



T.C.
İSTANBUL ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ



YÜKSEK LİSANS TEZİ

İNŞAAT PROJELERİNDE ZAMAN MALİYET ÖDÜNLEŞİM
PROBLEMİNİN EVRİMSEL TABANLI METASEZGİSEL
ALGORİTMALARLA OPTİMİZASYONU

Çağrı UYSAL

İnşaat Mühendisliği Anabilim Dalı

İnşaat Mühendisliği Programı


DANIŞMAN
Prof. Dr. Ekrem MANİSALI

Ağustos, 2018

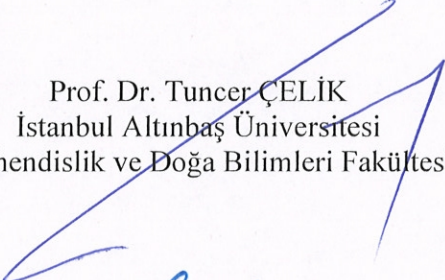
İSTANBUL

Bu çalışma, 7.08.2018 tarihinde aşağıdaki jüri tarafından İnşaat Mühendisliği Anabilim Dalı, İnşaat Mühendisliği Programında Yüksek Lisans tezi olarak kabul edilmiştir.

Tez Jürisi



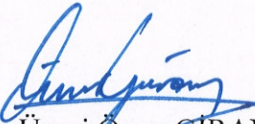
Prof. Dr. Ekrem MANISALI(Danışman)
İstanbul Üniversitesi
Mühendislik Fakültesi



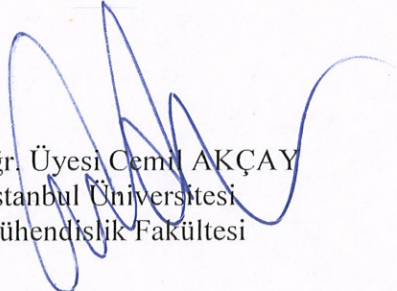
Prof. Dr. Tuncer ÇELİK
İstanbul Altınbaş Üniversitesi
Mühendislik ve Doğa Bilimleri Fakültesi



Doç. Dr. Yusuf Sait TÜRKKAN
İstanbul Üniversitesi
Mühendislik Fakültesi



Dr. Öğr. Üyesi Ömer GIRAN
İstanbul Üniversitesi
Mühendislik Fakültesi



Dr. Öğr. Üyesi Cemil AKÇAY
İstanbul Üniversitesi
Mühendislik Fakültesi



20.04.2016 tarihli Resmi Gazete’de yayımlanan Lisansüstü Eğitim ve Öğretim Yönetmeliğinin 9/2 ve 22/2 maddeleri gereğince; Bu Lisansüstü teze, İstanbul Üniversitesi’nin aboneli olduğu intihal yazılım programı kullanılarak Fen Bilimleri Enstitüsü’nün belirlemiş olduğu ölçütlere uygun rapor alınmıştır.

ÖNSÖZ

Çalışmamın ortaya çıkmasında yol göstericiliği ve bilgisiyle destek olan saygıdeğer danışman hocam Prof. Dr. Ekrem MANİSALI'ya; tez yazım sürecinde gerek psikolojik gerekse içerik anlamında yol göstericiliğini eksik etmeyen değerli hocam Dr. Öğr. Üyesi Ömer GİRAN'a; çalışmamın belirlenmesi ve teslim sürecinde yardımlarını esirgemeyen Arş. Gör. Dr. Osman Hürol TÜRKAKIN'a; yazılım ve kodlama bilgisini her daim paylaşan değerli meslektaşım Hazal NOYAN AKARSU'ya; tez yazım sürecini denetleyerek nihayete ermesini sağlayan değerli iş arkadaşım Rahime YILMAZ'a; Lisans ve Yüksek Lisans öğrenimimi beraber sürdürdüğümüz, değerli arkadaşım ve meslektaşım Mohammad Lemar ZALMAI'ye; yardım ve destekleriyle daima yanımda olan ikinci ailem Fatma ve Nazif NUR çiftine, tez yazım sürecinde varlıklarıyla bana moral ve güç veren, maddi manevi desteklerini benden esirgemeyen annem Müzeyyen UYSAL ve babam Metin UYSAL'a, beraber birçok çalışmaya imza atacağımız müstakbel meslektaşım, kardeşim R. Gizem UYSAL'a ve yazdığım her kelimedede bir şekilde payı olan; sabrına, anlayış ve hoşgörüsüne sığındığım sevgili eşim Emel NUR UYSAL'a teşekkürlerimi sunarım.

Ağustos 2018

Çağrı UYSAL

İÇİNDEKİLER

Sayfa No

ÖNSÖZ	iv
İÇİNDEKİLER.....	v
ŞEKİL LİSTESİ	viii
TABLO LİSTESİ.....	xi
SİMGE VE KISALTMA LİSTESİ	xiv
ÖZET	xvi
SUMMARY	xviii
1. GİRİŞ	1
2. GENEL KISIMLAR	3
2.1. PROJE YÖNETİMİ KAVRAMI	3
2.2. PROJE PLANLAMA VE ÇİZELGELEME	7
2.3. ŞEBEKE ÇİZELGELEME TEKNİKLERİ.....	15
2.3.1. Faaliyetlerin Düğümler Üzerinde Gösterimi	17
2.3.2. Faaliyetlerin Oklar Üzerinde Gösterimi	23
2.3.3. Program Değerlendirme ve Gözden Geçirme Tekniği (PERT).....	28
2.3.4. Kritik Yol Metodu (CPM)	31
2.3.5. Matris Yöntem	38
2.4. ZAMAN MALİYET ÖDÜNLEŞİM PROBLEMİ.....	42
2.5. OPTİMİZASYON	44
2.5.1. Çok Amaçlı Optimizasyon	45
2.5.2. Pareto Cephesi	48
2.5.3. Deterministik ve Stokastik Yaklaşım	50
2.6. SEZGİSEL VE METASEZGİSEL METOTLAR.....	51
2.6.1. Metasezgisel Algoritmalar.....	54
2.6.2. Evrimsel Tabanlı Optimizasyon ve Evrimsel Algoritmalar	54
2.6.2.1. Evrimsel Strateji.....	55
2.6.2.2. Evrimsel Programlama.....	56
2.6.2.3. Genetik Programlama.....	57
2.7. GENETİK ALGORİTMALAR.....	58
2.7.1. Gen, Kromozom ve Kodlama Yapısı	64

2.7.2. Popülasyon.....	64
2.7.3. Uygunluk Değeri (fi) ve Uygunluk Fonksiyonu.....	64
2.7.4. Seçim Operatörü	65
2.7.4.1. <i>Rulet Çarkı (Tekerleği) Yöntemi</i>	65
2.7.4.2. <i>Elitizm</i>	66
2.7.5. Çaprazlama Operatörü.....	66
2.7.5.1. <i>Tek Noktalı Çaprazlama</i>	67
2.7.5.2. <i>İki Noktalı – Çok Noktalı Çaprazlama</i>	67
2.7.5.3. <i>Uniform Çaprazlama</i>	68
2.7.6. Mutasyon Operatörü	69
2.7.7. Durdurma Kriteri	69
3. MALZEME VE YÖNTEM.....	70
3.1. PROBLEM TANIMI.....	70
3.2. PROBLEM ÇÖZÜMÜ.....	73
3.2.1. Başlangıç Popülasyonunun Oluşturulması	74
3.2.2. CPM Hesaplamaları.....	77
3.2.3. Seçim Operatörü	79
3.2.4. Çaprazlama Operatörü.....	82
3.2.5. Mutasyon Operatörü	84
3.3. MATLAB PROGRAM KODU	87
4. BULGULAR.....	92
4.1. POPÜLASYON BÜYÜKLÜĞÜNÜN PARETO ÇÖZÜMÜNE ETKİSİ.....	92
4.1.1. 50 Popülasyon Büyüklüğü ile Çözüm	92
4.1.2. 100 Popülasyon Büyüklüğü ile Çözüm	94
4.1.3. 500 Popülasyon Büyüklüğü ile Çözüm	95
4.1.4. 1000 Popülasyon Büyüklüğü ile Çözüm	97
4.1.5. 5000 Popülasyon Büyüklüğü ile Çözüm	98
4.1.6. Popülasyon Büyüklüğüne Göre Çözümlerin Karşılaştırılması.....	100
4.2. İTERASYON SAYISININ PARETO ÇÖZÜMÜNE ETKİSİ	102
4.2.1. 50 İterasyon ile Çözüm.....	103
4.2.2. 100 İterasyon ile Çözüm.....	104
4.2.3. 500 İterasyon ile Çözüm.....	105
4.2.4. 1000 İterasyon ile Çözüm.....	107
4.2.5. 5000 İterasyon ile Çözüm.....	109

4.2.6. İterasyon Sayısına Göre Çözümlerin Karşılaştırılması	111
4.3. MGA VE GA YAKLAŞIMLARININ KARŞILAŞTIRILMASI	113
4.4. MGA SONUÇLARININ CPM HESAPLAMALARI	117
5. TARTIŞMA VE SONUÇ	122
KAYNAKLAR.....	124
ÖZGEÇMİŞ	129



ŞEKİL LİSTESİ

	Sayfa No
Şekil 2.1: Proje yaşam döngüsü (PMI, 2008).	7
Şekil 2.2: Prefabrik bir yapıya ait iş alt kırılım yapısı gösterimi.	10
Şekil 2.3: İnşaat projeleri için mevcut planlama teknikleri (Aziz, 2013).	13
Şekil 2.4: Temel inşaatı işlerinin Gantt Şeması ile gösterimi.	14
Şekil 2.5: Aktivite düğümüne ait standart veriler.	18
Şekil 2.6: Faaliyet ilişkilerinde hatalı (a) ve doğru (b) gösterimler.	18
Şekil 2.7: Düğümden çıkan okların hatalı (a) ve doğru (b) gösterimi.	19
Şekil 2.8: Kesişen okların hatalı (a) ve doğru (b) gösterimleri.	19
Şekil 2.9: Projenin başlangıç ve bitişinin hatalı (a) ve doğru (b) gösterimleri.	19
Şekil 2.10: Aktivitelerin düğümler üzerinde gösterimi.	20
Şekil 2.11: Bitiş – Başlangıç tipi öncelik ilişkisi.	22
Şekil 2.12: Bitiş – Başlangıç tipi öncelik ilişkisi	22
Şekil 2.13: Başlangıç – Başlangıç tipi öncelik ilişkisi.	22
Şekil 2.14: Başlangıç – Bitiş tipi öncelik ilişkisi.	23
Şekil 2.15: Tamamlanma yüzdesinin belirtildiği öncelik ilişkisi.	23
Şekil 2.16: Aktivitelerin oklar üzerinde gösterimi.	24
Şekil 2.17: Üç faaliyet arasındaki ilişkilerin AoA üzerinde gösterimi.	25
Şekil 2.18: İki düğümün birden fazla okla birleştirilmesi hatası.	26
Şekil 2.19: Kukla faaliyetin serim üzerinde gösterilmesi.	26
Şekil 2.20: Kukla faaliyet kullanmadan yapılan hatalı gösterim.	26
Şekil 2.21: Lojik kukla faaliyetin eklenmesi ile doğru gösterim.	27
Şekil 2.22: Kukla faaliyetin hatalı (a) ve doğru (b) kullanımı.	27
Şekil 2.23: CPM/PERT diyagramlarının standart gösterimi.	29

Şekil 2.24: Kritik yol için beklenen zaman analizi.	30
Şekil 2.25: Faaliyet ve günlerin belirtildiği CPM şebeke ağı.	33
Şekil 2.26: CPM şebekesi ileri hesap.	34
Şekil 2.27: CPM şebekesi geri hesap.	35
Şekil 2.28: Faaliyetlerin toplam bolluk hesabı.	36
Şekil 2.29: CPM şebekesinin serbest bolluk ve kritik yolları.	36
Şekil 2.30: PDM şebekesi.	38
Şekil 2.31: Pareto optimal cephede arama hedefleri (Deb, 2001).	48
Şekil 2.32: İdeal pareto çözümleri (Deb, 2001).	49
Şekil 2.33: Baskılanan ve baskılanamayan çözümler (Afruzi ve diğ., 2013).	50
Şekil 2.34: Algoritmaların sınıflandırılması.	51
Şekil 2.35: Optimizasyon Problemlerinin Sınıflandırılması (Yang X.-S. , 2010).	53
Şekil 2.36: Nümerik GA ve biyolojik evrim analogjisi.	59
Şekil 2.37: Genetik algoritmanın akış diyagramı.	63
Şekil 3.1: 18 faaliyetli test probleminin AoN tipi şebeke diyagramı.	71
Şekil 3.2: 2000 adet çözüm içeren olası çözüm uzayı.	73
Şekil 3.3: Kromozom 1'e ait CPM şebekesi diyagramı.	77
Şekil 3.4: Rulet çemberi tekniğinde bireylerin olasılık dağılımları.	82
Şekil 3.5: GA ile MGA çözümlerinin pareto grafikleri.	87
Şekil 4.1: 50 başlangıç popülasyonu ile elde edilen pareto cephesi.	93
Şekil 4.2: 100 başlangıç popülasyonu ile elde edilen pareto cephesi.	95
Şekil 4.3: 500 başlangıç popülasyonu ile elde edilen pareto cephesi.	97
Şekil 4.4: 1000 başlangıç popülasyonu ile elde edilen pareto cephesi.	98
Şekil 4.5: 5000 başlangıç popülasyonu ile elde edilen pareto cephesi.	100
Şekil 4.6: Popülasyon büyüklüğüne göre pareto çözümleri grafiği.	101
Şekil 4.7: Başlangıç popülasyonuna göre üç boyutlu pareto çözümleri grafiği.	101
Şekil 4.8: Popülasyon büyüklüğüne bağlı çözüm süresi grafiği.	102

Şekil 4.9: 50 iterasyon ile elde edilen pareto çözümleri grafiği.....	104
Şekil 4.10: 100 iterasyon ile elde edilen pareto çözümleri grafiği.....	105
Şekil 4.11: 500 iterasyon ile elde edilen pareto çözümleri grafiği.....	107
Şekil 4.12: 1000 iterasyon ile elde edilen pareto çözümleri grafiği.....	109
Şekil 4.13: 5000 iterasyon ile elde edilen pareto çözümleri grafiği.....	111
Şekil 4.14: İterasyon sayısına göre değişen pareto çözümleri grafiği.....	112
Şekil 4.15: İterasyon sayısına göre değişen üç boyutlu pareto çözümleri grafiği.....	112
Şekil 4.16: İterasyon sayısına bağlı analiz sürelerinin karşılaştırılması.	113
Şekil 4.17: GA – MGA pareto cephesi karşılaştırması.	115
Şekil 4.18: GA – MGA pareto cephesi üç boyutlu grafik karşılaştırması.	115

