabilen yenilenebilir kaynaklardır. Yenilenebilir kaynaklara işgücü-makine-ekipman örnek verilebilir. Diğer kaynak türü ise proje başında miktarı belirlenen ve proje süresince bu miktarı kullanan kullanıldıktan sonra azalan yenilenemeyen kaynak türüdür. Hammadde ve sarf malzemeler Yenilenemeyen kaynak türüne örnek verilebilir.

Şekil 6.7’te görüldüğü gibi “Kaynak Gir” düğmesi basıldığında “Kaynak Bilgisi” ekranı açılır. Bu ekranda girilecek kaynağın adı ve kapasitesi girilir. Kaynağın türü "Yenilenebilir Kaynak" veya "Yenilenemeyen Kaynak" türlerinden biri olarak seçilir.



Şekil 6.7 Genetik algoritma proje çizelgeleyici kaynak bilgisi ekranı

Ana menüye döndüğünde şekil 6.8’de görüldüğü gibi “Faaliyet Sayısı Gir:” bölümü ve “Faaliyetleri Gir” düğmesi aktifleşmiştir. “Faaliyet Sayısı Gir:” bölümüne projedeki faaliyet sayısını girilir ve faaliyetlerin ayrıntılarını girebilmek için “Faaliyetleri Gir” düğmesine tıklanır.

Şekil 6.8 Genetik algoritma proje çizelgeleyici kaynak bilgisi girildi ekranı

Şekil 6.9’da görüldüğü gibi "Proje Bilgileri" ekranında faaliyetin adı, faaliyetin süresi, öncelikli faaliyetleri, kaynak kullanımları bilgileri girilir. İşlem süreleri gün cinsindendir. Başlangıç ve bitiş faaliyetlerinin işlem süresi sıfır olarak girilmelidir. Girilen her faaliyet “Öncelikli Faaliyetler:” bölümünde görülecektir. Böylece öncelik ilişkisini kurmak için yanlarındaki kutucuğu işaretlenir. Genetik algoritma proje çizelgeleyici 10 farklı kaynağa kadar destek vermektedir. Kaynak sayısı daha fazla ise kod’da yapılacak küçük bir değişiklikle kaynak sayısı arttırılabilir.

The screenshot shows a window titled "Proje Bilgileri" (Project Information). It contains the following elements:

- Activity Name:** A text input field labeled "Faaliyet Adı :".
- Activity Duration:** A text input field labeled "Faaliyet Süresi :".
- Priority Activities:** A large empty text area labeled "Öncelikli Faaliyetler:". Below it are two buttons: "Önceki Faaliyet" (Previous Activity) and "Sonraki Faaliyet" (Next Activity).
- Resource Usage Table:** A table titled "Kaynak Kullanımı" (Resource Usage) with the following data:

Kaynak	Kullanım
mühendis	2
usta	0
işçi	0
Toma	0
Freze	0

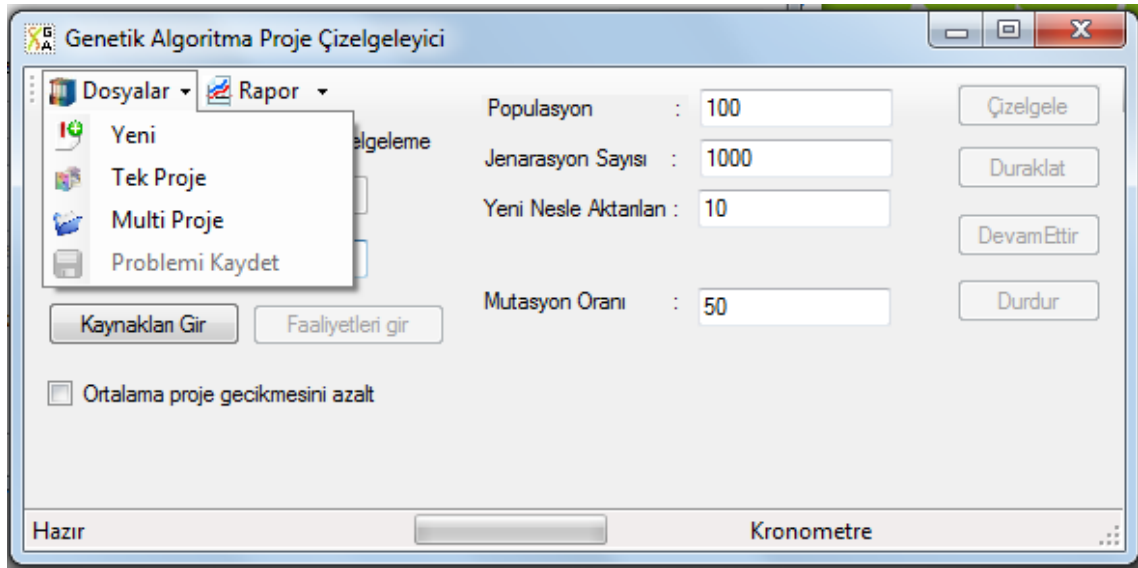
Şekil 6.9 Genetik algoritma proje çizelgeleyici proje bilgisi ekranı

6.3.2 Mevcut bir proje bilgi kaydının açılması

Genetik algoritma proje çizelgeleyici programında iki farklı kayıt açma yöntemi vardır. Bunlardan birincisi programın tek projeli kaynak kısıtlı proje çizelgeleme problemi kayıt türü olan ".sm" uzantılı dosyaları açmakta kullanılan metodudur. Şekil 6.10'de görüldüğü gibi "Tek Proje" seçeneği seçilerek ilgili faaliyet bulunduğu konumdan açılır.

Diğer kayıt açma yöntemi ise çok projeli kaynak kısıtlı proje çizelgeleme problemi dosyası olan ".zip" uzantılı dosyaları açmaya yarar. Bu seçenek şekil 6.10'te gösterilen "Multi Proje" butonudur.

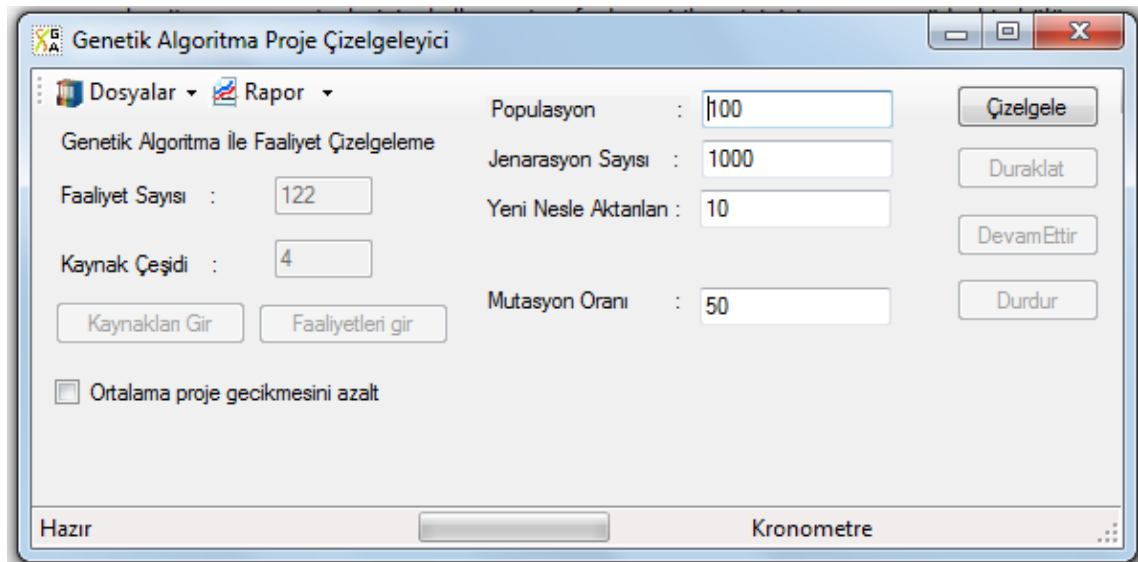
Bu dosya formatları sırası ile Kolisch ve Sprecher'in [5], ve Homberger'in [26] çalışmalarındaki çalışmalarında oluşturdukları kaynak kısıtlı proje çizelgeleme ve çok projeli kaynak kısıtlı proje çizelgeleme problemleri kütüphanelerinin örnek formatıdır. Sezgisel algoritma kütüphanelerinde bulunan binlerce örnek proje üzerinde programı deneme ve sonuçları değerlendirme imkânına sahip olunmuştur.



Şekil 6.10 Genetik algoritma proje çizelgeleyici faaliyet aç bölümü

6.4 Uygulamanın Genetik Algoritma Ayarları

Uygulamada popülasyon sayısı, jenerasyon sayısı, Mutasyon oranı, gibi genetik algoritma parametrelerinin kullanıcı tarafından girilmesini için ana menüde bir bölüm hazırlanmıştır. Bu sayede kaynak kısıtlı proje çizelgeleme problemlerinin değişik kısıtlar altında test edilebilmesine imkân sağlanmıştır.



Şekil 6.11 Çizelgelemeye hazır ana ekran

Şekil 6.11’de kullanıcı tarafından girilebilen gerekli genetik algoritma parametreleri gösterilmektedir. Bu değerler:

Popülasyon: faaliyet öncelikleri dikkate alınarak hazırlanan çözümlerden her biri popülasyon içerisindeki bir bireyi oluşturur. Popülasyon bölümüne girilen değer

yaratılacak birey sayısıdır. Birey sayısı arttıkça optimal çözümü bulma şansı da artar fakat çözüm süresi de uzar.

Jenerasyon sayısı: genetik algoritmanın yapısı gereği bir popülasyondaki bireylerden en iyi bir grup hiç değişmeden, rastsal olarak seçilen bir grup kendi arasında bilgi değişimiyle, rastsal olarak seçilen başka bir grupta kendi başına bilgisinin değişmesi ile yeni popülasyona katılır. Bu döngü böylece devam ederek her bir popülasyon başka bir popülasyon yaratır. Jenerasyon sayısı oluşturulan her bir popülasyonun sayısıdır. Jenerasyon sayısı arttıkça yapılan iyileştirme sayısı arttığı için optimum çözümü bulma şansı artar fakat çözüm süresi uzar.