TABLO LİSTESİ

	Sayfa No
Tablo 2.1: Planlama sürecinin girdi ve çıktıları.....	8
Tablo 2.2: AoN tipi diyagram için proje verileri.	20
Tablo 2.3: Örnek öncelik ilişkileri tablosu.	27
Tablo 2.4: 11 aktiviteli bir projenin faaliyet listesi.	28
Tablo 2.5: CPM hesaplamaları özet tablosu.....	32
Tablo 2.6: Örnek CPM şebekesi verileri.	33
Tablo 2.7: CPM Şebekesinin sonuç değerleri.....	37
Tablo 2.8: Sıralama sayısı hesabı.	37
Tablo 2.9: Matris metot uygulaması için proje verileri.	39
Tablo 2.10: Matris metot başlangıç (setup) matrisi.	39
Tablo 2.11: Lojik ilişki matrisi.	40
Tablo 2.12: ES(Early start) –LS (Late start) matrisi.	40
Tablo 2.13: Serbest bolluk (Free Float) matrisi.	41
Tablo 2.14: Matris metot sonuç tablosu.	41
Tablo 2.15: Genetik algoritmanın sözde kodu.....	62
Tablo 2.16: Tek noktalı çaprazlama	67
Tablo 2.17: İki noktalı çaprazlama.	68
Tablo 2.18: Binary kodlama yapısında, uniform çaprazlama.	68
Tablo 2.19: Permütasyon tipi kodlamada uniform çaprazlama.....	69
Tablo 3.1: 18 faaliyetli test problemi verileri.	70
Tablo 3.2: Olası mod bileşimi ve çözüm sayısı.	72
Tablo 3.3: Örnek çözüm için genetik algoritma parametreleri.	74
Tablo 3.4: Faaliyetlerin opsiyon bileşenleri ve rassal başlangıç popülasyonu kodu.....	75

Tablo 3.5: Rassal başlangıç popülasyonu ve 1. bireye karşılık gelen opsiyonlar.....	76
Tablo 3.6: Bireylerin genlerine karşılık gelen maliyetler.....	77
Tablo 3.7: Bireylerin genlerine karşılık gelen süreler.....	77
Tablo 3.8: Bireylerin toplam süre ve toplam maliyet hesabı ile matlab kodu.....	78
Tablo 3.9: Ebeveynlerin uygunluk değerleri tablosu.....	79
Tablo 3.10: Seçim için uygunluk değerlerinin yüzde cinsinden ifadesi.....	80
Tablo 3.11: Ebeveynlerin seçim için olasılık aralıkları ve seçim operatörü kodu.....	81
Tablo 3.12: Çaprazlama için seçilen ebeveynler.....	82
Tablo 3.13: Çaprazlama gerçekleşme olasılığının incelenmesi.....	83
Tablo 3.14: Çaprazlama sonucu oluşan yeni nesil ve matlab kodu.....	83
Tablo 3.15: Her bir gen için üretilen rassal sayılar.....	84
Tablo 3.16: Genlerde gerçekleşen mutasyon ve matlab kodu.....	84
Tablo 3.17: GA yaklaşımı ile 2. iterasyonda elde edilen pareto çözümleri.....	85
Tablo 3.18: MGA yaklaşımı ile elde edilen pareto çözümleri.....	86
Tablo 4.1: Popülasyon büyüklüğünün değerlendirilmesine yönelik parametreler.....	92
Tablo 4.2: Başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.....	92
Tablo 4.3: Başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.....	94
Tablo 4.4: 500 Başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.....	95
Tablo 4.5: 1000 başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.....	97
Tablo 4.6: 5000 başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.....	99
Tablo 4.7: Başlangıç popülasyonuna göre analiz süreleri.....	102
Tablo 4.8: İterasyon sayısının etkisini ölçmek için kullanılan MGA parametreleri.....	103
Tablo 4.9: 50 iterasyon sayısı ile bulunan pareto çözümleri.....	103
Tablo 4.10: 100 iterasyon sayısı ile bulunan pareto çözümleri.....	104
Tablo 4.11: 500 iterasyon sayısı ile bulunan pareto çözümleri.....	106
Tablo 4.12: 1000 iterasyon sayısı ile bulunan pareto çözümleri.....	108
Tablo 4.13: 5000 iterasyon sayısı ile bulunan pareto çözümleri.....	110

Tablo 4.14: GA – MGA analizleri için parametreler.....	113
Tablo 4.15: GA ve MGA yaklaşımı ile analiz pareto sonuçları.	114
Tablo 4.16: MGA pareto çözümleri ve opsiyon bileşenleri.	116
Tablo 4.17: 100-136708 pareto çözümünün CPM hesaplamaları.	117
Tablo 4.18: 101-128320 pareto çözümünün CPM hesaplamaları.	118
Tablo 4.19: 104-120715 pareto çözümünün CPM hesaplamaları.	119
Tablo 4.20: 105-120170 pareto çözümünün CPM hesaplamaları.	120
Tablo 4.21: 109-119108 pareto çözümünün CPM hesaplamaları.	121



SİMGE VE KISALTMA LİSTESİ

Simgeler

Açıklama

a	: İyimser zaman
b	: Kötümser zaman
m	: Muhtemel zaman
t_e	: Beklenen süre

Kısaltmalar

Açıklama

ACO	: Karınca Kolonisi Optimizasyonu (Ant Colony Optimization)
AoA	: Aktivitelerin oklar üzerinde gösterilmesi (Activity on Arrow)
AoN	: Aktivitelerin düğüm üzerinde gösterilmesi (Activity on Node)
C	: Cost (Maliyet)
CP	: Toplam Süre
CPM	: Kritik Yol Metodu (Critical Path Method)
CProbs	: Kümülatif Olasılık Değerleri.
EA's	: Evrimsel Algoritmalar (Evolutionary Algorithms)
EF	: Early Finish (En Erken Bitiş Süresi)
ES	: Early Start (En Erken Başlama Süresi)
FF	: Finish to Finish (Bitiş-Bitiş)
FF	: Serbest Bolluk (Free Float)
Fn	: Bitiş (Finish)
FS	: Finish to Start (Bitiş-Başlangıç)
GA	: Genetik Algoritma
GERT	: Grafik Değerlendirme ve İnceleme Tekniği (GERT)
MAWA	: Modifiye Adaptif Ağırlıklandırma Yöntemi (Modified Adaptive Weigthing Approach)
MGA	: Modifiye Genetik Algoritma
MOP	: Çok Amaçlı Optimizasyon Problemi (Multiobjective Optimization Problem)
MOW	: Çok Amaçlı Ağırlıklandırma (Multiobjective Weighting)
NP-Zor	: Polinomial Zamanlı Olmayan Zor Problem (Non-Polinomial Hard Problem)

Pc	: Çaprazlama oranı
PDM	: Öncelik Diyagramı Metodu (Precedence Diagram Method)
Pm	: Mutasyon oranı
PMI	: Proje Yönetimi Enstitüsü (Project Management Institute)
Probs	: Rulet Çemberi için Olasılık değerleri
Psize	: Popülasyon büyüklüğü
Range	: Olasılık Aralığı
SFO	: Sıçrayan Kurbağa Optimizasyonu (Shuffled Frog Optimisation)
St	: Başlangıç (Start)
T	: Time (Süre)
TC	: Toplam Maliyet
TCTP	: Zaman Maliyet Ödünleşim Problemi (Time-Cost Trade-off Problem)
TF	: Toplam Bolluk (Total Float)

ÖZET

YÜKSEK LİSANS TEZİ

İNŞAAT PROJELERİNDE ZAMAN MALİYET ÖDÜNLEŞİM PROBLEMİNİN EVRİMSEL TABANLI METASEZGİSEL ALGORİTMALARLA OPTİMİZASYONU

Çağrı UYSAL

İstanbul Üniversitesi

Fen Bilimleri Enstitüsü

İnşaat Mühendisliği Anabilim Dalı

Danışman : Prof. Dr. Ekrem MANİSALI

İnşaat sektöründeki rekabet ortamı, proje planlamasının önemini giderek artırmaktadır. Firmalar rekabet avantajı sağlamak üzere proje sürelerinin kısaltılması ve/veya proje maliyetinin düşürülmesi, bununla beraber kaynakların etkin kullanımı için çabalamaktadır. Verimliliğin artması için optimum sonuçların bulunması gerekir. İnşaat projelerinde çok amaçlı zaman-maliyet ödünleşim problemi, bir optimizasyon problemi olarak ele alındığında, Np-Zor tipi problem sınıfına girmektedir ve bu tip problemlerde, sezgisel algoritmalarla, kabul edilebilir sınırlarda uygun veya optimum çözümler elde edilebilir.

Süre-maliyet ödünleşim problemi inşaat projelerinin planlamasında ve kontrolünde en önemli unsurlardandır. Planlamacılar inşaat sürecinin değişken yapısı içerisinde sürekli bir seçim kararıyla karşı karşıyadır. Uygun işgücü miktarı, ekipman, yöntemler, kaynaklar, süre, maliyet bu kararlar sonucunda değişkenlik gösterecektir. Bu ölçütlerin hepsinin birden optimize edilmesi, bunu yaparken de kabul edilebilir hesap süresi ve sonuçlarını elde edecek farklı çözüm algoritmalarından hangisinin daha efektif olacağını bulmak önem arz etmektedir.

Bu tez çalışması kapsamında literatürdeki evrimsel tabanlı meta sezgisel algoritmalar incelenerek, incelenen probleme göre modifiye edilmiş genetik algoritma literatürde kabul görmüş 18 faaliyetli test veri setine uygulanmıştır.

İnşaat projelerinin zaman-maliyet ödünleşim probleminin en iyi veya en iyiye yakın sonuçları, önerilen modifiye edilmiş evrimsel tabanlı meta sezgisel algoritmalarla analiz edilmiştir.

Ağustos 2018, 147 sayfa.

Anahtar kelimeler: İnşaat Proje Planlama, Zaman-Maliyet Ödünleşim, Optimizasyon, Meta Sezgisel Algoritmalar, Evrimsel Tabanlı Optimizasyon, CPM.



SUMMARY

M.Sc. THESIS

EVOLUTIONARY BASED METAHEURISTIC OPTIMISATION OF CONSTRUCTION PROJECTS' TIME-COST TRADE-OFF PROBLEM

Çağrı UYSAL

İstanbul University

Institute of Graduate Studies in Science and Engineering

Department of Civil Engineering

Supervisor : Prof. Dr. Ekrem MANİSALI

The competitive environment on the construction industry is increasingly important for project planning. Construction companies are striving to shorten project durations and / or reduce project costs, ensure efficient use of resources to provide competitive advantage. Optimal results need to be calculated to increase productivity. In construction projects, multiobjective time-cost trade-off problem, taken as an NP-Hard type optimization problem class. In such problems, heuristic algorithms can be used to obtain feasible or optimal solutions at acceptable boundaries.

Time-cost trade-off is an important issue in planning and control of construction projects. Planners must always make some choices within the variable structure of the construction process. The appropriate number of workers, equipment types, construction methods, resource types therefore time and cost will vary as a result of these decisions.

In this thesis evolutionary based meta-heuristic algorithms in the literature were examined and the modified genetic algorithm was applied to the 18 activity test data set accepted in the literature. The best or near-best results of the time-cost trade off problem have been analyzed with the proposed modified evolutionary meta-heuristic algorithms.

August 2018, 147 pages.

Keywords: Construction Project Planning, Time-Cost Trade-Off, Optimization, Metaheuristic Algorithms, Evolutionary Based Optimization, CPM.



1. GİRİŞ

Günümüz rekabetçi iş ortamında proje yöneticileri, projenin mümkün olan en kısa sürede minimum maliyetle tamamlanmasını hedeflenmektedir.

Proje faaliyetlerinin tamamlanması için zaman ve maliyet arasında sürekli bir ödünleşim vardır. Zaman maliyet ödünleşim problemi proje yönetiminde en önemli konulardan biridir. Genellikle kaynakların kullanımı azaldığında faaliyetin tamamlanma süresi artar. Projenin kısıtlarına ve çevresel faktörlerin durumuna göre faaliyetlerin hangi süre ve maliyetle gerçekleştirilecekleri büyük önem arz eder. Bir faaliyetin süresine veya kaynaklarına bağlı maliyetinin belirlenmesi başlı başına bir problem iken, bu faaliyetlerin birbirleri ile ilişkilendirilmesi sonucu ortaya çıkabilecek milyonlarca farklı zaman maliyet bileşiminin olması problemi karmaşık hale getirmektedir.

Literatürde, inşaat yönetiminin başlıca araştırma konuları olan; kaynak dengeleme, proje çizelgeleme, zaman maliyet ödünleşim problemleri sıklıkla incelenmiştir. Özellikle çok modlu veri seti ve çok amaçlı uygunluk fonksiyonu barındıran problemlerin çözümü için deterministik modellerin yetersiz kalması araştırmacıları stokastik modellere yöneltmiştir. Bir takım basit sezgisel algoritmalarla çizelgeleme problemlerine etkili yaklaşımlar geliştirilmiş, sonrasında daha karmaşık algoritmalar ve operatörler içeren meta sezgisel metotlarla efektif çözümler ortaya konmuştur.

1950'lerden bu yana halen canlılığını koruyan bir araştırma alanı olan evrimsel tabanlı algoritmaların (EA's) kolay kodlama yapısı, evrimsel operatörlerle geliştirilen çözüme yakınsama hızı, popülasyon tabanlı algoritma olması, lokal optimum değerlerine takılmaması gibi kullanım kolaylıklarının olması, NP-Zor tipi gerçek hayat problemlerine sıklıkla uygulanmasını sağlamıştır.

Endüstri mühendisliği alanında sıklıkla karşılaşılan üretim planlama ve optimizasyon problemlerinin yanı sıra, proje bazlı çalışılan inşaat sektörünün de büyük çaplı organizasyon ve planlamaya ihtiyaç duyması, inşaat proje planlamasının çözümü zor gerçek hayat problemleri arasına girmesine neden olmuştur.

Bu çalışma kapsamında öncelikle proje yönetimi, proje planlama, proje çizelgeleme tekniklerinden bahsedilecek, ardından zaman maliyet ödünleşim problemi tanıtılacaktır. Bu probleme yönelik çözüm önerileri optimizasyon, stokastik çözüm metotları, meta sezgisel algoritmalar, evrimsel tabanlı meta sezgisel algoritmalar tanıtılarak sunulacaktır. Genetik algoritma yaklaşımı ile çok amaçlı, çok modlu, kaynak kısıtlı, ayrık, zaman maliyet ödünleşim problemine yönelik özelleştirilmiş bir algoritma Matlab ile yazılarak, pareto değerleri, optimum sonuçların mod bileşenleri, bu bileşenlerden oluşan projenin kritik faaliyetlerinin belirlenmesi konuları işlenecektir.



2. GENEL KISIMLAR

2.1. PROJE YÖNETİMİ KAVRAMI

Proje, farklı organizasyon ve kişilerce birçok kez tanımlanmıştır. Genel olarak, hepsi bir projeyi belirli bir süre içinde sınırlı sayıda kaynak kullanarak gerçekleştirmek zorunda olan belirli hedefleri olan bir defalık bir faaliyet olarak tanımlarlar (Klein, 2000). PMI (Project Management Institute) projeyi, benzersiz bir ürün, hizmet veya sonuç elde etmek üzere yapılan geçici bir çaba olarak tanımlar. Gray ve Larson (2003)'a göre ise proje karmaşık, sıra dışı, bir defaya mahsus, süre maliyet kaynak ve performans kriterleri ile sınırlandırılmış, müşteri ihtiyaçlarını karşılamak üzere gösterilen çaba olarak tanımlanır.

Taha (2007)'ya göre proje, birbiri ile ilişkili, zaman ve kaynak tüketen faaliyetler bütünüdür. Söz konusu faaliyetler birbiriyle mantıksal bir sıra içinde ilişki halindedir, yani bazı faaliyetler onlardan önce gelen faaliyetler tamamlanmadan başlayamaz (Copertari, 2002).

Proje, planlanmış ve sonucunda katma değer yaratan bir dizi hedefi gerçekleştirmeyi üstlenen faaliyetlerden oluşan geçici bir çabadır. Faaliyetler proje hedeflerine ulaşmak için proje ekibi tarafından yürütülen önemli aktivitelerdir (Tutunji, 2010). Bir proje daha önce gerçekleşmemiş ve bir daha aynı koşullar altında gerçekleşmeyecektir (Robert ve Rudd, 2003).

Tutunji (2010) projelerin genel özelliklerini aşağıdaki gibi sıralar:

- Projelerin bir dizi amaç ve hedefleri vardır.
- Projelerin zaman ve kaynak kısıtları vardır.
- Projeler gerçekçi olmalıdır.
- Projelerde faaliyetler ve yönetim için tahsis edilmiş gruplar vardır.
- Projelere değer biçilebilir
- Projeler aşamalardan oluşur.

Burke (2001)'e göre projenin temel özellikleri şunlardır:

- Bir başlama ve bitiş noktası
- Başlangıç, bitiş ve aralarında farklı safhalar barındıran proje yaşam döngüsü.
- İlişkili bir nakit akışına sahip bir bütçe.

- Özgün ve tekrarı olmayan, bir defaya mahsus aktiviteler.
- Farklı departmanlardan gelen ve koordinasyona ihtiyaç duyan kaynakların kullanılması.
- Tek bir sorumluluk noktası (ör. Proje yöneticisi).
- Değişime tabi olan ve geliştirilmesi, tanımlanması ve kurulması gereken takım rolleri ve ilişkileri (ekip oluşturma).

Klein (2000)'a göre projelerin farklı tanımlarını kapsamlı bir şekilde analiz etmek, daha geniş bir perspektif verir ve aşağıdaki tipik özellikler karşımıza çıkar:

- Bir proje, bir dizi alt faaliyetin gerçekleştirilmesi gereken bir kerelik bir faaliyeti temsil eder. Bir projenin süresi sınırlandırılmıştır, yani tanımlanan amaçlara ulaşıldığında, proje sona erdirilir.
- Bir proje, yürütmeyi tamamen rutine indirgemekten kaçınan bazı özelliklere sahip olması bakımından benzersizdir. Dahası, alt faaliyetleri yerine getirirken dikkatli bir koordinasyon gerektirecek kadar karmaşıktır.
- Bir projeyi tamamlamak, bir üst kuruluşun veya farklı kuruluşların birkaç işlevsel bölümünün işbirliğini gerektirir. Buna ek olarak, bir proje eşzamanlı yürütülen diğer projelerle etkileşime girebilir.
- Bir projeyi yürütmek için mevcut kaynaklar kısıtlıdır. Bütçe, uygun insan kaynakları ve ekipmanın bulunabilmesi için önemlidir.
- Bir projenin uygulanması belirgin bir derecede belirsizlik içerebilir. Başlıca belirsizlik sebepleri arasında, kaynakların performansındaki değişiklikler, eksik veya yanlış veriler ve tecrübe eksikliği nedeniyle gelecek olayların tatmin edici bir şekilde tahmin edilememesi yer alır.

Tüm bu özellikler, bir projenin gerçekleştirilmesinin kapsamlı yönetim katılımına ihtiyaç duyan karmaşık bir görev olduğunu göstermektedir. Projeler, başlangıcından sonuna kadar planlama ve denetlemeye gereksinim duyar. Projelerin genel özellikleri ve karmaşık yapısından dolayı, dikkatli ve özenle yönetilmesi gerekir.

İnşaat projelerinin tam zamanında ve belirli kaynaklarla tamamlanması, sektörünün rekabetçi doğasından dolayı zorunludur. Bu nedenle uygun proje planlama teknikleriyle etkin bir proje yönetimine ihtiyaç vardır.

Projelerin tüm paydaşlarının projeden farklı beklentileri vardır. İşveren, istediği kalite düzeyine erişilmesini en uygun fiyatla beklerken; yüklenici firma projeyi daha çok zaman ve maliyet açısından değerlendirir. Proje yönetiminde bilgi, tecrübe, araç ve tekniklerin, ihtiyaç ve proje beklentilerini karşılamaya yönelik kullanılması söz konusudur. Projenin hedeflerine ulaşabilmesi için proje faaliyetlerinin planlanması, programlanması (çizelgelenmesi) ve kontrol edilmesi zorunludur.

Uluslararası bir meslek kuruluşu olan RICS'a göre proje yönetimi işverenin beklentisini karşılayacak süre, maliyet ve kalite düzeyinin sağlanabilmesi için projenin baştan sona kadar planlanması kontrolü ve koordinasyonu olarak tanımlanır. Literatürde proje yönetimi hakkında birçok tanımlama mevcuttur:

Young (1998)'a göre proje yönetimi, açıkça belirlenmiş stratejik hedeflere ulaşmak için; organizasyonun uygun kaynaklarını kontrollü ve yapılandırılmış bir şekilde kullanan dinamik bir süreçtir. PMBOK proje yönetimini, proje gereksinimlerini karşılamak için, bilgi beceri araç ve tekniklerin proje faaliyetlerine uygulanması olarak tanımlar.

Tutunji (2010)'ye göre ise proje yönetimi; belirli kapsam ve kısıtlar dâhilinde, belirli amaç ve hedefleri karşılayacak projelerin sonuçlandırılması için planlama, örgütlenme, izleme, kontrol ve kaynak denetimini içeren disiplindir.

Projeler boyut ve karmaşıklık bakımından büyüdüğünde, projeleri planlama ve kontrol etme kabiliyeti, çok önemli bir proje yönetimi fonksiyonu haline gelmektedir. Proje yöneticisi, problem çözme ve karar verme sürecini kolaylaştırmak için büyük miktarda veriyi hızlı ve doğru bir şekilde planlamak, yönlendirmek, gözlemlene ve kontrol etmek için bütünleşik bir bilgi ve kontrol sistemi geliştirmelidir. Bu hedeflere ulaşmak için, proje yöneticisi kapsamlı bir araç setine ihtiyaç duyar , organizasyon şemaları, iş alt kırılım yapıları, çubuk diyagramlar, kaynak histogramları ve nakit akış tabloları üretir. (Burke, 2001)

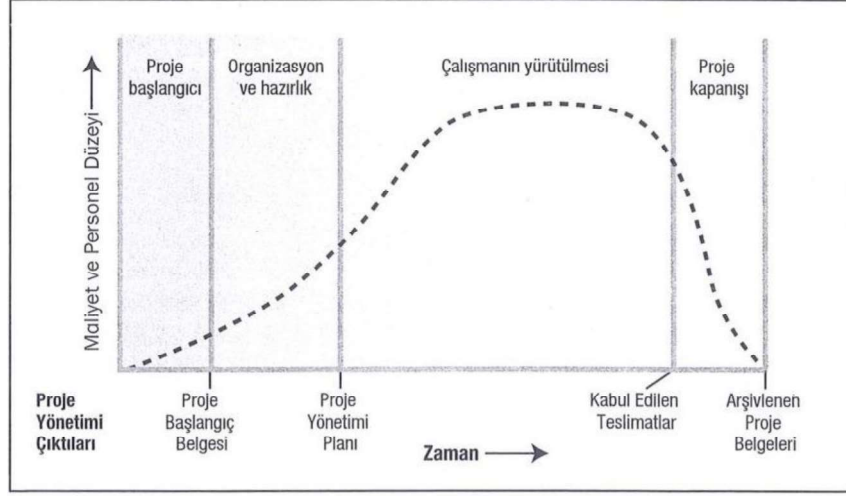
İnşaat endüstrisindeki her tür değişikliğe rağmen, proje yöneticisinin rolü; iş süreci ve çalışmaya ilişkin maliyetlerin kontrolünü yapmak üzere sabit kalır. Levy (2012) Bir inşaat projesinin yönetimi dört ana bileşene ayırmıştır:

- **Yapım Mühendisliği:** Malzemelerin, bileşenlerin, ekipman ve sistemlerin uygun tekniklerle montajı, en uygun yapım teknolojisinin seçimi ve kullanımı.

- **İnşaat Sürecinin Yönetimi:** İnşaat sürecini hayata geçirmek için; işçi, malzeme ve ekipmanların akışının kontrol ve koordinasyonu için bir iş planı da barındıran, en etkin yolun oluşturulması.
- **İnsan Kaynakları Yönetimi:** İşgücü verimliliği ve uyumlu bir çalışma ortamı başarılı bir projenin temel unsurları olduğundan, insan kaynağı kontrolü, kalifiye işçi ve deneyimli yöneticilerin kısıtlı olduğu günümüzde daha çok önem kazanmıştır.
- **Finansal Yönetim:** İnşaat tarihsel olarak düşük kar marjları ile yüksek riskli bir iştir. Maliyetler, nakit akışları ve yeterli proje finansmanının sağlanması işin başarısı için kritik öneme sahiptir.

Yapı projelerinde belirli bir zaman ve bütçe kısıtı olduğundan proje yöneticilerinin bu iki kısıt üzerinde bir denge kurması gerekir. Genellikle projelerde kullanılan kaynakların sınırlı olması nedeni ile faaliyet süre ve ilişkilerinde değişiklikler yapılarak bu denge kurulur. Örneğin bir faaliyette çalışan işçi sayısı arttıkça faaliyet süresinin kısalması, buna bağlı olarak paralel faaliyetlerdeki işçi sayısının azalması ile bu faaliyetlerin sürelerinin artması söz konusu olabilir. Zaman maliyet optimizasyonu proje yöneticisinin kararlarını ve projenin durumunu doğrudan etkilediğinden önemli bir proje yönetimi problemi olarak karşımıza çıkar.

Projeler özgün, karmaşık ve belirli derecede risk içerdiğinden şirketlerin daha iyi yönetim kontrolü sağlamak için projelerini birkaç safhaya ayırması kaçınılmaz olmuştur. Çoğu proje, başlangıçtan bitime kadar benzer aşamalardan geçer. Proje Yaşam Döngüsü projenin başlangıç aşamasından kapanışa kadar geçen tüm proje uygulama sürecidir. PMBOK' a (2008) göre Proje Yaşam Döngüsü sıralı ve iç içe geçen proje fazları grubudur. Yaşam Döngüsü projeye ilişkili özgün işlerden bağımsız bir şekilde proje yönetimine ilişkin temel çerçeveyi sağlar. Çoğu projenin Şekil 2.1' de gösterildiği gibi, dört aşamalı bir yaşam döngüsünden geçtiğine dair genel bir görüş vardır (Burke, 1992).



Şekil 2.1: Proje yaşam döngüsü (PMI, 2008).

Proje başlangıç aşaması tanımlama aşamasıdır. Projenin özellikleri belirlenir, proje hedefleri belirlenir, proje ekipleri oluşturulur ve ana görevler atanır. Organizasyon ve hazırlık aşamasında planlama yapılır. Proje aşamaları, paydaşlar, zaman çizelgeleri, kalite standartları ve bütçeyi belirlemek için planlar geliştirilir. Hem fiziksel, hem zihinsel anlamda en fazla çaba sarf edilen aşama çalışmanın yürütülmesi aşamasıdır. Kontrol için zaman, maliyet ve şartname bilgileri kullanılır. Proje yöneticileri, projenin planlanmış bütçe ve zaman ve teknik şartnamelerde belirtilen özelliklerde devam ettiğinden emin olmalı, herhangi bir değişikliğin gerekli olup olmadığını kontrol etmelidir. Proje kapanışı aşamasında tamamlanan projenin teslimi gerçekleştirilir.

2.2. PROJE PLANLAMA VE ÇİZELGELEME

Bir inşaat projesini veya herhangi bir projeyi planlamak temelde aynıdır. Aktivitelerle bu aktivitelere atanan kaynaklar (iş gücü, malzeme, makine vb.) birbirini karşılamalıdır. İnşaat projeleri, çok fazla parametre içerdiğinden, inşaat projelerinin çizelgelenmesi daha karmaşıktır. Karmaşıklığın sebepleri aşağıda maddeler halinde verilmiştir:

- İnşaat projelerinde, yüksek sayıda aktivitenin organize edilerek doğru sıraya konması için büyük bir bilgi birikimi gerekmektedir.
- Projelerde yüzlerce bağımsız kaynak kullanılabilir ve bu kaynaklar, çoğunlukla hem miktar hem de zaman boyutunda kısıtlıdır.

- Proje çizelgelemesi yapılırken projenin majör fazlarının son tarihleri, inşaatın tamamlanma zamanı, bütçe kısıtları, proje değişiklikleri ile sınırlandırılmak çoğu zaman proje müdürü için en ciddi problemdir.
- Karmaşık hesap işleri çizelgelemeyi zorlaştırır.

Uygun ve iyi tasarlanmış bir proje planı fazla maliyet ve istenmeyen gecikmeleri önler. Bunun sağlanabilmesi için endüstride; proje yöneticilerinin çalışmalarını daha verimli hale getirebilmeleri için özel planlama yöntemleri geliştirilmiştir (Hadju, 1997).

Planlama, hedefe ulaşmak için takip edilmesi gereken net bir yol haritası belirlemek için kullanılır. Planlama, projeleri tanımlanabilir ve ölçülebilir faaliyetlere bölerek bu faaliyetler arasında mantıksal ilişkiler kurar. Planlama genellikle aşağıdaki üç temel soru sorularak yapılır:


- Ne yapılacak?
- Nasıl yapılacak?
- Kim yapacak?