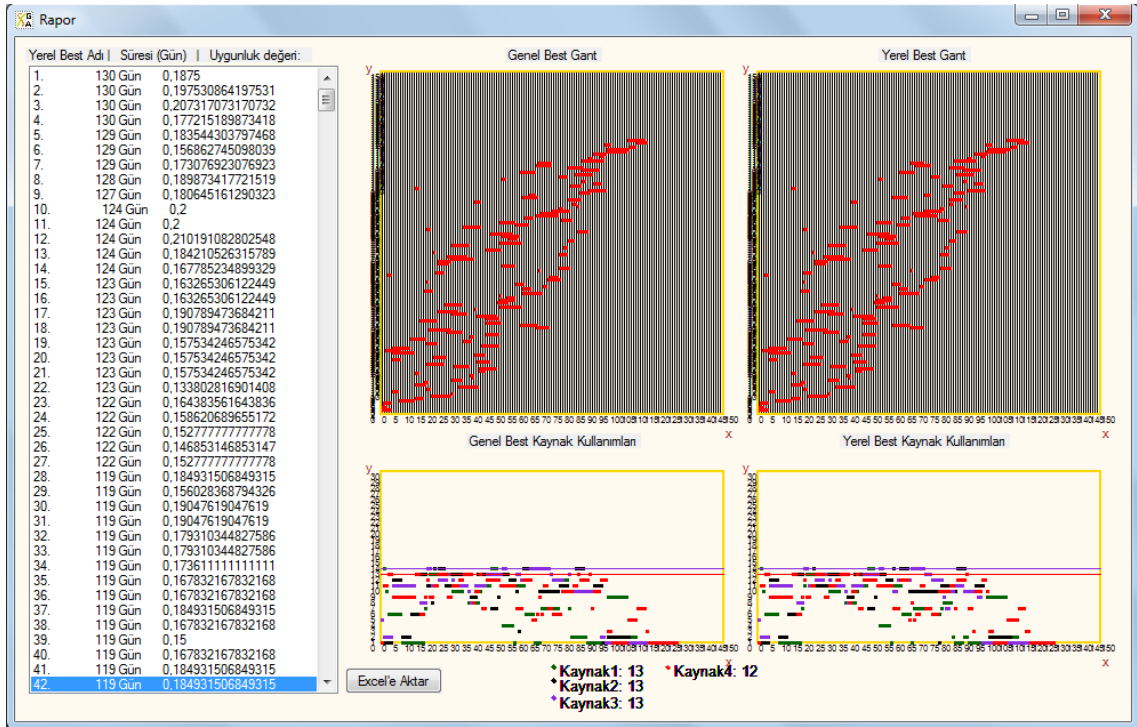
Yeni nesle aktarılan: bir popülasyondaki çözümler aldıkları puanlara göre en iyiden en kötüye sıralanırlar yeni nesle aktarılan miktarı çözümlerin yüzde cinsinden ne kadarının yeni nesle değişmeden aktarılacağı belirlenmesini sağlar. Yeni nesle aktarılanlar popülasyondaki en iyi bireylerden seçilirler. Bu seçime elitist seçim denir.

Mutasyon oranı: çaprazlama operatörü sonucunda oluşan bireyler belirlenen olasılığa göre mutasyon operatörüne gönderilirler. Mutasyon operatöründe tek noktalı mutasyon yapılmaktadır. Mutasyon işlemi daha önce 5.1.3.2'nolu bölümde anlatılmıştır. Mutasyon oranı, mutasyona maruz bırakılma oranını belirler.

Ortalama proje gecikmesini azalt: Uygunluk fonksiyonunu toplam proje tamamlanma süresini en aza indirmeden, ortalama proje gecikme süresini en aza indirmeye çeviren seçenek bölümüdür.

6.5 Uygulaman Sonuç Rapor Bilgileri

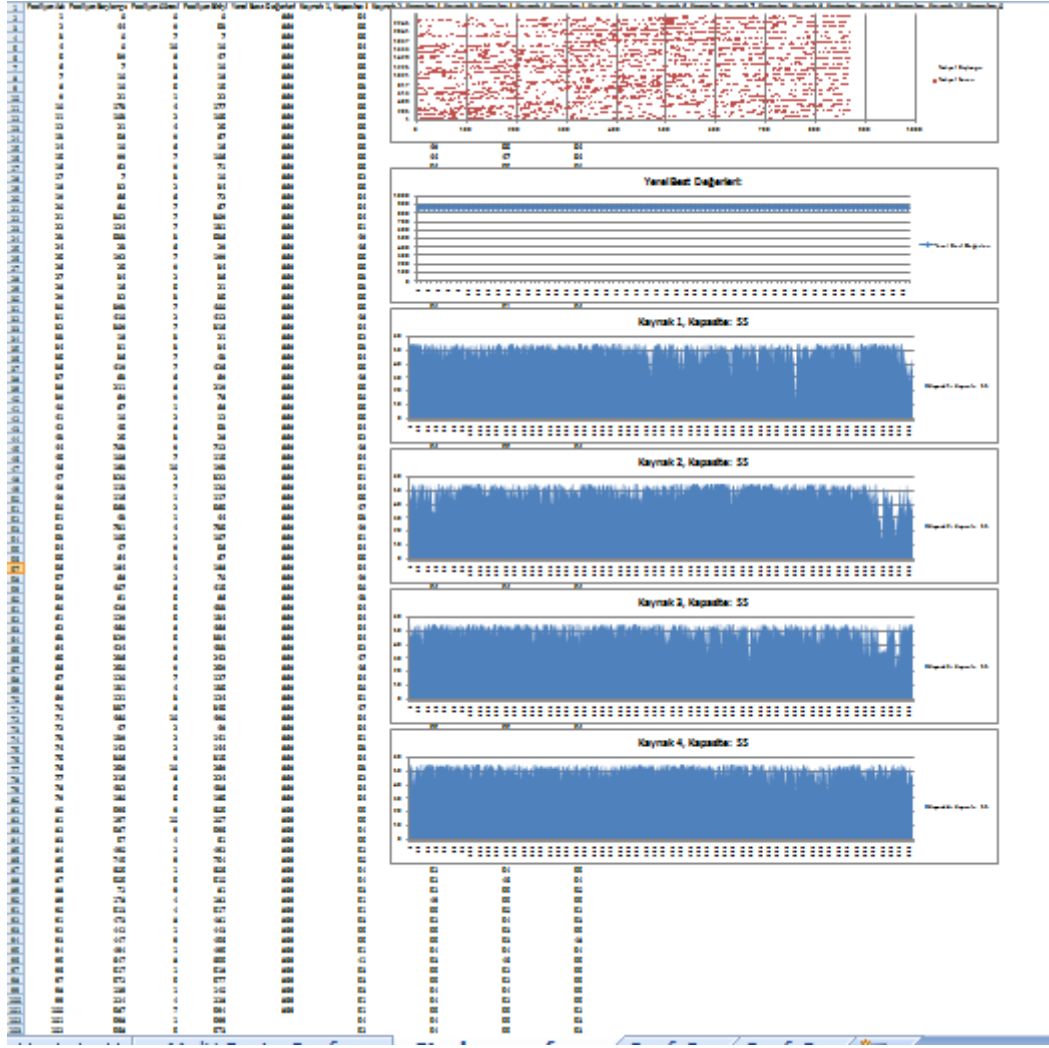
Uygulamanın rapor ekranı şekil 6.12'da görülmektedir. Bu bölümde her bir Jenerasyondaki en iyi değer yerel best olarak adlandırılır. Tüm jenerasyonlar içerisinde ise en iyi değer ise genel best olarak adlandırılır. Rapor bölümünden yerel en iyilerin listesinden herhangi biri seçildiğinde onun gant şeması ve kaynak kullanım grafiği genel en iyinin grafikleri ile birlikte ekrana gelmektedir. İstenirse seçilen bir yerel en iyi değer "Excel'e Aktar" düğmesine basılarak excel dosyası halinde sonuç raporu oluşturulur.



Şekil 6.12 Rapor ekranı

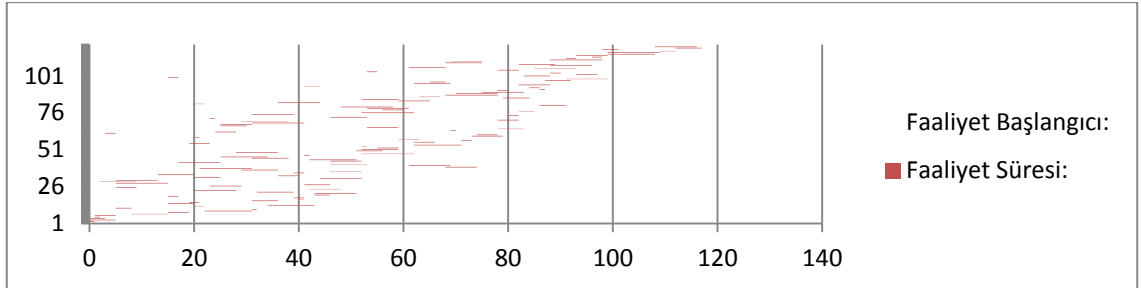
Excel'e aktarılan rapor dosyasında seçtiğimiz yerel en iyi değer vardır. Çizelgeleme bittikten sonra açılan excel'de ise genel en iyi değer vardır. Son neslin yerel en iyi değeri genel en iyi değeri olduğu için bu değer seçilerek de yerel en iyi değer raporu alınabilir.

Excel raporunda şekil 6.13'te görüldüğü gibi seçilen yerel en iyi sonucun veya genel en iyi sonucun raporunda; ilk sütunda faaliyet isimleri, ikinci sütunda faaliyet başlangıcı, üçüncü sütunda faaliyet süresi, dördüncü sütunda faaliyet bitiş zamanı, beşinci sütunda tüm yerel en iyi değerleri, altıncı satırdan sonra ise zamana göre kaynak kullanım miktarları verilmiştir.



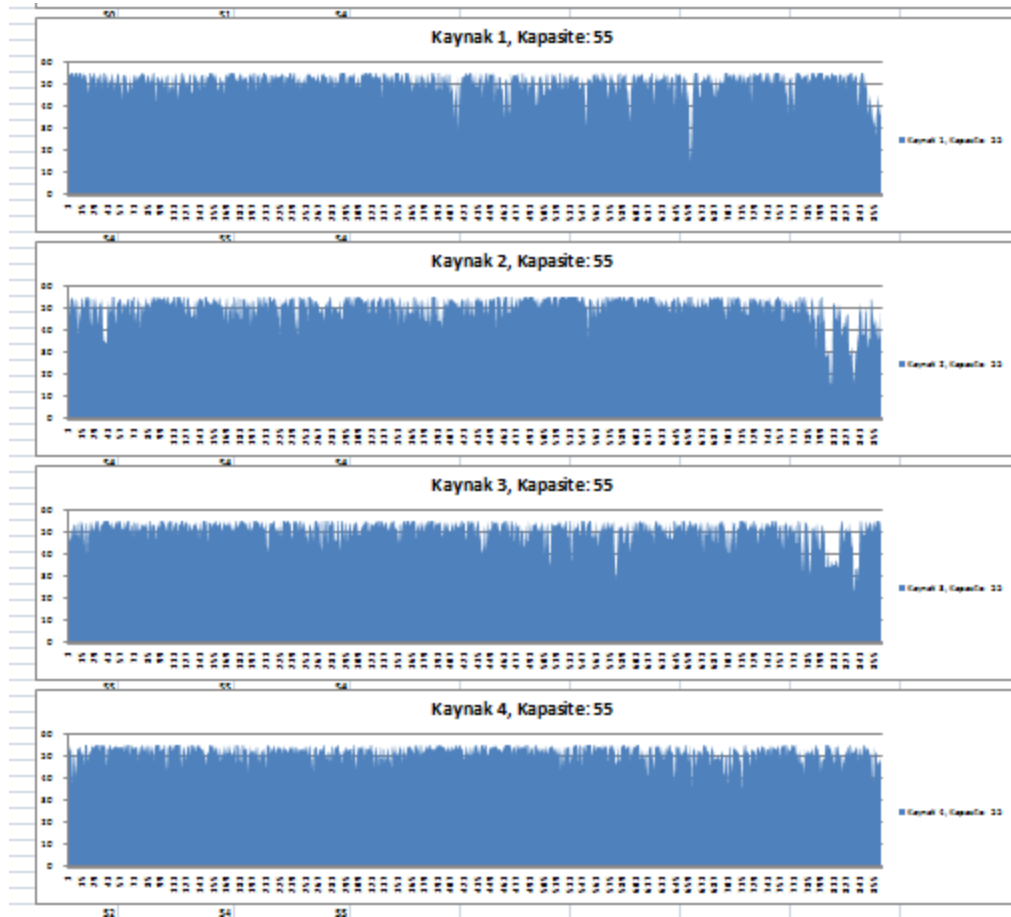
Şekil 6.13 Excel rapor sayfası

Excel sayfasındaki gant şeması oluşturmak için mavi kısmın dolgusu kaldırılır böylece faaliyet başlangıcı görünmez ve tablo bir gant şemasına dönüşür. Gant şeması faaliyetlerin zamanlarını ve önceliklerini gösterir.



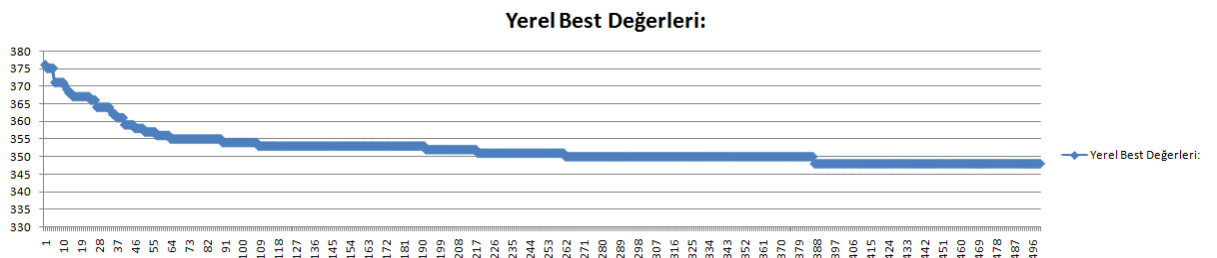
Şekil 5.14 Excel'de gant şeması

Şekil 6.15'te görüldüğü gibi kaynak kullanımı grafiği ne zaman, hangi kaynakta ve ne kadar kullanım olduğunu gösteriyor. Aynı zamanda bu grafik maksimum kaynak kullanımı ve toplam kaynak kullanımı değerlerini de görmektedir.



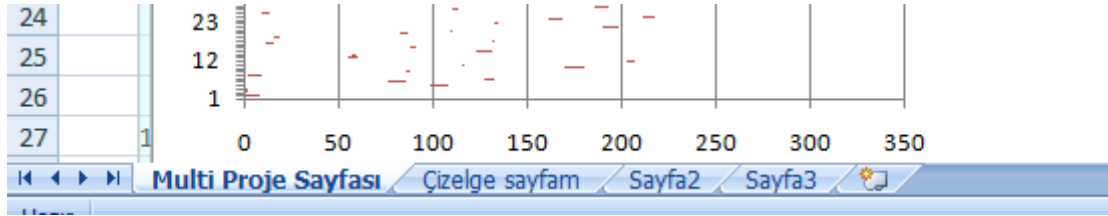
Şekil 6.15 Excel'de kaynak kullanımı grafiği

Şekil 6.16'te gösterilen yerel en iyi değerleri grafiği ise genetik algoritmanın oluşturduğu sonuçların zaman cinsinden değerlerini göstermektedir. Bu grafik yerel en iyi değerlerinin hangi nesillerde değiştiğini gösterir. Böylece bu sonuçların kümesi karar verici için tercih yapma imkânı sağlar.