İnşaat projelerinde planlar; kurumsal stratejik planlar, ihale öncesi planlar, sözleşme öncesi planlar, kısa vadeli inşaat planları ve uzun vadeli inşaat planları olmak üzere çeşitli düzeylerde olabilir. Bu planlar birbirinden farklı olmalarına rağmen dört ana adımdan oluşur:

- Projeye dâhil olan faaliyetlerin iş alt kırılım yapısının oluşturulması.
- Faaliyetlerin yürütülmesi için uygun sıranın belirlenmesi.
- Faaliyetlerin çizelgelenmesi.
- Bireysel faaliyetlerin kaynaklarının, zamanının ve maliyetinin tahmin edilmesi.

İnşaat projelerinde planlama sürecinin girdi ve çıktıları Tablo 2.1’de gösterilmiştir.

Tablo 2.1: Planlama sürecinin girdi ve çıktıları.

GİRDİLER		ÇIKTILAR
Sözleşme Bilgileri	PLANLAMA SÜRECİ 	Faaliyetler
Proje Teknik Çizimleri		Faaliyetler Arası İlişkiler
Proje Özellikleri		Yapım Yöntemi Beyanı
Mevcut Kaynaklar		İş Tanımları ve Sorumluluklar
Kaynak Miktarı		Kilometre Taşı Raporları
Şantiye Sahası Raporları		Proje Ağ Diyagramı
Örgütsel veriler		Faaliyetlerin Süresi
Yapım Yöntemleri		Faaliyetlerin Maliyeti

Aşağıdaki adımlar bir proje planı oluşturmak için bir rehber veya kontrol listesi olarak kullanılabilir (Ebeltagi, 2014):

- İşin kapsamının belirlenmesi, metodun bildirilmesi ve işin sıralaması
- Faaliyetlerin tamamının bir listesini oluşturmak için iş alt kırılım yapısının oluşturulması
- Organizasyon şemasını oluşturarak faaliyetlere sorumlularının atanması
- Aktiviteler arası ilişkilerin belirlenmesi
- Faaliyetlerin süre, maliyet ve kaynak ihtiyaçlarının tahmini
- Proje çizelgesinin oluşturulması

Projedeki faaliyetlerin planlaması ve kontrolünü kolaylaştırmak için faaliyetler yönetilebilir ve karmaşık olmayan alt faaliyet gruplarına ayrıştırılmalıdır. Ayrıştırma sürecinde amaç her bir aktivitenin zaman, maliyet ve iş içeriğinin yönetilebilir bir yapıya kavuşturulmasıdır. Böylece kontrol sürecinde her bir iş grubu belirlenecek olan yönetici ve ekibinin proje bileşenlerini yönetmesine ve etkili kontrol yapmasına imkân verecektir. (Burke, 1992)

WBS, mantıksal olarak projenin tüm çalışma öğelerini grafik sunum haline getirmek üzere tasarlanan hiyerarşik bir yapı olarak tanımlanmaktadır. Proje yöneticisinin, çalışmayı küçük yönetilebilir çalışma paketlerine bölmesine yardımcı olur.(Klein, 2000) Proje için işin genel kapsamı şemanın tepesine yerleştirilir ve daha sonra her bir alt seviyedeki iş alt öğelere ayrılır. WBS'nin en alt düzeyinde iş unsurlarına iş paketi denir. WBS'nin etkin kullanımı projenin kapsamını ve her bir çalışma paketinin sorumluluğunu ana hatlarıyla ortaya çıkaracaktır.

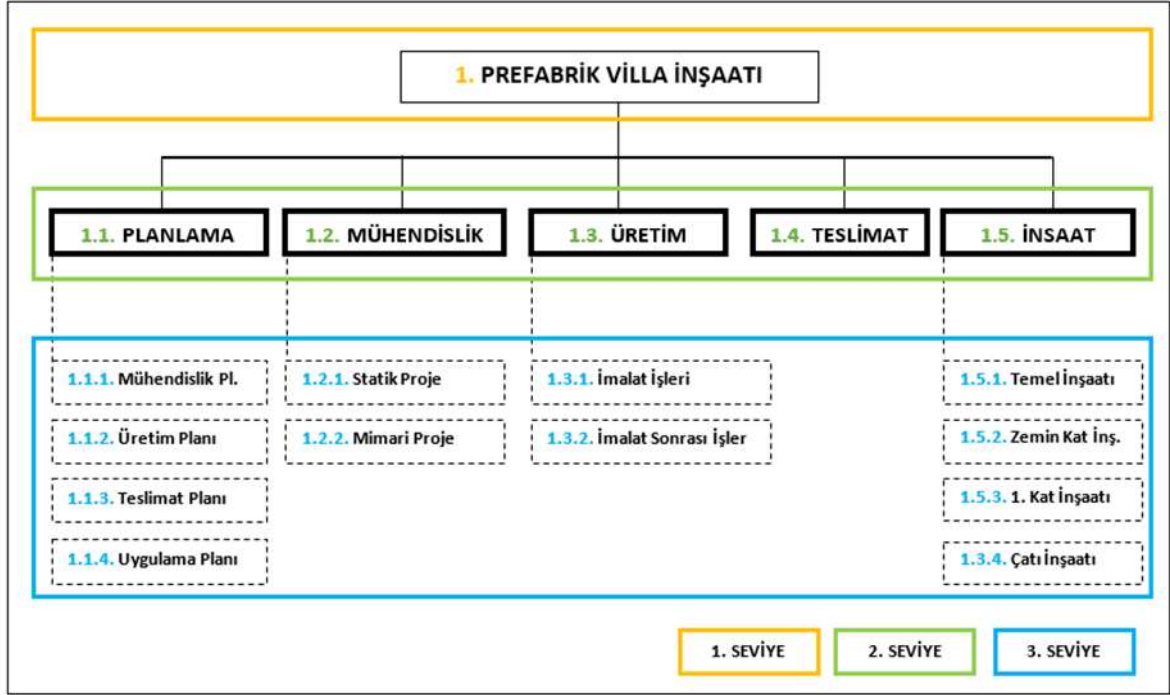
WBS oluşturulurken faaliyetler, tüm anlamlı görevler veya iş paketleri tanımlanmaya ve her bir görev veya paket tek tek planlanabilir, bütçelenebilir, planlanır, izlenir ve kontrol edilebilir seviyeye gelene kadar parçalanmaya devam edilmelidir. Açıktır ki, iş paketleri eksikse veya doğru bir şekilde düzenlenmemişse, projenin istenen zaman, bütçe ve özellikte tamamlanması mümkün değildir. (Meredith ve Mantel, 2009)

WBS uygulamaları proje detayına göre değişmekle birlikte genellikle üç seviyede tasarlanabilir:

- Seviye 1: Projenin kendisi / Nihai teslimat
- Seviye 2: Seviye 1'den türetilen farklı safhalar / teslimatlar

- Seviye 3: Seviye 2'den türetilen iş paketleri

Şekil 2.2'de prefabrik bir villaya ait bir WBS uygulaması gösterilebilir.



Şekil 2.2: Prefabrik bir yapıya ait iş alt kırılım yapısı gösterimi.

İnşaat endüstrisi, inşaat projelerinin bütçe ve süre kısıtları dolayısı ile giderek daha rekabetçi bir hal almaktadır. Bu rekabetçi piyasa koşullarında, bir projenin planlanan bütçe ve süre zarfında tamamlanması önem kazanmaktadır. Projedeki gecikmeler yüklenicilerin saygınlığını ciddi şekilde etkileyebilir. Bunlara ek olarak inşaat projeleri, çeşitli aktiviteler, farklı ekipler, farklı taraflar içerir ve tüm bunlar birbirleri ile karmaşık ilişkilere sahiptirler. Bu nedenle, inşaat projeleri dikkatli planlama ve kapsamlı kontrol gerektirir. Kısaca inşaat sektöründe daha iyi planlama ve kontrol, projeyi başarıya ulaştırırken; zayıf planlama ve kontrol, anlaşmazlıklar ve anlaşmazlıkların yanı sıra ciddi finansal kayıplara neden olabilir.

İnşaat projelerinin gelişimini takip etmek ve yapım sürecinin nasıl ilerleyeceğini müşteri veya işverene anlaşılır bir şekilde ifade etmek için simgesel gösterimlerle desteklenmiş çizelgeleme tekniklerine ihtiyaç vardır. Bu ihtiyacı gidermek üzere geliştirilmiş grafiksel proje planlama teknikleri proje yönetiminde oldukça faydalıdır.

Çizelgeleme, pek çok faaliyeti içeren bir işe ait faaliyetlerin, atanmasına yönelik bir karar sürecidir. Proje yönetiminde çizelgeleme projeyi oluşturan faaliyetlerin öncelik ilişkileri, süreleri ve kaynaklar dikkate alınarak atanması/sıralanmasıdır. Çizelgeleme aşamasında proje elemanlarının hepsi birden dikkate alınmalıdır. Proje temel öğelerini; faaliyetler, faaliyetler arasındaki ilişkiler ve kaynaklar oluşturmaktadır (Namlı, 2012).

Çizelgeleme problemlerine yönelik ilk bilimsel düşünce 21. Yüzyılın başlarına, Henry Gantt'ın çalışmasını yayınladığı tarihe dayanır. Bu çalışma bugünkü bar diyagramların temeli olarak düşünülebilir. Bar diyagram düşey eksenle aktiviteler sıralanırken, yatay eksen zamanı belirtir (Haidu, 1997).

Projelerin büyüklükleri ve karmaşıklığı sebebiyle planlanması ve kontrolü proje yönetimi açısından oldukça zor ve önemlidir. Temelde proje planlama projeye ilişkin bir çizelge oluşturulması ve bu çizelge kapsamındaki faaliyetlerin bütçelenmesi esasına dayanır. Bu alanda yapılan çalışmalar ise, genellikle ağ(şebeke) planlama yaklaşımlarıyla yapılmaktadır. Bunlar arasında en çok bilinenleri CPM (Kritik Yol Metodu) ve PDM (Precedence Diagramming Method)'dir.

Proje yöneticisi iyi bir planlama ve çizelgeleme için öncelikle aşağıdaki adımları gerçekleştirmelidir:

- Tamamlanması gereken faaliyetleri tanımlanması
- Faaliyetlerin sırasının belirlenmesi
- Faaliyetlerin bağımlılık ilişkilerinin belirlenmesi
- Faaliyet sürelerinin belirlenmesi
- Görevlere kişilerin atanması
- Kilometre taşlarının belirlenmesi (Tutunji, 2010).

Elbeltagi (2009)'a göre yeni bir proje takvimi başlatıldığında yapılacak ilk iş bireysel görevlerin belirlenmesidir. Projenin amaç ve kapsamını bilmek, bunlara ulaşabilmek için gerçekleştirilmesi gereken adımların belirlenmesine yardımcı olur. İlk görev oluşturulmadan önce proje kapsamı dâhilinde daha ayrıntılı proje parametreleri için düşünülmesi gerekmektedir.

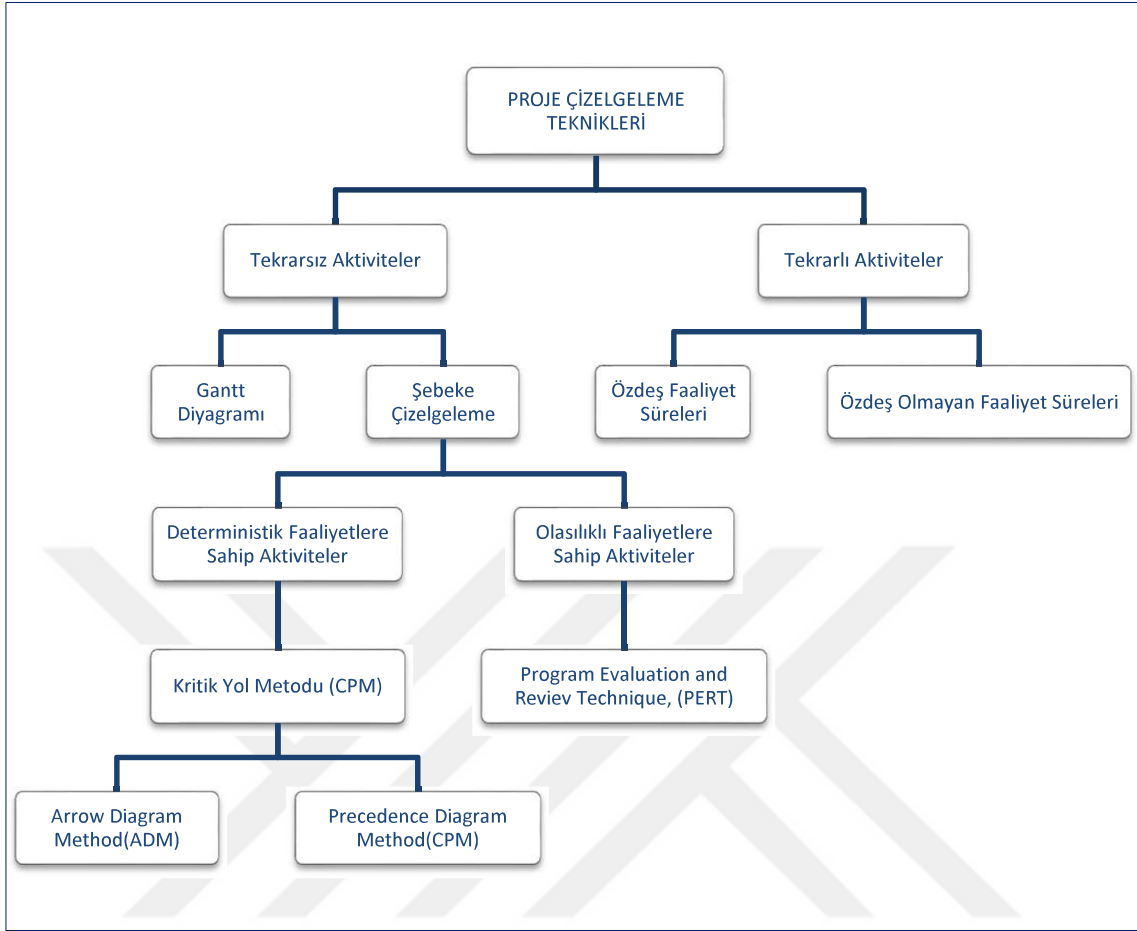
- Lojistik Girdileri: Proje sırasında üretilen somut ürün, hizmet veya sonuçlardır. Projenin bir yerinde her bir çıktının teslimatını yansıtacak görevler olmalıdır.
- Önemli Tarihler: Proje bitiş tarihine ek olarak proje boyunca diğer önemli tarihler de belirtilmelidir.
- Tamamlama Kriteri: Projenin tamamlanma kriterini bilmek, çalışanların hedefe odaklanmasını sağlar ve projenin son fazını oluşturmaya yardımcı olur.
- Beklentiler: Çalışanlar ve yönetimden beklenti bilinirse bazı görevlerin tanımlanması kolaylaşır.
- Potansiyel Riskler: Potansiyel sorun alanlarını tanımlamak, ortaya çıkabilecek sorunların kontrolü ya da en aza indirilmesi için yardımcı olur.

Yöneticiler, rekabet gücü yüksek endüstrilerin karakteristik olan karmaşıklıkları, veri yığınları ve sıkı teslim tarihleri ile başa çıkmak için sürekli yeni ve daha iyi kontrol teknikleri araştırmaktadır. Arica müşterilere teknik ve maliyet verilerini sunmak için daha iyi yöntemlere gereksinim duymaktadırlar.

Çizelgeleme teknikleri bu hedeflere ulaşmanıza yardımcı olur. En genel teknikler şunlardır:

- Gantt veya çubuk grafikler
- Kilometre taşı grafikleri
- Denge çizgisi
- Şebeke diyagramları
- Program Değerlendirme ve İnceleme Tekniği (PERT)
- Ok Diyagram Yöntemi (ADM) [Kritik Yol Metodu (CPM)]
- Öncelik Şeması Metodu (PDM)
- Grafik Değerlendirme ve İnceleme Tekniği (GERT)

İnşaat Sektöründe karşılaşılan çizelgeleme problemlerine göre kullanılacak proje çizelgeleme teknikleri Şekil 2.3'te gösterilmiştir (Aziz, 2013):



Şekil 2.3: İnşaat projeleri için mevcut planlama teknikleri (Aziz, 2013).

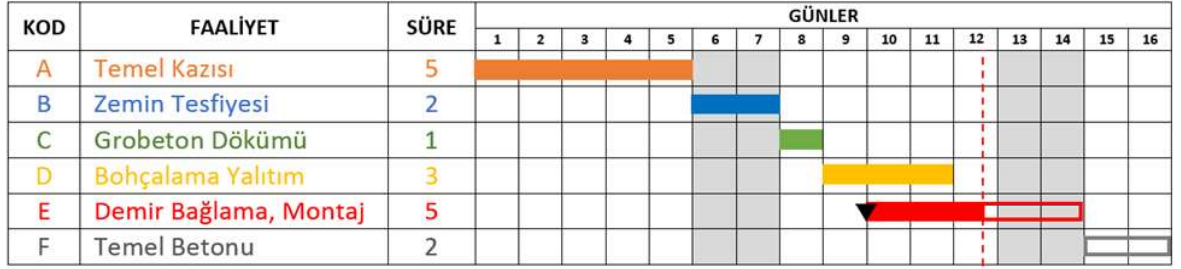
Gantt Diyagramı

Gantt Grafiği veya çubuk diyagram olarak da adlandırılan Gantt Şeması 1918’de Henry Gantt tarafından geliştirilmiştir. Oluşturması kolay ve en yaygın kullanılan grafiksel planlama/programlama metodudur. Gantt Şeması ile Ağ diyagramlarından elde edilen sonuçlar çabuk ve etkili bir biçimde aktarılabilir. Uygun olarak izlenildiği takdirde zamanı yönetmek ve sorunları bulmak için kullanılabilir. Çok çabuk ve kolay hazırlanması en önemli özelliğidir. Fakat bu özelliğinin yanında Gantt şeması geniş bilgiler ve teknik isteyen kapsamlı projelerde yetersiz kalmaktadır (Albayrak, 2009).

Bar diyagram, bugünlerde bile inşaat uygulamalarında geniş kullanım alanı bulan, anlaması kolay ve etkili bir araçtır. En büyük dezavantajı faaliyetler belirli bir sayıyı aştığında ortaya çıkar. Gantt şeması faaliyetler arasındaki ilişkileri değil sadece faaliyetlerin ne zaman başlayacağını belirtir. Binlerce faaliyetli bir projede herhangi bir faaliyet programının gerisinde kaldığında, bu gecikmeden etkilenecek faaliyet sayısı bilinemeyecektir. Tüm artı ve eksileri

düşünüldüğünde Gantt şeması, küçük projelerin planlanmasında ve her tip projenin görselleştirilmesinde kullanılabilecek iyi bir araçtır (Tutunji, 2010).

Gantt şemasında dikey eksen faaliyetleri, yatay eksen zamanı belirtir. Faaliyetlerin başlangıç, bitiş zamanı ve süreleri bellidir. Faaliyetler süreleri ile orantılı kutularla zaman çizelgesinde belirtilir. Şemanın türüne bağlı olarak değişkenlik göstermekle beraber, tamamlanan devam edilen işler bir simge ile görselleştirilebilir. Proje için önemli faaliyet veya olaylar kilometre taşı olarak genellikle üçgen şeklinde sembollerle gösterilir. Faaliyetler arası öncüllük ilişkileri genellikle gösterilmemekle beraber oklarla belirtilen diyagramlarda mevcuttur. Şekil 2.4'te bir inşaat projesinin temel inşaatı ile ilgili; Gantt şemasının genel özelliklerini barındıran bir diyagram verilmiştir:



Şekil 2.4: Temel inşaatı işlerinin Gantt Şeması ile gösterimi.

Tutunji (2010)'ye göre Gantt grafiğinin avantajları:

- Proje planının kolay çizelgelenmesini ve okunmasını sağlar.
- Projedeki iş akış sırasını doğru bir şekilde ortaya koyar.
- Zaman belirgindir.
- Tüm faaliyetler görülür.
- Proje bitiş tarihleri gösterilir.
- Proje mevcut durumu ve gelişimi tespit edilebilir.

Albayrak (2009)'a göre Gantt Şemasının dezavantajları şu şekilde sıralanabilir:

- Muhtemel sorunlar ve faaliyet aşamaları arasındaki zayıf ilişkileri tanımlamaz.
- Faaliyet aşamalarından birinde meydana gelecek bir gecikmenin, bundan sonraki aşamaları nasıl etkileyeceğini belirtmez.
- Kritik noktalar hakkında bilgi içermez.
- Gerekli kaynaklar ile gereksinimleri koordine etmez.

Özetle Gantt şeması faaliyet ve kapsam yönünden küçük boyutlu projelerin tanımlama, izleme ve kontrolü için kullanılabilir en uygun araçtır. Çabuk ve kolay hazırlanması yönünden tercih edilebilir. Bunun yanında geniş bilgi ve teknik isteyen, faaliyetler arası ilişkileri fazla olan projelerde yetersiz kalabilir. Sunum ve görsel olarak anlaşılabilirliğin önemli olduğu durumlarda kullanımını daha uygundur.

2.3. ŞEBEKE ÇİZELGELEME TEKNİKLERİ

Planlamada kullanılan, şebeke çizelgeleme metotlarından en çok bilinenleri, CPM (Critical Path Method), PERT (Program Evaluation and Review Technique) ve Monte Carlo Simülasyon tekniğidir. Bu metotlardan en yaygın ve bu tez kapsamında da kullanılanı CPM' dir.

Şebeke planlama tekniklerinin inşaat yönetimine olumlu birçok katkısı vardır. Şebeke tekniklerinin avantajları şunlardır (Manisalı, et al., 2000; Kerzner, 2009) :

- Belirli bir süre dâhilinde gerçekleştirilecek inşaat projesi, matematiksel model haline gelir.
- Duyarlılık analizi yapılmasına olanak sağlar. Projenin ilerlemesi sırasında farklı durumlar sonucunda nelerin ortaya çıkabileceğinin değerlendirilmesini sağlar.
- Proje paydaşları arasında çıkabilecek ihtilafların çözümünde hakem rolü oynar.
- Projenin ilerleyişinin kolaylıkla ölçümü ve denetimini sağlar. Programları kontrol etmek için görünürlük sağlar ve yönetimi kolaylaştırır.
- Nakit akış analizinin yapılabilmesine olanak verir. Maliyet kontrolü yapılabilmesini sağlar.
- Zaman ve maliyet amaçlarına ulaşmak, kaynaklarını nasıl kullanacaklarına karar vermek için kullanılan tüm planlama ve tahmin yöntemlerinin temelini oluşturur.
- Yönetimin, alternatif gecikmelerin projenin tamamlanmasını nasıl etkileyeceği, faaliyetler arasında gevşeme bulunduğu durumlarda; tamamlanma tarihini karşılamak için hangi faaliyetlerin önemli olduğu sorusuna cevap vererek alternatiflerin değerlendirilmesine yardımcı olur.
- İlerlemeyi kontrol etmek için bir araç olmasının yanı sıra, insan gücü, malzeme ve sermaye gereksinimlerini belirlemek için temel yöntem olarak adlandırılan şebeke analizi kullanır.
- Raporlama için temel oluşturur.

- “Eğer (What if?)” analizine olanak sağlar.
- En uzun ve kritik yolu belirler.
- Sınırlı kaynakların projeye etkisini gösterir.
- İhtiyaçların değişmesinin projeye etkisini gösterir.
- Proje için (ve her WBS ögesi için) nakit akışı belirler.
- Fazla mesainin projeye etkisini gösterir.
- Projenin kısıtlamalarını karşılamak için ek kaynaklara ihtiyaç olup olmadığını gösterir.
- Belirli bir WBS unsurunun önceliğindeki bir değişikliğin projenin tamamını nasıl etkileyeceğini gösterir.

Proje çizelgelemede, faaliyetler ilgili öncüllük ilişkilerine göre sıralanırken bir ağ yapısı ortaya çıkar. Faaliyetlerin süreleri, öncül ve ardılları, en erken ve en geç başlama ve bitiş süreleri, bollukları vb. bilgilerin gösterildiği bu ağ benzeri yapı şebeke diyagramları olarak adlandırılır.

Şebeke diyagramları, faaliyet ve olaylardan oluşur. Şebeke diyagramlarını daha iyi anlamak için aşağıdaki terimlerin bilinmesinde yarar vardır (Kerzner, 2009):

- **Olay (Event):** Zaman içerisinde meydana gelen bir veya birden fazla paralel faaliyetin başladığı ya da sonuçlandığı durumu gösterir ve gerçekleşmesi için hiçbir kaynak ve zaman kullanımını gerektirmez (Sezen, 1994).
- **Faaliyet (Activity):** Projeyi oluşturan ve belirli bir zaman dilimi içerisinde ve belli miktarda kaynak tüketerek tamamlanması gereken, en küçük proje görevi birimidir.
- **Süre (Duration):** Etkinliği tamamlamak için gereken toplam süredir.
- **Çaba (Effort):** Belirli bir süre içinde gerçekten yapılan iş miktarı. Örneğin, bir etkinliğin süresi bir ay olabilir, ancak çaba süre içinde yalnızca iki haftalık bir kısmı kapsayabilir.
- **Kritik Yol (Critical Path):** Ağ üzerinden en uzun yol budur ve projenin süresini belirler. Ayrıca, projeyi gerçekleştirmek için gerekli olan en kısa zaman miktarıdır.

Herhangi bir çizelgeleme sorununun tanımlanmasında önemli bir nokta, işlemciler (makinelere) tarafından işlenecek görevler arasındaki öncelik kısıtlamalarıdır. Farklı alanlardan ortaya çıkan bu kısıtlamalar, teknolojik zorunluluktan doğan sıralamaları, aktivite önceliklerini veya paralel ve sıralı görevleri temsil etmek için kullanılır. Öncelik kısıtlamalarını tasvir etmek için iki temel grafiksel gösterimin kullanılabileceği bilinmektedir. Bunlardan ilki aktivitelerin oklar üzerinde

gösterildiği AoA (Activity on Arcs) yani her faaliyeti simgeleyen bir ok ve belirli olayları temsil eden düğümlerden oluşan ağ çizelgeleme tekniğidir. Diğer bir teknik ise faaliyetlerin düğüm noktaları üzerinde gösterildiği (activity on node) ağ yapısıdır. AoN yapısında düğümler aktiviteleri ve oklar öncelik ilişkilerini temsil eder. AoA yapısı ağırlıklı olarak proje çizelgelemede kullanılırken, AoN yapısı programlamaya daha yatkın olmasından dolayı bilgisayar destekli problem çözümlerinde daha sık kullanılır (Blazewics & Kobler , 2002).

Bir projeyi tanımlamak için üç tür bilgi gereklidir.

- Faaliyet bilgileri: Proje bireysel faaliyetlerine (istenilen detay seviyesinde) ayrılmalıdır.
- Öncelik ilişkileri: Her bir etkinlik için öncül veya öncüller tanımlanmalıdır.
- Zaman bilgisi: Her etkinliğin süresini tahmin edilmelidir.

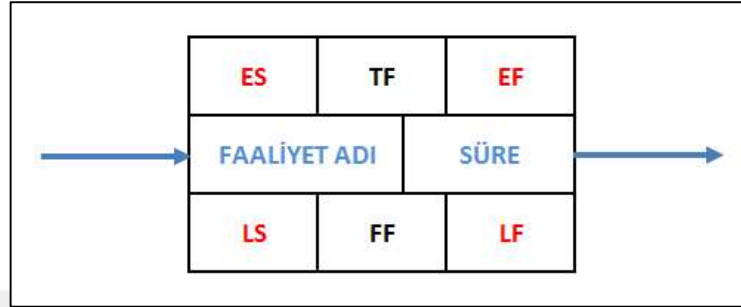
Proje ağı tüm bu bilgileri içermelidir. Bunu yapmak için iki alternatif proje ağ yapısı kullanılabilir.

2.3.1. Faaliyetlerin Düğümler Üzerinde Gösterimi

Proje yönetiminin uygulama alanlarının gün geçtikçe artması ve farklı problem tiplerinde proje yönetimi araç ve tekniklerinin kullanılması yazılım alanında da karşılık bulmuştur. Bilgisayar yazılımları ve programlama dilleriyle uyumlu grafiksel gösterimlere ihtiyaç duyulduğundan öncelik diyagramları gibi şebeke çizelgeleme teknikleri geliştirilmiştir.