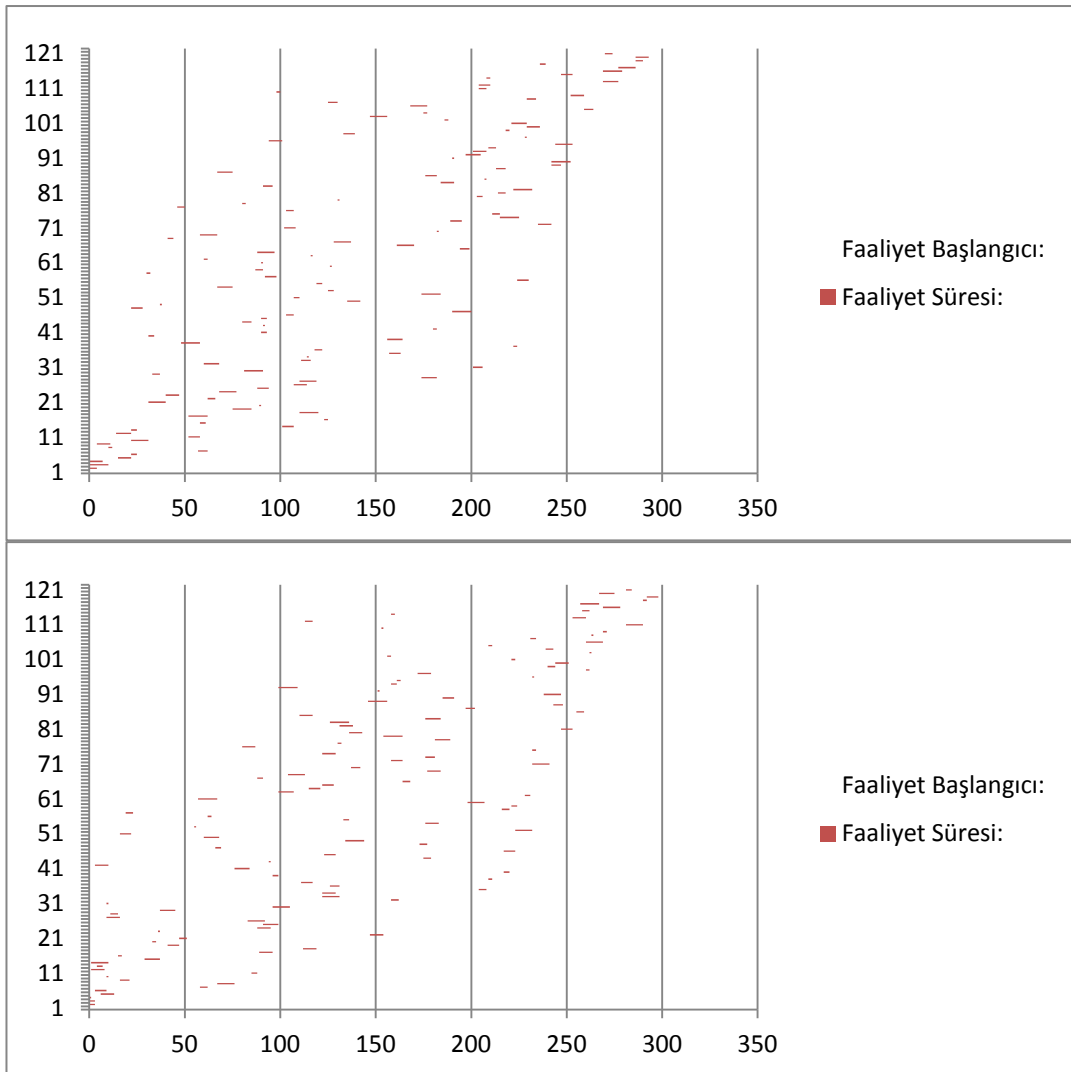


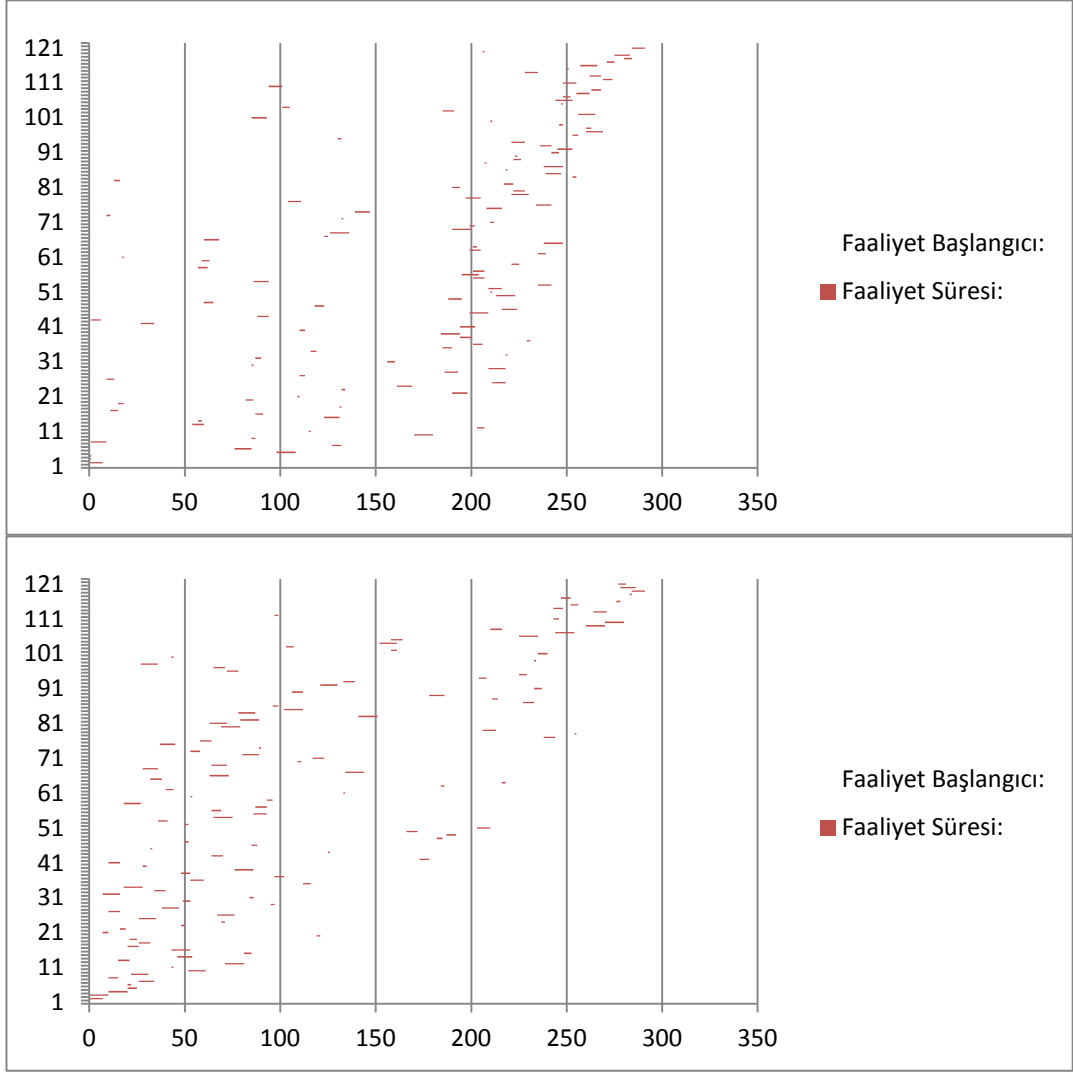
Şekil 6.16 Excel'de yerel en iyi değerleri

Şekil 6.17’de görüldüğü gibi çok projeli problemler ile çalışıldığında raporlama için ayrı bir sekme açılarak bu sekmede her projenin faaliyetlerinin başlangıç ve bitiş zamanları ayrı olarak gösterilir. Aynı zamanda şekil 6.18’de görüldüğü gibi her projenin kendi gant şeması çıkarılarak her projenin nasıl yönetileceği kolayca görülebilir.



Şekil 6.17 Excel’de çoklu proje sekmesi





Şekil 6.18 Excel’de çoklu proje gant şemaları

6.6 Algoritma Performansının Değerlendirilmesi

6.6.1 Algoritma performansını değerlendirme yolları

Oluşturulan algoritmaların yeterliliklerini ve diğer algoritmalarından üstünlüklerini değerlendirmek yapılan çalışmanın değerini belirlemek için çok önemlidir. Algoritmaların karşılaştırılabilmesi için problem kütüphaneleri kurulmuştur. Bu kütüphanelerdeki problemlerin sonuçları önceden bulunmuştur böylece algoritmaların oluşturduğu sonuçlarla kütüphanedeki sonuçların kıyaslanması yapılabilir. İstatistiksel analizlerle bulunan sonuçların asıl sonuçtan ne kadar saptığı bulunarak algoritma derecelendirilir.

Ayrıca ortak problem kütüphaneleri sayesinde bu kütüphaneler ile algoritmalarını değerlendiren diğer araştırmacıların çalışmaları ile karşılaştırma yapılabilir ve hangi algoritmanın daha kısa sürede daha iyi sonuçlar üretebildiği değerlendirilebilir.

Algoritmanın değerlendirilmesinde gerekli ayarlar kesin olarak belirtilmemiştir. Her algoritmanın en iyi sonuçları vermesini sağlayan ayarları denenerek belirlenmelidir. Algoritmaların çözüm üretme sürelerini kıyaslamada kullanmak çok doğru değildir, çünkü hızlı gelişen teknoloji ile birlikte bilgisayar hızları artmıştır. Bu nedenle eski çalışmaların hızları algoritmaları daha hızlı olsa bile günümüzdeki çalışmaların altında kalmışlardır. Çözüm üretme sürelerini kıyaslaya bilmek için günümüze çok yakın çalışmalarla veya yakın işlemci hızlarına sahip bilgisayarlarla yapılan çalışmalar kullanılabilir.

Bazı problemlerde en iyi çözümü bulan bir algoritma, bazı problemlerde en iyi çözümü bulamayabilir. Bu nedenle algoritmanın performansının değerlendirilmesinde çok miktarda problem denenerek algoritmanın ortalama performansının bulunması sağlanmalıdır.

Kütüphanelerdeki sınaama problemleri belirli parametreler esas alınarak üretilir. Büyüklük parametresi projedeki faaliyet sayısıdır. Proje karmaşıklığı (network complexity), (ayrıt sayısı/düğüm sayısı) olarak tanımlanmaktadır. Kaynak faktörü (resource factor), faaliyet başına düşen kaynak turunu ifade eder. Her faaliyetin kullandığı kaynak adedinin toplanarak faaliyet adedine bölünmesi ile bulunur. Kaynak gücü (resource strength), bir kaynağın üst kullanım sınırının bu kaynağın faaliyet başına ortalama kullanım değerine bölünmesi ile elde edilir[1].

6.6.2 Kaynak kısıtlı proje çizelgeleme problemleri kütüphanesi (PSPLİB)

Kolisch ve Sprecher'in [5] kurduğu PSPLİB (<http://129.187.106.231/psplib/>); tek modlu veya çok modlu kaynak kısıtlı proje çizelgeleme problemlerini çözen algoritmaların sonuçlarının kıyaslanani bileceği bir örnek problem kütüphanesidir. Bu kütüphanedeki örnekler Kolisch vd. [4] tarafından geliştirilen standart proje yaratıcısı (ProGen) ile yaratılmıştır. Kütüphanede problemlerin, optimumu bulunabilen problemler için optimum sonuçları, optimumu bulunamayan problemler için ise şimdiye kadar bulunan

en iyi sezgisel sonuçları bulunmaktadır. Kütüphane herkesin kullanımına açık olduğu için algoritmasını test etmek isteyen herkes kütüphaneden kaynak kısıtlı proje çizelgeleme problemlerini ve problemlere ait sonuçları indirebilirler. Mevcut en iyi çözümden daha iyisini bulabilen kullanıcılar ise kendi çözümlerini kütüphaneye göndererek kütüphanedeki çözümleri güncelleyebilirler. PSPLIB, proje çizelgeleme alanında birçok araştırmacının kullandığı, algoritmaların geliştirilmesini sağlayan çok değerli bir kaynaktır[5].

Tek modlu kaynak kısıtlı proje çizelgeleme problemleri için kütüphanede iki faaliyet kukla faaliyet olmak üzere, 32 faaliyetli, 62 faaliyetli, 92 faaliyetli ve 122 faaliyetli olmak üzere çeşitli büyüklük ve zorluklarda farklı örnek problem grupları mevcuttur. Her bir örnek problem grubunda ise 480 adet örnek problem vardır. Her bir problemde 4 farklı kaynak türü kullanılmıştır. 32 faaliyetli modlu kaynak kısıtlı proje çizelgeleme problemlerinin optimal çözümleri deterministik yöntemler ile bulunmuş ve kütüphaneye yüklenmiştir. 62 faaliyetli ve daha büyük problemler için ise optimal çözümler deterministik yöntemler ile bulunamamaktadır. Bu nedenle bu büyük problemler için sezgisel algoritmalar ile şimdiye kadar bulunan en iyi çözümler kütüphaneye yüklenmiştir.

6.6.3 Algoritma performansının değerlendirilmesi

Bu çalışmada geliştirilen algoritmanın performansını değerlendirmek için PSPLIB ve MPSPLIB Kütüphanesi kullanılmıştır. Kütüphanede bulunan tek modlu problemlerden 32 faaliyetli 30j grubundan 480, 62 faaliyetli 60j grubundan 200, 122 faaliyetli 120j grubundan ise 480 örnek problem ile algoritma değerlendirilmiştir. Başka algoritmalar ile kıyaslayabilme adına jenerasyon sayısı 1000 seçilmiştir. N sayıda faaliyet için N ve 2N popülasyon büyüklükleri alınıp popülasyon büyüklüğünün çözüm sonucuna ve çözüm hızına etkisi ölçülmüştür.

Çok modlu proje kütüphanesi olan MPSPLIB, PSPLIB problemlerinin birleştirilmesi ile oluşturulmuştur. 90 faaliyetli ve 120 faaliyetli problemleri 2li, 5li, 10lu ve 20li olarak birleştirilmiş ve çok projeli problemler oluşturulmuştur.