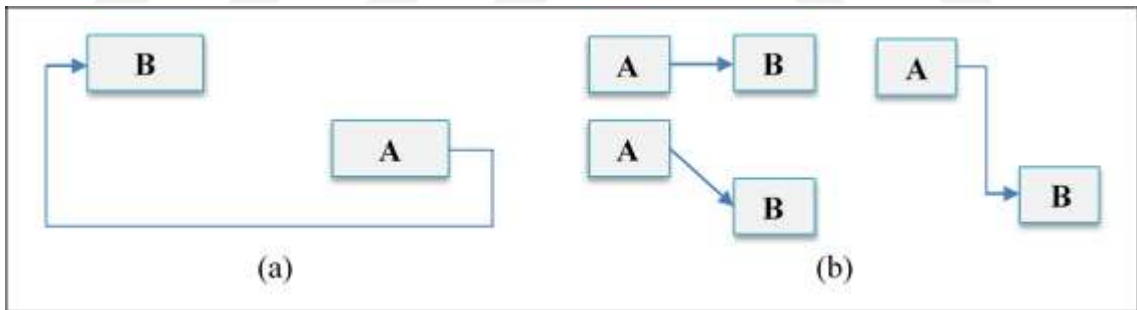
Literatürde kullanılan öncüllük ilişkilerinin en yaygın gösterimi; aktivitelerin düğümler üzerinde gösterildiği AoN (Activity on Node / Vertex) diyagramı gösterimidir. Bir ağ diyagramda temel analiz birimi faaliyettir. Şekil 2.5’ te görüldüğü gibi ağ diyagramındaki her faaliyet, düğüm olarak adlandırılan bir dikdörtgen veya daire gibi görsellerle gösterilir. Oklar, etkinlikler arasındaki öncül / ardıl ilişkilerini temsil eder (Wyssocki & McGary, 2003). AoN diyagramlarında kukla faaliyete gereksinim olmadığından dolayı daha az teknik bilgi gerektirir ve çizimi ok diyagramlara göre daha kolaydır. Estetik sebepler ve bilgisayar hesapları dolayısıyla projenin tek bir faaliyetle bağlayıp tek faaliyetle bitmesi tercih edilir. Birden fazla faaliyetle başlayan veya biten projelerde hayali faaliyetler (Başlangıç “St” ve Bitiş “Fn” faaliyetleri) gerekebilir.

Düğümelerde faaliyetlerin karakteristik bilgileri ve zamanla ilgili bilgiler tutulur. Literatürde genellikle, faaliyet ID, faaliyet adı, faaliyet süresi, ES, EF, LS, LF, TF, FF değerlerine ek olarak faaliyet maliyeti, tamamlanma yüzdesi, kalan süre de veri olarak düğümlerde gösterilebilmektedir. Şekil 2.5’te örnek bir faaliyet düğümü gösterilmiştir.



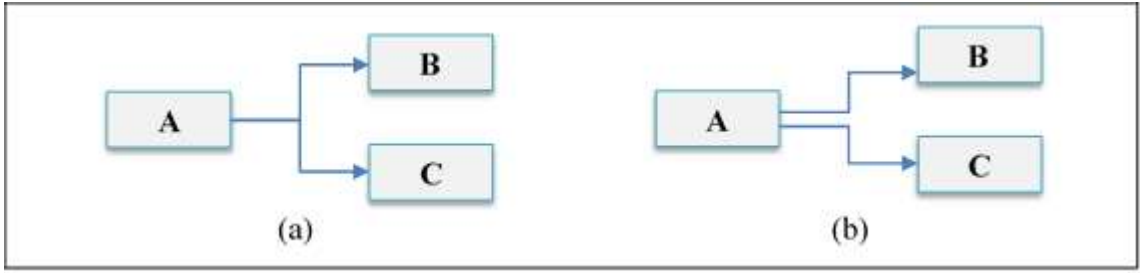
Şekil 2.5: Aktivite düğümüne ait standart veriler.

AoN Şebekelerinin sunumu ve hesaplamaları sırasında hataya sebebiyet vermemek için görsel anlamda bazı kurallara uyulması uygun olacaktır. Buna göre faaliyetler Şekil 2.6’da gösterildiği gibi soldan sağa doğru ve sıralama sayılarına uygun olarak dizilmelidir ve geriye dönmük ok kullanılmamalıdır.



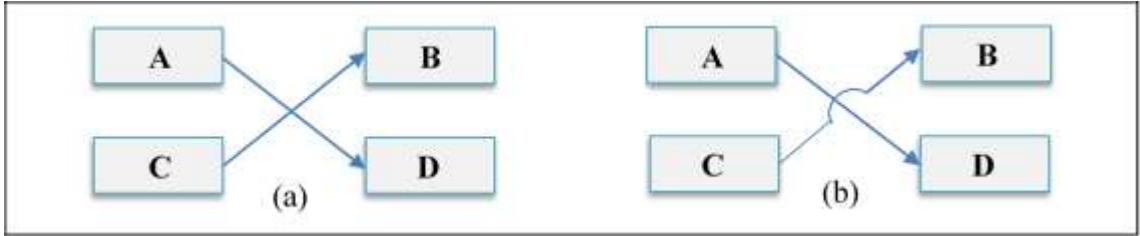
Şekil 2.6: Faaliyet ilişkilerinde hatalı (a) ve doğru (b) gösterimler.

Bir faaliyet, iki faaliyetin birden öncülü durumunda ise ilk faaliyetten çıkan oklar Şekil 2.7’de gösterildiği gibi birbirinden ayrılmalıdır.



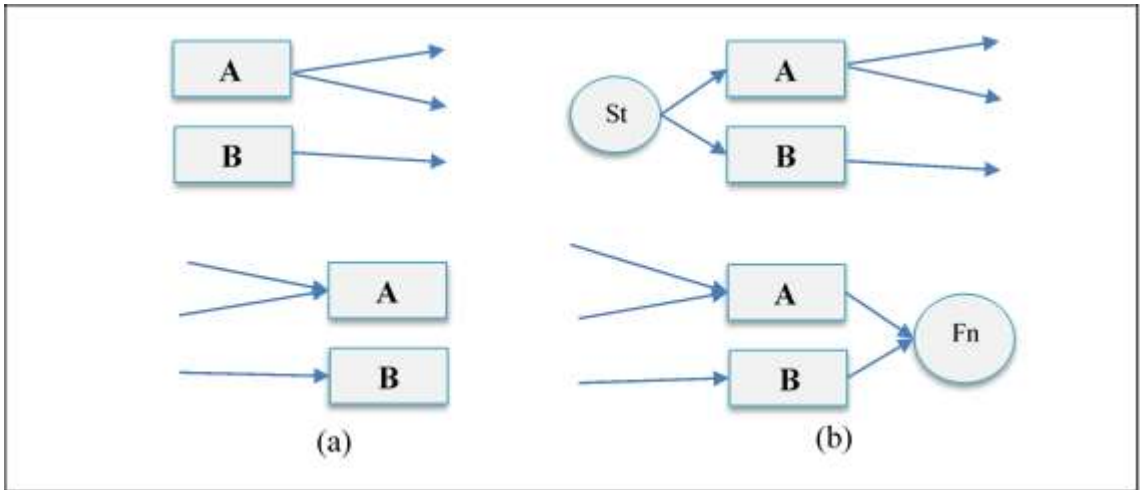
Şekil 2.7: Düğümde çıkan okların hatalı (a) ve doğru (b) gösterimi.

Oklar mümkün olduğunca kesişmemeli, kesişmek durumunda olan oklar Şekil 2.8'deki gibi gösterilmelidir.



Şekil 2.8: Kesişen okların hatalı (a) ve doğru (b) gösterimleri.

Birden fazla başlangıç veya bitiş faaliyetine sahip şebekelerde projenin tek bir faaliyetle başlanıp bitirilmesi prensibine uygun olarak, Şekil 2.9'da gösterildiği gibi sıfır süreli bir başlangıç ve bir bitiş faaliyeti eklenmelidir.



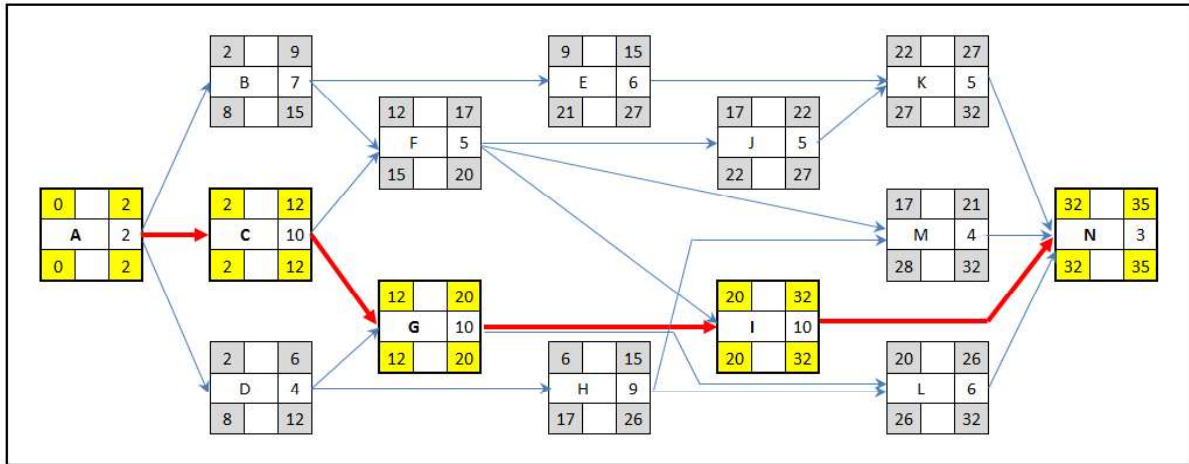
Şekil 2.9: Projenin başlangıç ve bitişinin hatalı (a) ve doğru (b) gösterimleri.

Tablo 2.2'de bir projeye ait faaliyet adı, faaliyet öncelileri ve faaliyetlere ait süre bilgileri verilmiştir.

Tablo 2.2: AoN tipi diyagram için proje verileri.

FAALİYET	HÖF	SÜRE	FAALİYET	HÖF	SÜRE
A	-	2	H	D	9
B	A	7	I	F, G	12
C	A	10	J	F	5
D	A	4	K	E, J	5
E	B	6	L	G, H	6
F	B, C	5	M	F, H	4
G	C, D	8	N	I, K, L, M	3

Tablo 2.2.'de verilen faaliyetler öncelikle sıralama sayısına göre eşit aralıklarla yerleştirilip, ardından faaliyetler arasındaki ilişkileri sembolize eden oklarla birleştirilmişlerdir. Her bir düğüm bir faaliyeti temsil etmektedir. Yapılan CPM hesaplamalarından sonra toplam bollukları sıfır çıkan faaliyetler kritik faaliyetler olarak belirlenmiş ve Şekil 2.10'da belirtilmiştir. A-C-G-I-N kritik yol, proje süresi de 35 gün olarak belirlenmiştir.

**Şekil 2.10:** Aktivitelerin düğümler üzerinde gösterimi.

Çizelgeleme, faaliyetlerin sıralamasının yanı sıra, başlangıç ve bitiş zamanlarının bilgisini de içerir. Birçok durumda çizelgenecek bir faaliyetin kaynak kısıtından dolayı kendisinden önceki faaliyetin bitmiş olması gerekliliği söz konusudur. Faaliyetler arasındaki ilişkileri saptamak için, projedeki her faaliyetle ilgili aşağıdaki sorular cevaplamalıdır (Elbeltagi, 2009):

- Mevcut faaliyet başlamadan önce hangi faaliyetlerin bitirilmesi gerekir?
- Mevcut faaliyet ile eşzamanlı olarak hangi faaliyetler oluşturulabilir?
- Mevcut faaliyetten hemen önce bitmesi gereken faaliyet nedir?

Öncelik ilişkilerinin (bağımlılıklar) üç temel tipi vardır:

Zorunlu bağımlılıklar (sabit mantık): Kolonlar yerleştirilmeden önce temelin atılmış olması gerekliliği.

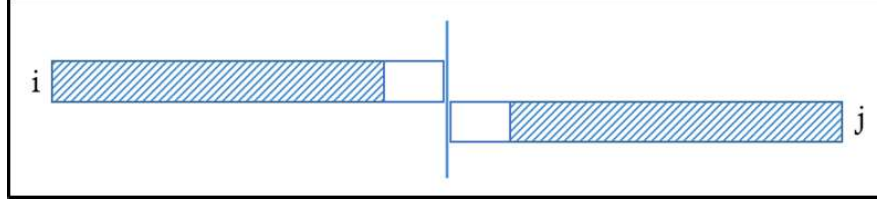
İsteğe bağlı bağımlılıklar (yumuşak mantık): Bunlar, proje yöneticisinin takdirine bağlı olan veya projeden projeye değişebilir bağımlılıklardır. Örneğin tedarik başlamadan önce tüm malzeme reçetesinin (Bill of Materials) belirlenmesi gerekmez.

Dış bağımlılıklar: Proje yöneticisinin kontrolü dışındaki bağımlılıklardır. Müteahhitlerden kaynaklanan ve kritik yolun dışına çıkmaya neden olan durumlar.

Öncüllük ilişkileri, faaliyetler arasındaki başlangıç-bitiş ilişkilerini, zaman bazlı olarak belirler. Genel kabul gereği, bir faaliyet kendinden önce gelen faaliyetlerin hepsi tamamlanmadan başlayamaz. Fakat uygulamada farklı durumlar da söz konusu olabilmektedir (Atlı , 2012). Örneğin, bir faaliyet tamamlanmamış olsa da bir miktar ara ürün oluşmuşsa, hemen sonraki faaliyet önceki faaliyetin yarısı tamamlandıktan sonra başlayabilir. Proje süresini kısaltmak için bazı faaliyetlerin paralel olarak çizelgelenmesi istenebilir. Bu tür durumlarda farklı öncüllük ilişkilerinden faydalanılabilir. Bu ilişkiler literatürde genelleştirilmiş öncüllük ilişkileri olarak tanımlanmıştır.

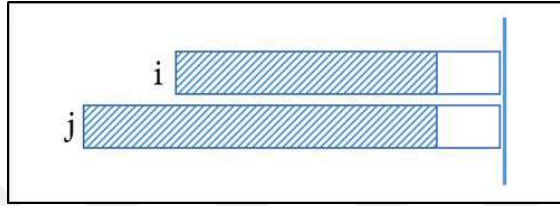
Öncüllük ilişkileri, faaliyetlerin birbirine bağımlı özelliklerinden veya teknolojik imkânlardan kaynaklanabilen kısıtlardır. Precedence diyagramlarında kullanılabilecek farklı öncüllük ilişkileri tanımlamaları mevcuttur:

Bitiş – Başlangıç (FS): Bir faaliyetin başlaması, Şekil 2.11’de gösterildiği gibi hemen önceki faaliyetin tamamlanmasına bağlıdır. Herhangi bir tanımlama yapılmıyorsa faaliyetler arası ilişki FS olarak varsayılır.



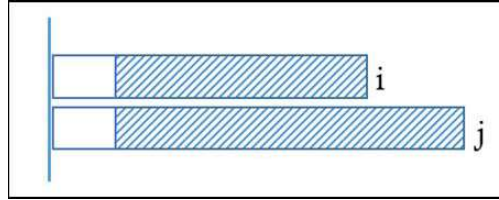
Şekil 2.11: Bitiş – Başlangıç tipi öncelik ilişkisi.