Algoritmanın daha iyi sonuçlar üretebilmesi için çeşitli mutasyon değerleri denenmiştir. Bölüm 6.1.3.2 'de incelendiği gibi en iyi sonuçları veren mutasyon değeri olan %50 mutasyon şansı seçilmiştir.

6.6.3.1 Algoritmanın J30 grubu problemlerle test edilmesi

32 faaliyetli 30j grubu problemler deterministik metotlar ile çözüle bilmektedir. Bu problemlerin optimum sonuçları bulunmuştur. Bu problem grubunda çalışmanın en büyük avantajı optimum değerlerin bulunmuş olması sayesinde eğer algoritma bu optimumdan daha düşük bir sonuç üretiyorsa bulunan sonucun bir geliştirme değil aksine algoritmada hata olduğu anlamına gelmesidir. 32 faaliyetli problem sınıfından 480 adet örnek üzerinde 1000 jenerasyon değeri, 30 ve 60 olmak üzere iki farklı popülasyon değerleri ve genetik ayarlar olarak da %10 elit seçim oranı ve %50 mutasyon şansı kullanılmıştır.

Çizelge 6.3'te bulunan sonuçların optimumdan sapmaları ve diğer araştırmacıların yaptıkları çalışmalarda buldukları sonuçların optimumdan sapmaları verilmiştir.

Çizelge 6.3 Sezgisel yöntemlerin performanslarının karşılaştırılması (j30 kümesi)

Araştırmacılar	Yıl	Sezgisel Yöntemler	Çizelgeleme Yöntemi	Üst sınırdan Ortalama Sapma
				1000 nesilde
Geliştirilen algoritma (30 popülasyonlu)	2013	GA- Random Key	s	0.72
Geliştirilen algoritma (60 popülasyonlu)	2013	GA- Random Key	s	0.33
J.L. Ponz-Tienda ve arkadaşları[39]	2012	GA		18.45
S. Paksoy[16]	2007	Random Key ve öncelik kuralı	s	25.36
D. Debels ve arkadaşları[15]	2006	Dağıntık arama ve Genetik algoritma		28.355
P.Tormos ve A.Lova[14]	2003	Pişmanlığa dayalı örnekleme- çok geçişli	s/p	0.23
K.Bouleimen, H.Lecocq[24]	2003	SA – Activity list	s	0.38
P.Tormos ve A.Lova[56]	2001	Pişmanlığa dayalı örnekleme	s/p	0.25
S. Hartmann[13]	2000	GA	s/p	0.37
S. Hartmann[12]	1998	Öncelik Kuralı	s	1.38
S. Hartmann[12]	1998	GA- Random Key	s	1.03
A.Schirmer, Riesenber[10]	1998	Sampling-adaptive	s/p	0.71
S. Hartmann[12]	1998	GA- Activity list	s	0.54
R. Kolisch, A. Drexl[6]	1996	Sampling-adaptive	s/p	0.74
R. Kolisch[78][7]	1996	Sampling-WCS	p	1.40
R. Kolisch[78]	1996	Sampling- MSLK	p	1.40
R. Kolisch[78]	1996	Sampling- MSLK	s	0.83
V.Leon, B. Ramamoorthy[79]	1995	GA- Rastgele anahtar	Mod. p	2.08
R. Kolisch[4]	1995	GA- Random Key	P	1.77
R. Kolisch[4]	1995	GA- Random Key	s	1.44

6.6.3.2 Algoritmanın J60 grubu problemlerle test edilmesi

62 faaliyetli j60 grubu problemler deterministik metotlar ile çözülememekte çözülebilse bile çözüm süresi kabul edilebilir sınırların çok üstünde olmaktadır. Bu nedenle bu problemlerin optimum sonuçlar yerine şimdiye kadar bulunan en iyi sonuçlar toplanmıştır. Bu grupta optimum değerler bilinmediği için araştırmacılar için bulunan en iyi sonucu geliştirmek önemli bir başarı sayılmaktadır. Ayrıca bulunan sonuç kütüphanede yayınlanmakta ve başka çalışmalara referans olmaktadır. 62 faaliyetli problem sınıfından 200 adet örnek üzerinde 1000 jenerasyon değeri, 60 ve 120 olmak üzere iki farklı popülasyon değerleri ve genetik ayarlar olarak da %10 elit seçim oranı ve %50 mutasyon şansı kullanılmıştır.

Çizelge 6.4'te bulunan sonuçların optimumdan sapmaları ve diğer araştırmacıların yaptıkları çalışmalarda buldukları sonuçların optimumdan sapmaları verilmiştir.

Çizelge 6.4 Sezgisel yöntemlerin performanslarının karşılaştırılması (j60 kümesi)

Araştırmacılar	Yıl	Sezgisel Yöntemler	Çizelgeleme Yöntemi	Üst sınırdan Ortalama Sapma
				1000 nesilde
Geliştirilen algoritma (120 popülasyonlu)	2013	GA- Random Key	s	0.93
Geliştirilen algoritma (60 popülasyonlu)	2013	GA- Random Key	s	1.25
J.L. Ponz-Tienda ve arkadaşları[39]	2012	GA		23.10
S. Paksoy[16]	2007	Random Key ve öncelik kuralı	s	36.53
D. Debels ve arkadaşları[15]	2006	Dağınık arama ve Genetik algoritma		38.554
P.Tormos ve A.Lova[14]	2003	Pişmanlığa dayalı örnekleme- çok geçişli	s/p	0.50
K.Bouleimen, H.Lecocq[24]	2003	SA – Activity list	s	1.17
P.Tormos ve A.Lova[56]	2001	Pişmanlığa dayalı örnekleme	s/p	0.55
S. Hartmann[13]	2000	GA	s/p	0.71
S. Hartmann[12]	1998	Öncelik Kuralı	s	1.32
S. Hartmann[12]	1998	GA- Random Key	s	2.36
A.Schirmer, Riesenberger[10]	1998	Sampling-adaptive	s/p	1.26
S. Hartmann[12]	1998	GA- Activity list	s	0.99
R. Kolisch, A. Drexl[6]	1996	Sampling-adaptive	s/p	1.48
R. Kolisch[78][7]	1996	Sampling-WCS	p	1.76
R. Kolisch[78]	1996	Sampling- MSLK	p	1.76
R. Kolisch[78]	1996	Sampling- MSLK	s	1.76
V.Leon, B. Ramamoorthy[79]	1995	GA- Random Key	Mod. p	2.48
R. Kolisch[4]	1995	GA- Random Key	P	2.71
R. Kolisch[4]	1995	GA- Random Key	s	3.34

6.6.3.3 Algoritmanın J120 grubu problemlerle test edilmesi

122 faaliyetli j120 grubu problemler deterministik metotlar ile çözülememekte çözülebilsen bile çözüm süresi kabul edilebilir sınırların çok üstünde olmaktadır. Bu nedenle bu problemlerin optimum sonuçları yerine şimdiye kadar bulunan en iyi sonuçlar toplanmıştır. Bu grupta optimum değerler bilinmediği için araştırmacılar için bulunan en iyi sonucu geliştirmek önemli bir başarı sayılmaktadır. Ayrıca bulunan sonuç kütüphanede yayınlanmakta ve başka çalışmalara referans olmaktadır. 122 faaliyetli problem sınıfından 200 adet örnek üzerinde 1000 jenerasyon değeri, 120 ve 240 olmak üzere iki farklı popülasyon değerleri ve genetik ayarlar olarak da %10 elit seçim oranı ve %50 mutasyon şansı kullanılmıştır.

Çizelge 6.5'te bulunan sonuçların optimumdan sapmaları ve diğer araştırmacıların yaptıkları çalışmalarda buldukları sonuçların optimumdan sapmaları verilmiştir.

Çizelge 6.5 Sezgisel yöntemlerin performanslarının karşılaştırılması (j120 kümesi)

Araştırmacılar	Yıl	Sezgisel Yöntemler	Çizelgeleme Yöntemi	Üst sınırdan Ortalama Sapma
				1000 nesilde
Geliştirilen algoritma (120 popülasyonlu)	2013	GA- Random Key	s	5.28
Geliştirilen algoritma (240 popülasyonlu)	2013	GA- Random Key	s	4.74
J.L. Ponz-Tienda ve arkadaşları[39]	2012	GA		27.86
S. Paksoy[16]	2007	Random Key ve öncelik kuralı	s	47.65
D. Debels ve arkadaşları[15]	2006	Dağıntık arama ve Genetik algoritma		75.537
P.Tormos ve A.Lova[14]	2003	Pişmanlığa dayalı örnekleme- çok geçişli	s/p	0.23
K.Bouleimen, H.Lecocq[24]	2003	SA – Activity list	s	5.73
P.Tormos ve A.Lova[56]	2001	Pişmanlığa dayalı örnekleme	s/p	0.44
S. Hartmann[13]	2000	GA	s/p	1.29
S. Hartmann[12]	1998	Öncelik Kuralı	s	2.58
S. Hartmann[12]	1998	GA- Random Key	s	6.85
A.Schirmer, Riesenbergl[10]	1998	Sampling-adaptive	s/p	3.28
S. Hartmann[12]	1998	GA- Activity list	s	2.59
R. Kolisch, A. Drexl[6]	1996	Sampling-adaptive	s/p	3.95
R. Kolisch[78][7]	1996	Sampling-WCS	p	2.71
R. Kolisch[78]	1996	Sampling- MSLK	p	2.69
R. Kolisch[78]	1996	Sampling- MSLK	s	4.55
V.Leon, B. Ramamoorthy[79]	1995	GA - Problem space	Mod. p	5.09
R. Kolisch[4]	1995	GA- Random Key	P	6.21
R. Kolisch[4]	1995	GA- Random Key	s	9.39

6.6.3.4 Algoritmanın çok projeli problemler ile test edilmesi

Homberger'in [26] kurduğu çok projeli kaynak kısıtlı proje çizelgeleme problemleri kütüphanesi MPSPLIB(<http://www.mpsplib.com/>) örnekleri kullanılarak algoritma sınanmıştır. MPSPLIB kütüphanesindeki problemler Kolisch'in [4] çalışmasında kullandığı problemlerin (PSPLIB) birleştirilmesi ile oluşmuştur. 8 adet problem kümesi vardır bunlar MP90_2, MP120_2, MP90_5, MP120_5, MP90_10, MP120_10, MP90_20 ve MP120_20'dir. Çizelge 6.6'da bilgileri verilen bu problemlerin proje sayıları $n = 2, 5, 10$ veya 20 'dir. Her bir problem kümesinde 10 adet çok projeli örnek vardır. Bu örneklerde projelerin bazıları başlangıç anında mevcut değildir ve belirli bir süre sonunda

gelmektedir. Böylece daha dinamik bir yapı sağlanmaktadır. Problemlerde 4 adet global kaynak mevcuttur ve projeler bu kaynaklar için yarışmaktadırlar.

Çizelge 6.6 Çok projeli problem örnekleri

Problem	Kümedeki örnek sayısı	Örnekteki proje sayısı	Bir projedeki faaliyet sayısı	Global kaynak sayısı	Kaynak faktörü
MP90_2	10	2	90	4	2.27
MP120_2	10	2	120	4	1.36
MP90_5	10	5	90	4	4.99
MP120_5	10	5	120	4	3.80
MP90_10	10	10	90	4	3.85
MP120_10	10	10	120	4	2.61
MP90_20	10	20	90	4	2.70
MP120_20	10	20	120	4	3.65

Hazırlanan algoritma toplam proje süresini en aza indirmeyi hedeflemektedir. Kıyaslamanın daha doğru yapılabilmesi adına diğer çalışmalarda kullanılan toplam 100050 çizelge oluşturması şartına uyulmuştur ve her problem için 100000 adet çizelge oluşturulmuştur. Yeterli sayıda çizelgenin hazırlanması için her nesilde 100 adet olmak üzere toplam bin adet nesil ile toplam 100000 çizelge hazırlanmaktadır. Bölüm 5.1.3.2 'de incelendiği gibi en iyi sonuçları veren mutasyon değeri olan %50 mutasyon şansı seçilmiştir.

İki değere göre kıyaslama yapılmıştır bunlar ortalama proje gecikmesi ve ortalama proje tamamlanma zamanlarıdır. Ortalama proje tamamlanma zamanı (TMS) en geç biten faaliyetin bitiş tarihi ve ilk başlayan faaliyetin başlangıç tarihi arasındaki süredir.

$$TMS = \max_b \text{Faaliyet}_{b.\text{bitiş}} - \min_a \text{Faaliyet}_{a.\text{başlangıç}} \quad (6.4)$$

Ortalama proje gecikmesi (APD), ortalama proje tamamlanma zamanından o projenin kritik yol süresinin çıkarılması ile bulunmaktadır.

$$APD = \sum (M_i - CPL_i) / N. \quad (6.5)$$

Bu iki değer aracılığı ile hazırlanan algoritma 7 farklı çalışma ile kıyaslanmıştır. bunlar ; CMAS/ES: Homberger'ın[30] evrimsel aramaya dayanan bir koordinasyon mekanizması kullanan çoklu ajan sistemidir. 50 popülasyon 100 jenerasyon ve 1000 ajan ile toplam $50+100 \times 1000 = 100050$ çizelge ile çözüm aranmıştır.

CMAS/SA: Homberger'in[30] çalışmasında kullandığı fink[80]'ün somutlaştırılmış koordinasyon mekanizmasına dayanan bir çoklu ajan sistemidir. 10000 tur ver 90050 ajan ile toplam 100050 çizelgeyle çözüm aranmıştır.

MAS/CI: Homberger'in [26] çalışmasında kullanılan merkezileştirilmiş çözümler ile koordinasyona dayanan bir çoklu ajan sistemidir. 100050 çizelge ile çözüm aranmıştır.

RES: Homberger'in [26] evrimsel meta-sezgisel algoritmasıdır. CMAS/ES ve CMAS/SA yaklaşımlarını kıyaslayabilmek için geliştirilmiştir. Çok projeli problemi tek projeliye dönüştürerek çözmektedir.

SASP: kurtulus ve Davis'in [59] öncelik kuralına dayanan sezgisel algoritmasıdır. SASP proje gecikmesini en aza indiren en iyi öncelik kurallı yaklaşımlardan biridir.

DMAS/ABN: Adhau v.d.'nin [33] açık artırmaya dayalı müzakere kullanan bir dağıtılmış çoklu ajan sistemidir. Kaynak çakışmalarını önlemek ve problemin optimum sonucunu bulabilmek için en iyi metodu ihale yöntemi kullanarak arayama dayanır.

DMAS/RIA: Adhau v.d. 'nın [34] açık artırmaya dayalı müzakere kullanan bir çoklu ajan sistemidir. Farklı yerlerde bulunan projeler için kaynakların taşınması maliyeti ve süresini de içerecek şekilde geliştirilmiştir. Diğer algoritmalar ile kıyaslama yapabilmek adına kaynak taşıma süresi ve maliyeti ihmal edilerek hesaplamalarda yapılmıştır.

Çizelge 6.7'de yapılan kıyaslama gösterilmektedir. Bulunan en iyi çözümler altı çizili olarak gösterilmiştir. Önerilen genetik algoritma en kısa proje tamamlanma zamanlarını bulmuştur. Ortalama proje gecikmeleri için ise oldukça iyi çözümler bulmuştur. Hazırlanan algoritmanın hedefi toplam proje sürelerini en aza indirmek olduğu için kıyaslama sonucunda algoritmanın diğerlerinden daha iyi olduğu görülmektedir.

Çizelge 6.7 Sezgisel yöntemlerin performansının kıyaslanması (çok projeli)

Problem	GA/AL		DMAS/ABN		DMAS/RIA		MAS/CI	
	APD	TMS	APD	TMS	APD	TMS	APD	TMS
MP90_2	149,05	<u>229,7</u>	144.15	241.9	167,5	263,7	<u>127.75</u>	260
MP120_2	60,2	144,4	<u>47.0</u>	150.90	64,95	171,5	54.5	182.7
MP90_5	448,06	<u>536,2</u>	384.08	548.8	426,7	603,3	<u>287.76</u>	609.8
MP120_5	350,37	<u>441,6</u>	291.44	497.2	286,5	545,2	<u>247.76</u>	592.6
MP90_10	358,98	<u>455,1</u>	313.33	466.2	299,1	503	<u>244.92</u>	582.3
MP120_10	247,54	<u>350,2</u>	171.54	370.2	<u>167,4</u>	390,9	182.07	496.1
MP90_20	185,35	<u>281,6</u>	<u>146.37</u>	295.2	166,8	317,2	169.14	448
MP120_20	387,30	<u>497,1</u>	297.39	525.7	<u>263,6</u>	549	319.52	760.2
Problem	CMAS/ES		CMAS/SA		RES		SASP	
	APD	TMS	APD	TMS	APD	TMS	APD	TMS
MP90_2	146.7	240.2	189.8	242.11	149.22	232.22	158.6	260.6
MP120_2	51.6	140.1	55.55	145.9	49.5	<u>139.2</u>	56.7	167.8
MP90_5	426.8	538.4	455.37	548.6	421.18	538.2	404.6	574.8
MP120_5	352.7	480.7	368.18	486.1	353.24	481.8	258.8	547.87
MP90_10	336.2	468.6	341.8	460.7	366.42	458.3	283.9	493.9
MP120_10	232.2	350.7	253.2	354	299.9	352.7	181	394.1
MP90_20	171.66	286.4	174	288.1	170.63	285.9	161.8	316.6
MP120_20	369.7	508.5	368.5	508.6	363.65	506.4	297.4	552.4

Diğer araştırmalar ile daha kolay karşılaştırma yapabilmek adına algoritmanın hedefi ortalama proje gecikmesini en aza indirmek olarak değiştirilmiştir. 1000 nesil ve 100 popülasyon değeri ile toplam 100000 çizelge üreterek yapılan araştırma sonucunda ortaya çıkan sonuçlar çizelge 6.8'de gösterilmektedir. En iyi sonuçlar altı çizili olarak gösterilmiştir.

Algoritmada bu iki hedeften yalnızca biri seçilmelidir. Tek projeli problemler iki hedefte aynı amaca sahipken çok projeli problemlerde hedeflerden biri seçilerek bulunan sonuçlar birbirinden farklı olacaktır.

Algoritma hedefi ortalama proje gecikmesini en aza indirmek olduğu halde, toplam proje süresini diğer çalışmalara göre oldukça kısadır. Bu nedenle iki hedef fonksiyonu arasında bir oranlama yapılmaya gerek duyulmamıştır.