Bitiş – Bitiş (FF): Bir faaliyetin tamamlanması, Şekil 2.12’de gösterildiği gibi hemen önceki faaliyetlerin tamamlanmasına bağlıdır.



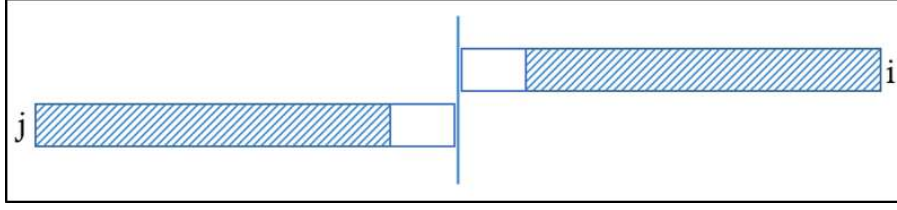
Şekil 2.12: Bitiş – Bitiş tipi öncelik ilişkisi

Başlangıç – Başlangıç (SS): Bir faaliyetin başlaması, Şekil 2.13’de gösterildiği gibi hemen önceki faaliyetlerin başlamasına bağlıdır.



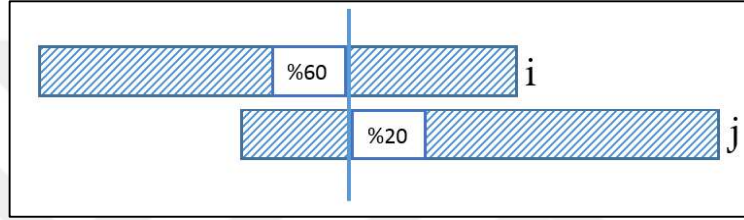
Şekil 2.13: Başlangıç – Başlangıç tipi öncelik ilişkisi.

Başlangıç – Bitiş (SF): Bir faaliyetin tamamlanması, Şekil 2.14’te gösterildiği gibi önceki faaliyetin başlamasına bağlıdır.



Şekil 2.14: Başlangıç – Bitiş tipi öncelik ilişkisi.

Tamamlanma Yüzdesi (Percent Complete): Şekil 2.15'te gösterildiği gibi bir faaliyetin belirli bir yüzdesinin tamamlanması ile diğer faaliyetin belirli bir yüzdesinin tamamlanmış olması gerekir.



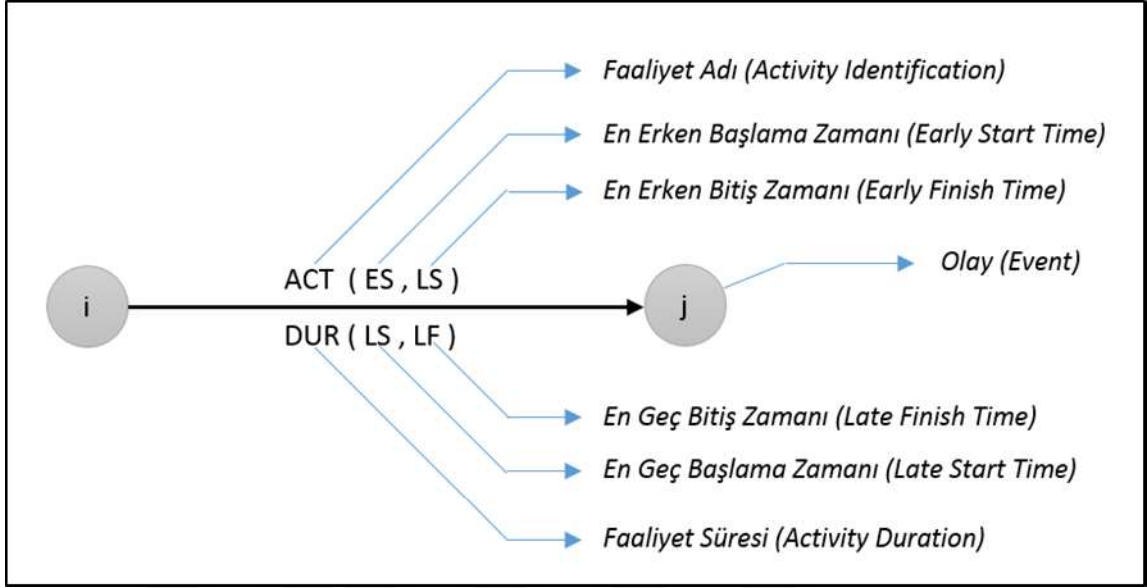
Şekil 2.15: Tamamlanma yüzdesinin belirtildiği öncelik ilişkisi.

2.3.2. Faaliyetlerin Oklar Üzerinde Gösterimi

Diğer bir öncelik kısıtı gösterimi faaliyetlerin oklar üzerinde gösterildiği AoA (Activity on Arrow) gösterimidir (Blazewics & Kobler , 2002). Bu yaklaşımda okun başlangıç ve bitişinde gösterilen düğümler olayları temsil eder. Her faaliyet bu yapılanma ile temsil edilirken düğümler ardışık olarak numaralandırılır. AoA yönteminin kısıtları nedeni ile ağ bütünlüğünü korumak adına kukla faaliyetler kullanılır. Yalnızca bitiş-başlangıç tipi öncelik ilişkisi kullanılır. Faaliyet ilişkilerinin tam olarak ifade edilememesi, kukla faaliyetlere ihtiyaç duyması ve dolayısıyla bilgisayar programlama tekniğine yatkın olmaması nedeniyle kullanım alanı kısıtlıdır.

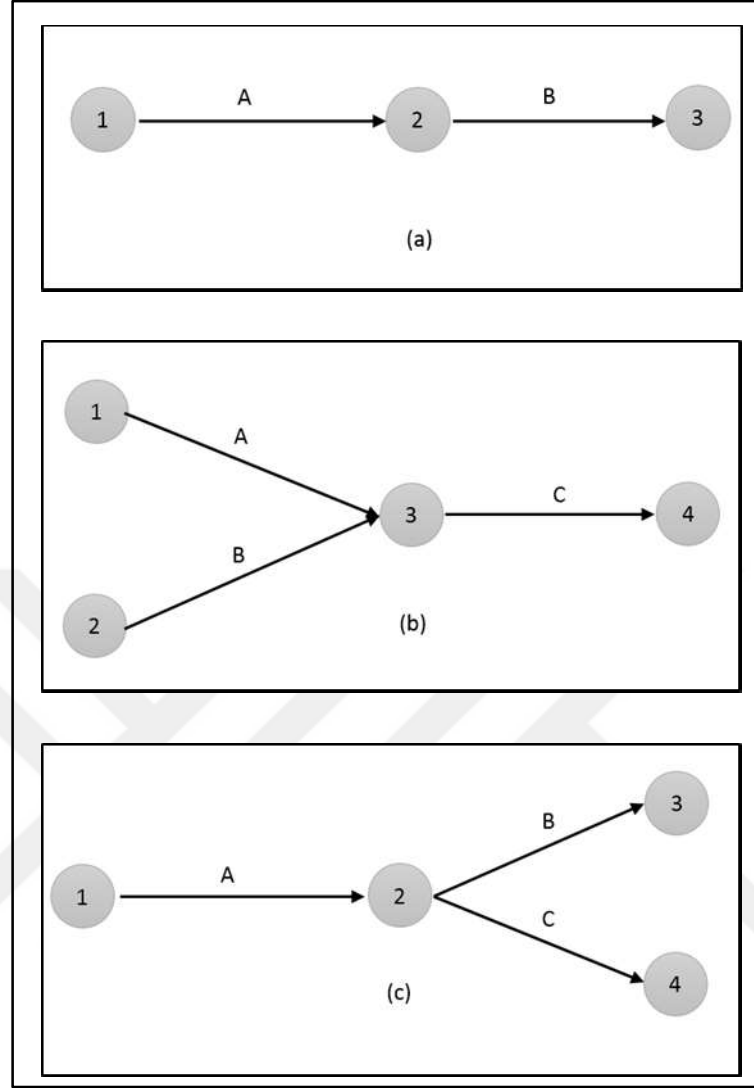
CPM ve PERT uygulamak için, projeyi oluşturan faaliyetlerin bir listesine ihtiyacımız vardır. Proje, tüm faaliyetler tamamlandığında tamamlanmış kabul edilir. Her faaliyet için, faaliyet başlamadan önce tamamlanması gereken bir dizi faaliyet (öncüller) vardır. Faaliyetler arasındaki öncelik ilişkilerini temsil etmek için bir proje ağı kullanılır. Faaliyetler yönlendirilmiş oklarla temsil edilir ve düğümler, bir dizi faaliyetin tamamlanmasını temsil etmek için kullanılır. Bu nedenle, genellikle proje ağındaki düğümler olaylar olarak adlandırılır. Bu tür proje ağına AoA (Activity on Arrow) ağı denir.

Tek bir faaliyetin AoA diyagramı üzerinde gösterimi Şekil 2.16'da belirtilmiştir:



Şekil 2.16: Aktivitelerin oklar üzerinde gösterimi.

AoA ağının öncelik ilişkilerini nasıl temsil ettiğini anlamak için, A etkinliğinin etkinlik B'nin bir öncülü olduğunu varsayalım. Bir AoA ağındaki her düğüm bir veya daha fazla etkinliğin tamamlandığını gösterir. Böylece, Şekil 2.17 (a)'daki düğüm 2, aktivite A'nın tamamlanmasını ve faaliyet B'nin başlangıcını temsil eder. Aktivite C başlamadan önce A ve B faaliyetlerinin tamamlanması gerektiğini varsayalım. Şekil 2.17 (b)'de, düğüm 3, aktiviteler A ve B'nin tamamlandığı olayı temsil etmektedir. Şekil 2.17 (c), A aktivitesini hem B hem de C faaliyetlerinin bir öncülü olarak göstermektedir.

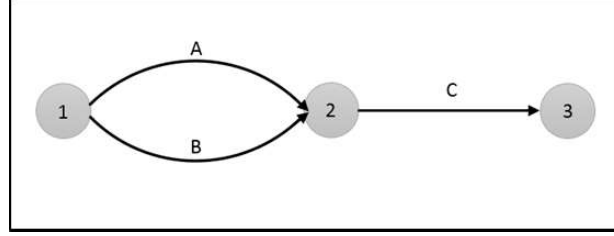


Şekil 2.17: Üç faaliyet arasındaki ilişkilerin AoA üzerinde gösterimi.

Bir projenin AoA tipi ağ diyagramı aşağıdaki kurallara göre oluşturulabilir:

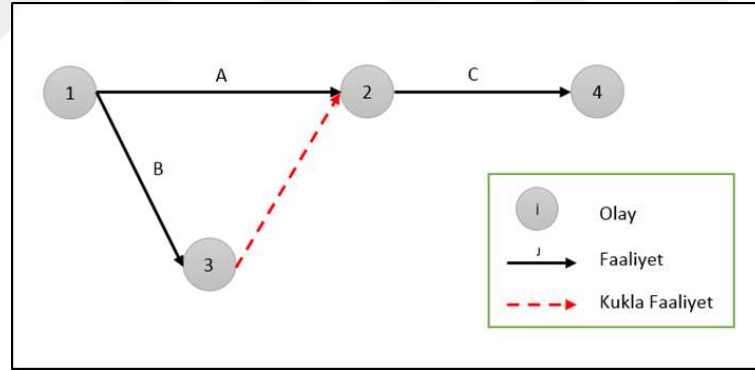
- Düğüm 1, projenin başlangıcını temsil eder. Önceliği olmayan her faaliyeti temsil etmek için düğüm 1'den bir ok çıkarılmalıdır.
- Projenin tamamlandığını gösteren bir düğüm (bitiş düğümü) ağa dâhil edilmelidir.
- Bir aktivitenin tamamlandığını gösteren düğümün daima bir etkinliğin başlangıcını temsil eden düğümden daha büyük bir sayıya sahip olması için ağdaki düğümler numaralandırılır. (Birden fazla numaralandırma düzeni olabilir).
- Bir faaliyet, şebekede birden fazla ok tarafından temsil edilmemelidir.
- İki düğüm en fazla bir ok ile birbirine bağlanabilir. Bu kuralı sağlamak için bazen zaman ve kaynak tüketmeyen bir sanal faaliyet eklenmesi gerekebilir.

Örneğin, A ve B faaliyetlerinin ikisinin de C etkinliğinden önce aynı anda başlayabileceğini varsayalım. İki düğümün en fazla bir ok ile bağlanabileceği kuralı olmasaydı şekil 2.18'deki durumla karşılaşılacaktır.



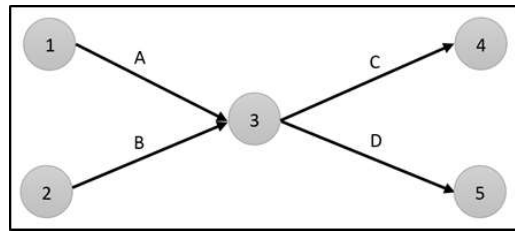
Şekil 2.18: İki düğümün birden fazla okla birleştirilmesi hatası.

Bu gibi hatalı durumlardan kaçınmak için kukla etkinlikler kullanmadan ağ bağımlılıklarını tanımlamak olanaksızdır. Kukla (sahte) faaliyetler, kesikli çizgi ile gösterilen suni faaliyetlerdir ve kaynakları tüketmez veya zaman gerektirmez, mantık hatasını gidermek veya numaralama kuralını gerçekleştirmek için ağa eklenir. Şekil 2.19'da numaralama kukla faaliyetinin (Numbering Dummy) şebeke üzerinde gösterimi verilmiştir.



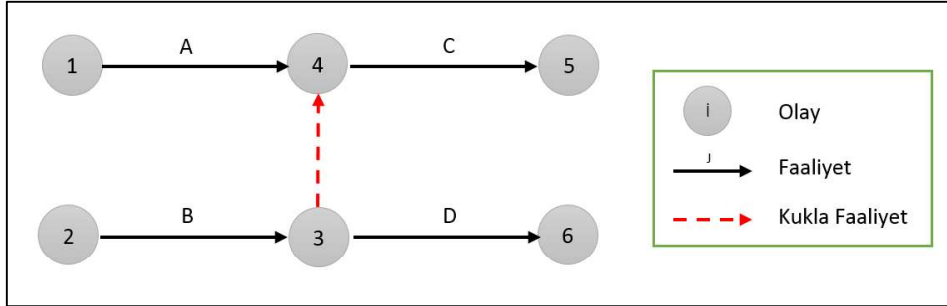
Şekil 2.19: Kukla faaliyetin serim üzerinde gösterilmesi.