Çizelge 6.8 Sezgisel yöntemlerin performansının kıyaslanması (çok projeli)

Problem	GA/AL		DMAS/ABN		DMAS/RIA		MAS/CI	
	APD	TMS	APD	TMS	APD	TMS	APD	TMS
MP90_2	145,7	<u>231,5</u>	144.15	241.9	167,5	263,7	<u>127.75</u>	260
MP120_2	50,45	151,2	<u>47.0</u>	150.90	64,95	171,5	54.5	182.7
MP90_5	413,3	<u>515,7</u>	384.08	548.8	426,7	603,3	<u>287.76</u>	609.8
MP120_5	324,98	496,6	291.44	497.2	286,5	545,2	<u>247.76</u>	592.6
MP90_10	341,85	460,9	313.33	466.2	299,1	503	<u>244.92</u>	582.3
MP120_10	226,75	360,1	171.54	370.2	<u>167,4</u>	390,9	182.07	496.1
MP90_20	158,71	294,7	<u>146.37</u>	295.2	166,8	317,2	169.14	448
MP120_20	330,01	523,3	297.39	525.7	<u>263,6</u>	549	319.52	760.2
Problem	CMAS/ES		CMAS/SA		RES		SASP	
	APD	TMS	APD	TMS	APD	TMS	APD	TMS
MP90_2	146.7	240.2	189.8	242.11	149.22	232.22	158.6	260.6
MP120_2	51.6	140.1	55.55	145.9	49.5	<u>139.2</u>	56.7	167.8
MP90_5	426.8	538.4	455.37	548.6	421.18	538.2	404.6	574.8
MP120_5	352.7	<u>480.7</u>	368.18	486.1	353.24	481.8	258.8	547.87
MP90_10	336.2	468.6	341.8	460.7	366.42	<u>458.3</u>	283.9	493.9
MP120_10	232.2	<u>350.7</u>	253.2	354	299.9	352.7	181	394.1
MP90_20	171.66	286.4	174	288.1	170.63	<u>285.9</u>	161.8	316.6
MP120_20	369.7	<u>508.5</u>	368.5	508.6	363.65	506.4	297.4	552.4

SONUÇ VE ÖNERİLER

Bu çalışmada polinomsal zorlukta, tek ve çok projeli, kaynak kısıtlı proje çizelgeleme problemleri için bir genetik algoritma önerilmiştir. Çalışmanın amacı Proje çizelgeleme problemleri için mevcut kaynak kapasitelerini aşmadan en kısa tamamlanma sürelerini bulmaktır. Algoritma C# programlama dilinde, .net 4.5 kod kütüphaneleri kullanılarak ve sınıf yapılarından faydalanılarak hazırlanmıştır. Önerilen algorithma seri çizelgeleme ve faaliyet listesi tasarımı kullanılmıştır. Yerel optimumlardan kaçınmak ve global optimum sonucu bulabilmek için rastsallığı önleyecek yöntemlerden kaçınılmıştır. Seri çizelgeleme yapısında faaliyetler tamamen eşit olasılıkla seçilmiş, çaprazlama ve mutasyon için tamamen rastsal olarak seçim yapılmıştır.

Önerilen algoritmanın iki hedefi vardır. Bu hedefler kaynak kapasitelerini aşmadan en kısa tamamlanma zamanını bulmak veya ortalama proje süresini en aza indirmek hedeflerinden biri olarak seçilebilir. Çok projeli problemlerde iki hedeften birinin seçilmesi sonuçları etkilemekteyken. Tek projeli problemlerde bu iki hedeften birinin seçilmesi sonucu değiştirmeyecektir.

Bir genetik algoritmanın en iyi sonucu vermesi için en doğru arama ayarlarına sahip olması gerekir. Bu nedenle algoritmanın farklı parametre ve ayarları denenmiş, kaynak kısıtlı proje çizelgeleme problemleri için en uygun çaprazlama ve mutasyon değerleri bulunmuştur.

Bulunan sonuç bilgileri Excel dosyasına yazdırıldığı için sonuçlar kolaylıkla saklanabilir ve daha ayrıntılı gant diyagramları ve grafikler hazırlanabilir.

Algoritmanın performansının literatürdeki diğer çalışmalar ile karşılaştırılabilmesi için PSPLIB ve MPSPLIB kütüphaneleri kullanılmıştır. Bu kütüphaneleri kullanarak değerlendirme yapan diğer araştırmacıların çalışmaları ile karşılaştırma yapılmış ve algoritma değerlendirilmiştir. Yapılan kıyaslama sonuçlarına göre, tek projeli problemlerde, 30j kümesinde 6., 60j kümesinde 4., 120j kümesinde 13. sıradadır. Çok projeli problemlerde ise projelerin nihai tamamlanma süresini en aza indirmede çok etkili olduğu yapılan kıyaslama sonucunda görülmüştür. Algoritmanın hedefinin toplam proje tamamlanma süresini en aza indirmek olarak seçildiğinde proje gecikmelerini azaltmada genellikle 5. veya 6. sırada olduğu görülmüştür. Algoritma hedefi ortalama proje gecikmelerini en aza indirmek olduğu zaman ise genellikle 3. veya 4. sırada yer aldığı görülmüştür.

Bundan sonraki proje çizelgeleme çalışmaları için, ilk popülasyonun oluşturulması sırasında birbirinin aynı bireylerin oluşmasının önüne geçilmesi sağlanabilir. Bir maliyet kriteri eklenerek türü yenilenebilir olmayan kaynaklar içinde sınırlama eklenebilir.

KAYNAKLAR

-
- [1] Ulusoy, G., (2000), "Proje Planlamada Kaynak Kısıtlı Çizelgeleme", Mühendislik ve Doğa Bilimleri Fakültesi, Sabancı Üniversitesi, Orhanlı, Tuzla, 81474 İstanbul
 - [2] Holland, J.H., (1976), Adaptation in Nature and Artificial Systems, University of Michigan Press, Ann Arbor.
 - [3] Şimşir, F., (2008), Kaynak Kısıtlı Bakım Çizelgeleme Problemine Bir Hibrid Çözüm Yaklaşımı, Sakarya Üniversitesi, Fen Bilimleri Enstitüsü Doktora Tezi, Sakarya.
 - [4] Kolisch, R. (1995), Project Scheduling under Resource Constraints, Physica-Verlag, Heidelberg.
 - [5] Kolisch, R. ve Sprecher A., (1996), "PSPLIB - A project scheduling problem library", ELSEVIER European Journal of Operational Research, 96:205-2016
 - [6] Kolisch, R. ve Drexel, A., (1996), "Adaptive search for solving hard project scheduling problems" Naval Research Logistics. 43,1:23-40
 - [7] Kolisch, R., (1996b), "Efficient priority rules for the resource-constrained project scheduling problem", Journal of Operations Management, 14:179:192.
 - [8] Brucker, P., v.d., (1998), "A branch and bound algorithm for the resource-constrained project scheduling problem", ELSEVIER European Journal of Operational Research, 107:272-288
 - [9] Tsai, Y. ve Gemmill, D.,D., (1998), "Using tabu search to schedule activities of stochastic resource-constrained projects", ELSEVIER European Journal of Operational Research 111:129-141
 - [10] Schirmer, A., (1998), "Case-based reasoning and improved adaptive search for project scheduling", Naval Research Logistics. 47:201-222.
 - [11] Reyck, B. ve Herroelen, W., (1998), "a branch-and-bound procedure for the resource-constrained project scheduling problem with generalized precedence relations", ELSEVIER European Journal of Operational Research, 111:152-174
 - [12] Hartmann, S., (1998), "A competitive genetic algorithm for resource-constrained scheduling" Naval Research Logistics 45,7:733-750

- [13] Hartmann, S., (2000), "A self-Adapting Genetic Algorithm for Project Scheduling Under Resource Constraints" *Naval Research Logistics* 49,5:433-448.
- [14] Tormos, P. ve Lova, A., (2003), "An efficient multi-pass heuristic for project scheduling with constrained resources" *International Journal of Production Research*,41,5:1071-1086
- [15] Debels, D., vd., (2006), " A hybrid scatter search/electromagnetism meta-heuristic for Project scheduling", *ELSEVIER European Journal of Operational Research*, 169:638-653.
- [16] Paksoy, S., (2007), Genetik algoritma ile proje çizelgeleme, Çukurova Üniversitesi Sosyal Bilimler Enstitüsü İşletme Anabilim Dalı Doktora Tezi, Adana.
- [17] Sengupta, A., Sedaghat, R. ve Salkar, P., (2012). "A multi structure genetic algorithm for integrated desing space exploration of scheduling and allocatin in high level synthesis for DSP kernels" *ELSEVIER Swarm and Evolutionary Computation*, 7: 35-46.
- [18] Boctor, F., F., (1996), "A new and efficient heuristic for scheduling projects with resource restrictins and multiple execution modes", *ELSEVIER European Journal of Operational Research*, 90:349-361.
- [19] Mori, M. ve Tseng C. C., (1997). "A genetic algorithm for multi-mode resource constrained Project scheduling problem" *ELSEVIER European Journal of Operational Resarch*, 100:134-141.
- [20] Sprecher, A. ve Drexl, A., (1998), "Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm", *ELSEVIER European Journal of Operational Research* 107:431-450
- [21] Ahn, T. ve Erengüc S., S., (1998), "The resource constrained project scheduling problem with multible crashable modes: A heuristic procedure", *ELSEVIER European Journal of Operational Research* 107:250-259,
- [22] Afshar-Nadjafi, B., Rahimi, A. ve Karimi, H., (2001). "A genetic algorithm for mode identity and the resource constrained Project scheduling problem" *Scientia Iranica, Makale Basımda*.
- [23] Ranjbar, M.R. ve Kianfar F., (2007), " Solving the time/resource trade-off problem in Project scheduling with genetic algorithms", *ELSEVIER Applied Mathematics and Comutation*, 191:451-456.
- [24] Bouleimen, K. ve Lecocq H., (2003), "A new efficient simulated annealing algorithm fort he resource-constrained Project scheduling problem and its multiple mode version" *ELSEVIER European Journal of Operational Research* 149:268-281
- [25] Shi-man, X., Jian-wei, C. ve Zhao-yan, X., (2012). "The study of Genetic Algorithm by Hierarchical Coded for the MMRCPSP" *ELSEVIER Physics Procedia*, 24: 2200-2206.

- [26] Homberger, J., (2007). "A multi-agent system for the decentralized resource-constrained multi-project scheduling problem", *International Transactions in Operational Research*, 14(6), 565-589.
- [27] Gonçalves, J., F., v.d. (2008), "A genetic algorithm for the resource constrained multi-project scheduling problem", *ELSEVIER European Journal of Operational Research*, 189:1171-1190.
- [28] Le, H., Q., (2008), "Resource-Constrained Multi-Project Scheduling with Resource Moving Time for Construction Projects in Vietnam", *First International Conference on Construction In Developing Countries (ICCIDC-I), "Advancing and Integrating Construction Education, Research & Practice"* August 4-5, Karachi, Pakistan.
- [29] Pamay, M.,B., Bülbül, K. ve Ulusoy, G.,(2012), " dynamic resource constrained multi-project scheduling problem with weighted earliness/tardiness costs" Sabancı University, Orhanlı, Tuzla 34956. Istanbul, Turkey.
- [30] Homberger, J., (2009), " A(m, l)-coordination mechanism for agent-based multi-project scheduling". *ORSpectrum* 34:107-132.
- [31] Browning, T.,R. ve Yassine, A.,A.,(2010), "Resource-constrained multi-project scheduling: Priority rule performance revised", *ELSEVIER Int. J. Production Economics*, 126:212-228.
- [32] Bakshi, T., Sarkar, B. ve Sanyal S.K., (2012), "An Evolutionary Algorithm for Multi-criteria Resource Constrained Project Scheduling Problem Based On PSO", *ELSEVIER Procedia Technology*, 6:231-238.
- [33] Adhau, S., v.d. (2012) "A multi-agent system for distributed multi-project scheduling: An auction-based negotiation approach", *ELSEVIER Engineering Applications of Artificial Intelligence*, 25:1738-1751.
- [34] Adhau, S., v.d. (2013) "A multi-agent system for decentralized multi-project scheduling with resource transfer", *ELSEVIER Int. J. Production Economics*, 146:646-661.
- [35] Beşikçi, U., v.d., (2012), "Multi-Mode Resource Constrained Multi-Project Scheduling and Resource Portfolio Problem", *Department of Industrial Engineering, Boğaziçi University, Istanbul*.
- [36] Can, A. ve Ulusoy, G.,(2012), "Multi-Project Scheduling With. 2-Stage Decomposition". Sabancı University, Orhanlı, Tuzla 34956. Istanbul.
- [37] Shadrokh, S. ve Kianfar, F., (2007), "A genetic algorithm for resource investment Project scheduling problem, tardiness permitted with penalty", *ELSEVIER European Journal of Operational Research*, 181:86-101.
- [38] Erdal, M., (2007), *Kısıtlı Kaynak Koşullarında Yapı Projelerinin Genetik Algoritma ile Programlanması*, 2007, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı Yüksek Lisans Tezi, Ankara.

- [39] Ponz-Tienda, J.L., vd., (2012), "The Resource Leveling Problem with multiple resource using an adaptive genetic algorithm", *ELSEVIER Automation in Construction*, 29:161-172.
- [40] Li, Y., Ip, W.H. ve Wang, D.W., (1998), "Genetic algorithm approach to earliness and tardiness production scheduling and planning problem", *ELSEVIER Int. J. Production Economics*, 54:65-76.
- [41] ElMaraghy, H., Patel, V. ve Abdallah, I., B., (1999), "A Genetic Algorithm Based Approach for Scheduling of Dual-Resource Constrained Manufacturing Systems" *Annals of the CIRP*, 48/1/1999.
- [42] Delavar, M., R., Hajiaghaei-Keshteli, M., ve Molla-Alizadeh-Zavardehi, S., (2010), "Genetic algorithms for coordinated scheduling of production and air transportation " *ELSEVIER Expert Systems with Applications*, 37:8255-8266.
- [43] Orero, S.O. ve Irving, M.R., (1996), " A genetic algortihm for generator scheduling in power systems" *ELSEVIER Electrical Power and Energy Systems*, 18:19-26.
- [44] Wong, S.Y.W., (2001),"Hybrid simulated annealing/genetic algorithm approach to short-term hydro-thermal scheduling with multiple thermal plants", *ELSEVIER Electrical Power and Energy Systems*, 23:565-575.
- [45] Brucker, P., (2002), "Scheduling and constraint propagation" *ELSEVIER Discrete Applied Mathematics*, 123:227-256.
- [46] Carter, A. E. ve Ragsdale, C.T., (2002), " Scheduling pre-printed newspaper advertising inserts using genetic algorithms", *PERGAAMON Omega* 30:415-421.
- [47] Gupta, P., Bazargan, M., ve McGrath, R., N., (2003), *Simulation Model for Aircraft Line Maintenance Planning*, 2003 Proceedings Annual Reliability and Maintainabilty Symposium.
- [48] Juang, Y, vd., (2007), "An adaptive scheduling system with genetic algorithms for arranging employee training programs" *ELSEVIER Expert Systems with Applications*, 33:642-651.
- [49] Chang C. K., vd., (2008), "Time-line based model for software Project scheduling with genetic algorithms" *ELSEVIER Information and Software Technology*, 50:1142-1154.
- [50] Papakostas, N., vd., (2010), "An Approach to operational aircraft maintenance planning" *ELSEVIER Decision Support Systems*, 48:604-612.
- [51] Beliën, J. ve arkadaşları, (2012), "Integrated staffing and scheduling for an aircraft line maintenance problem" *ELSEVIER Computers & Operations Research*, 40:1023-1033.
- [52] Remenyi, C., ve Staudacher, S., (2012), "Systematic simulation based approach for identification and implementation of scheduling rule in the aircraft engine maintenance" *ELSEVIER Int. J. Production Economics*,

- [53] Herroelen, W., vd., (1998), "A classification scheme for Project scheduling in: Project Scheduling: Recent Models Algorithms and Applications", Kluwer Academic, Amsterdam, 1-26.
- [54] Çubukçu, R., (2008), Proje Yönetiminde Zaman ve Maliyet Risklerinin Çizelgeleme Yöntemiyle Minimize Edilmesi, Çukurova Üniversitesi Fen Bilimleri Enstitüsü, Endüstri Mühendisliği Anabilim Dalı Doktora Tezi, Adana.
- [55] Balkaya, A., H., (2011), Kaynak Kısıtlı Proje Çizelgeleme Problemlerinin Genetik Algoritma Yaklaşımı İle Optimizasyonu Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Ekonometri Anabilim dalı Ekonometri Programı Yüksek Lisans Tezi, İzmir.
- [56] Lova, A., ve Tormos, P., (2001), "Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling.", *Annals of Operations Research* 201:263-286.
- [57] Hartmann, S. ve Kolisch R., (1999), "Heuristic Algorithms For Solving The Resource-Constrained Project Scheduling Problem; Classification and Computational Analysis", *International Series in Operations Research & Management Science*, 14:147-178.
- [58] Payne, J. H. (1995). Management of multiple simultaneous projects: A state-of-the-art review. *International Journal of Project Management*, 13(3), 163.
- [59] Kurtulus, I., ve Davis, E. W. (1982). "Multi-Project Scheduling: Categorization Of Heuristic Rules Performance".*Management Science* (Pre-1986), 28(2), 161.
- [60] Sprecher, A., (1993), "Resource-constrained project scheduling: Exact methods for the multi-mode case", Springer Verlag, Berlin 1-17.
- [61] Patterson, J. H., (1984), "A comparison of exact approaches for solving the multiple constrained resource project scheduling problems", *Management Science*, 30 (7): 854-867.
- [62] Thesen, A., (1976), "Heuristic scheduling of activities under resource and precedence restrictions", *Management Science*, 23 (4): 412-422.
- [63] Blazewicz, J., v.d., (2007) "Handbook on Scheduling, From Theory to Applications" bölüm 2 Basics, 35-37.
- [64] Özdemir, Y., S., (2008), Karınca Kolonisi Algoritması ile Bilgisayar Ağlarının Topolojik En İyilenmesi, Başkent Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı Yüksek Lisans Tezi, Ankara.
- [65] Kirkpatrick, S., vd., (1983), "Optimization by simulated annealing", *Science*, 220:671-680.
- [66] Şahin, R., (2008), "Dinamik Tesis Düzenleme Problemi İçin Tavlama Benzetimi Sezgiseli", *Gazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi* No:4, Ankara
- [67] Glover, F., (1989), Tabu search, *ORSA Journal on Computing*, 1:190-206 ve 2:4-32.

- [68] Yalçınöz, T., vd. (2001), Tabu Araştırması Uygulanarak Ekonomik Yük Dağıtım Probleminin Çözümü, Niğde Üniversitesi Mühendislik Mikarlık Fakültesi, Elektrik-Elektronik Mühendisliği Bölümü FBE 2001/04 No'lu proje, Niğde.
- [69] Moncayo-Martinez, L., A., ve Zhang, D., Z., (2011), "Mult-objective ant colony optimisation: A meta-heuristic approach to supply chain design", ELSEVIER Int. J. Production Economics, 131:407-420.
- [70] Oktay, S., ve Engin, O., (2006), "Scatter Search Method For Solving Industrial Problems", Selçuk Üniversitesi Mühendislik Mimarlık Dergisi. Konya.
- [71] Goldberg, D.E., (1989), "Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Reading, MA.
- [72] Emel, G., G. ve Taşkın, Ç., (2002), "Genetik algoritmalar ve uygulama alanları", Uludağ Üniversitesi, İktisadi ve İdari Bilimler Fakültesi Dergisi, 21:129-152.
- [73] Bolat, Bb, vd. (2004), "Mühendislik uygulamalarında genetik algoritma ve operatörlerin işlevleri", Mühendislik ve Fen Bilimleri Dergisi, 4:264-271.
- [74] Şen, Z., (2004), Genetik Algoritmalar ve En İyileme Yöntemleri, Su Vakfı, sf 87,İstanbul.
- [75] Braz, L., (2006), "Critical Path Method Implementation in C#", <http://www.codeproject.com/Articles/25312/Critical-Path-Method-Implementation-in-C>, 03 Nisan 2014.
- [76] Gotshall, S. ve Rylander, B., (2000), "Optimal Population Size and Genetik Algorithm", University of Portland press, Portland.
- [77] DAVIS, L., (1985), "Applying adaptive algorithms to epistatic domains" In: Proceedings of the International Joint Conference on Artificial Intelligence,
- [78] Kolisch, R., (1996a), "Serial and parallel resource-constrained project scheduling methods revisited: theory and computation. European Journal of Operational Research. 90:320-333.
- [79] Leon, V. ve Ramamoorthy, B., (1995), "Strength and adaptability of problem-space based neighborhoods for resource-constrained scheduling. O R Spektrum 17:173-182.
- [80] Fink, A., (2004), "Supply chain coordination by means of automated negotiations". In: Proceedings of the 37th Hawaii international conference on system sciences. IEEE Computer Society, Washington DC, USA, CD-ROM, p 10.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı :Uğur SATIÇ
Doğum Tarihi ve Yeri :15.06.1989 İstanbul
Yabancı Dili : İngilizce
E-posta : ugursatic@gmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	Endüstri Mühendisliği	Yıldız Teknik Üniversitesi	2011
Lise	Anadolu Lisesi	Pertevniyal Anadolu Lisesi	2007