C faaliyetinin öncülleri A ve B faaliyetleri ve D faaliyetinin öncülü sadece B faaliyeti ise Şekil 2.20'deki gösterim hatalı gösterim olacaktır.



Şekil 2.20: Kukla faaliyet kullanmadan yapılan hatalı gösterim.

Bu gösterimin doğru ifadesi Şekil 2.21’de gösterilmiştir. Burada kukla faaliyet (logic dummy) olarak karşımıza çıkar ve mantık hatasını gidermek üzere kullanılmıştır.



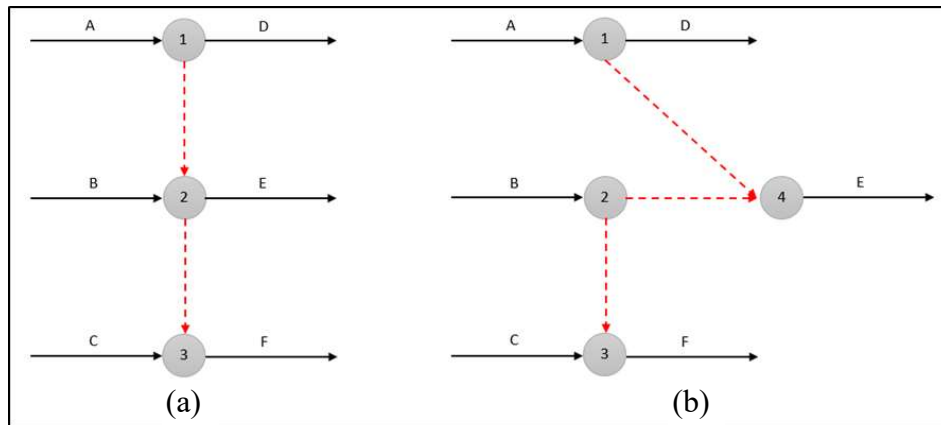
Şekil 2.21: Lojik kukla faaliyetin eklenmesi ile doğru gösterim.

Tablo 2.3’de verilen öncelik ilişkileri tablosunun, kukla faaliyet kullanımına ilişkin hatalı ve doğru gösterimleri Şekil 2.22’de gösterilmiştir.

Tablo 2.3: Örnek öncelik ilişkileri tablosu.

FAALİYET	HEMEN ÖNCEKİ FAALİYET
A	-
B	-
C	-
D	A
E	A, B
F	B, C

Şekil 2.22’de A, F’nin önceliği olmadığından hatalı gösterilmiştir. Bunun yerine şebekeye bir kukla faaliyet daha eklenerek doğru gösterim sağlanabilmektedir.



Şekil 2.22: Kukla faaliyetin hatalı (a) ve doğru (b) kullanımı.

2.3.3. Program Değerlendirme ve Gözden Geçirme Tekniği (PERT)

PERT, temel olarak belirli bir projeyi tamamlama ile ilgili görevleri, özellikle her görevi tamamlamak için gereken süreyi analiz etmek ve toplam projeyi tamamlamak için gereken minimum zamanı belirlemek için kullanılan bir yöntemdir.

Teknik, büyük ve karmaşık projelerin planlama ve zamanlama aşamalarını basitleştirmek amacıyla 1950'lerde geliştirilmiştir. Tüm faaliyetlerin detaylarını ve sürelerini tam olarak bilmeden bir projenin çizelgelenmesini mümkün kılarak belirsizliği gidermeyi başarmıştır. Başlangıç veya tamamlanma odaklı olmaktan ziyade olay odaklı bir tekniktir ve zamanın maliyetten daha önemli bir faktör olduğu projelerde kullanılır.

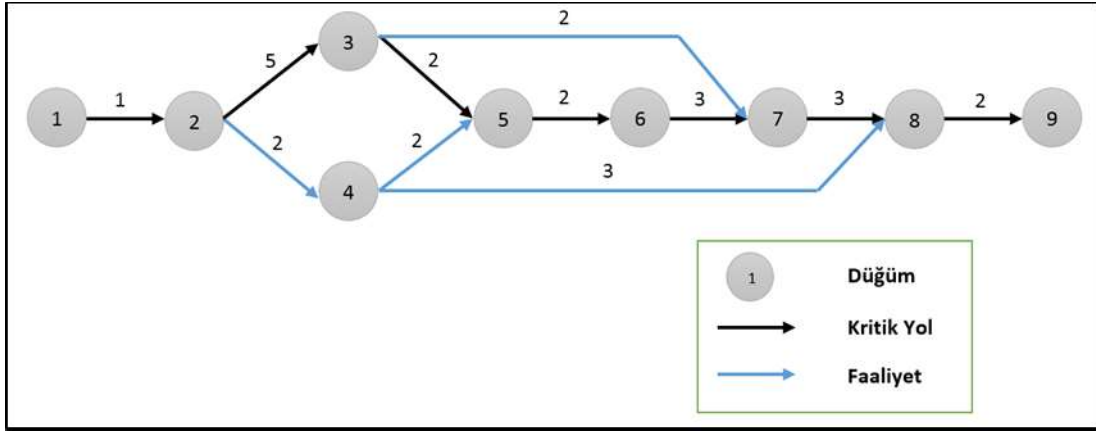
PERT grafiğini oluşturmanın amaçlarından biri, projeyi tamamlamak için ne kadar zaman gerektiğini belirlemektir. Bu nedenle, PERT, projenin başarısını doğrudan etkileyen öğeleri, yani zamanı, maliyeti ve performansı analiz etmek için zamanı ortak bir payda olarak kullanır.

PERT grafiği, karar vermeyi kolaylaştıran bir araçtır; PERT grafiğinin ilk taslağı, daha sonra farklı faaliyetlerin eklenmesine izin vermek için ardışık sayılarla numaralandırılır. PERT grafiğindeki iki ardışık faaliyet, genellikle ok tipi diyagramlarla gösterilir. Olaylar mantıksal bir sırayla sunulur ve bir önceki olay tamamlanuncaya dek hiçbir etkinlik başlamaz. Planlayıcı hangi kilometre taşlarının PERT olayları olduğuna ve aynı zamanda doğru sıralamaya karar verir.

Tablo 2.4: 11 aktiviteli bir projenin faaliyet listesi.

OLAY	FAALİYET	HÖF	SÜRE
1-2	A	-	1
2-3	B	A	5
2-4	C	A	2
3-5	D	B	2
3-7	E	B	2
4-5	F	C	2
4-8	G	C	3
5-6	H	D,F	2
6-7	I	H	3
7-8	J	E,I	3
8-9	K	G,J	2

Tablo 2.4'te verilen faaliyet listesine ait CPM/PERT şebekesi Şekil 2.23'te verilmiştir.



Şekil 2.23: CPM/PERT diyagramlarının standart gösterimi.

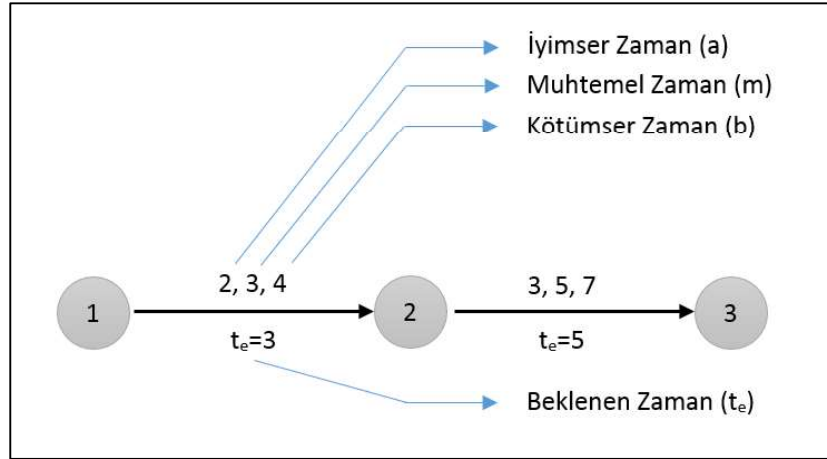
PERT ve CPM teknikleri arasında gösterim yönünden bir fark yoktur ve iki teknikte de genellikle AoN tipi diyagram kullanılır. CPM ve PERT arasındaki temel farklar aşağıdaki gibi sıralanabilir:

- PERT beklenen zamanı türetmek için, iyimser, büyük olasılıkla ve kötümser olmak üzere üç tahmin kullanır. CPM, normal zamanı temsil edecek daha doğru bir tahmin gerektirir.

İyimser tamamlanma zamanı (optimistic): Bu zaman her şeyin plana göre ve minimum güçlüklerle gerçekleşeceğini varsayar. Bu varsayım, zamanın yaklaşık yüzde 1'inde gerçekleşebilecek bir olasılıktır.

Karamsar tamamlanma zamanı (pessimistic): Bu zaman her şeyin plana göre gitmeyeceğini ve maksimum güçlüklerin oluşacağını varsayar. Bu varsayım zamanın yaklaşık yüzde 1'inde gerçekleşebilecek bir olasılıktır.

Büyük olasılıkla tamamlanma zamanı (most likely): Fonksiyonel yöneticinin zihninde, bu çabanın tekrar tekrar gerçekleştirilmesi durumunda en sık rastlanan zamandır. Şekil 2.24'te denklem 2.1 kullanılarak hesaplanan beklenen zaman değerleri görülmektedir.



Şekil 2.24: Kritik yol için beklenen zaman analizi.

- PERT, her etkinlik zamanı için bir beta dağılımına ve beklenen süre için normal bir dağılıma dayanır, olasılıktır. Bu, bir projenin tamamlanabilme yüzdelerinin hesaplanmasını sağlar. CPM tek bir zaman tahminine dayalıdır, deterministtir.
- PERT ve CPM, şebeke mantığını bozmamak için kukla faaliyetlerin kullanılmasına izin verir.
- PERT, sürelerinin hesaplanmasındaki risklerin yüksek değişkenliğe sahip olduğu Ar-Ge projeleri için kullanılır. CPM, kaynağa bağlı ve doğru zaman tahminlerine dayanan inşaat projeleri için kullanılır.

PERT Tekniğinde olaylar arasındaki beklenen zaman Denklem 2.1 ile ifade edilir. İlk varsayım, standart sapmanın zaman gereksinimi aralığının altıda biri olmasıdır. Bu varsayım olasılık teorisinden kaynaklanır, burada eğrinin uç noktaları ortalamadan üç standart sapma olarak bulunur. İkinci varsayım, bir etkinlik için gerekli süre dağılımının bir beta dağılımı olarak ifade edilebilmesini gerektirir (Kerzner, 2009).

$$t_e = \frac{a + 4m + b}{6} \quad (2.1)$$

Burada:

- **t_e:** Beklenen süre
- **a:** En iyimser zaman
- **b:** En kötümser zaman
- **m:** Muhtemel zamanı ifade eder.

2.3.4. Kritik Yol Metodu (CPM)

Kritik yol metodu (CPM), proje yönetimi için geliştirilmiş temel bir kantitatif yöntemdir. CPM deterministik faaliyet sürelerini varsayarak, projeyi tamamlamak için gereken minimum süreyi belirler (Ghods, et al., 2009).

1950'lerin sonlarından bu yana, CPM teknikleri, büyük projelerin planlanması ve çizelgelenmesi için yaygın olarak kabul görmüş bir araçtır (Azaron, Perkgoz, & Sakawa, 2005). İnşaat sektörü, projelerin planlanması ve kontrolü, planların iletilmesi ve yeni yöneticilerin eğitimi gibi bazı alanlarda bu metodun kullanımından faydalanmaktadır. (Lacouture, et al., 2009).

Genel olarak projelerin çizelgelenmesi, sürelerin tespiti, yani projeyi onaylamak için gerekli olan işlerin yürütüleceği tarihlerin oluşturulması ile ilgilidir. En yaygın kullanılan planlama yaklaşımları, yukarıda açıklandığı gibi projenin işleri arasındaki ilişkileri grafiksel olarak gösteren bir ağ oluşturulmasına dayanmaktadır. Projeyi temsil eden şebeke üzerinde, ileri ve geri hesap prosedürleri uygulanarak proje süresi hesaplanır. Bu yolla faaliyetlerin en erken ve en geç başlama ve bitiş süreleri belirlenmiş olur. Ayrıca her bir faaliyetle ilgili bolluklar da hesaplanarak bir faaliyetin proje süresini uzatmayacak şekilde ne kadar ertelenebileceği ortaya çıkmış olur. Bolluğu olmayan faaliyetler ise kritik olarak ele alınır ve bu faaliyetlere proje yöneticisi tarafından daha fazla dikkat edilmesi gerekmektedir (Klein, 2000).

İnşaat faaliyetlerinin planlanması ve çizelgelenmesi hayati öneme sahiptir çünkü kaynak ve malzeme gereksinimleri proje başlamadan önce bilinmelidir. (Bettemir Ö. H., 2009)

Şebeke diyagramının oluşturulması aşamasında olaylarla alakalı bir takım kabuller göz önünde bulundurulmaktadır (Winston, 2004):

- İki olay direkt olarak en fazla bir faaliyet ile bağlanabilir.
- Her olay numarası en fazla bir defa kullanılmalıdır.
- Bir şebeke diyagramı sadece bir başlangıç ve bir sonuç olayına sahip olabilir.

Kritik yol metodu aktivitelerin oklar ve düğümler üzerinde gösterildiği şebekelerde kullanılabilir. Stevens (1990) ve Mubarak (2010) 'a göre yapılmış tanımlamalar aşağıda verilmiştir:

- **İleri Hesap (Forward Pass):** Bir ağ üzerinde baştan sona gezinme ve en erken başlama ve en erken bitiş zamanları ile proje bitiş süresinin hesaplanması sürecidir.
- **Geri Hesap (Backward Pass):** En geç başlama ve en geç bitiş zamanını hesaplama sürecidir.
- **En Erken Başlama Zamanı (ES):** Bir aktivitenin başlamasının mümkün olduğu en erken zamandır.
- **En Erken Bitiş Zamanı (EF):** Bir aktivitenin tamamlanabilmesinin mümkün olduğu en erken zamandır.
- **En Geç Başlama Zamanı (LS):** Bir aktivitenin projenin tamamlanma zamanının aksatmadan başlayabilmesinin mümkün olduğu en geç zamandır.
- **En Geç Bitiş Zamanı (LF):** Bir aktivitenin projenin tamamlanma zamanının aksatmadan tamamlanabilmesinin mümkün olduğu en geç zamandır.
- **Toplam Bolluk (Total Float):** Bir aktivitenin, proje süresini geciktirmeyecek şekilde ertelenebileceği maksimum süredir.
- **Serbest Bolluk (Free Float):** Bir faaliyetin kendisinden sonra gelen aktivitenin en erken başlama zamanını geciktirmeyecek şekilde ertelenebileceği maksimum süredir.
- **Kritik Aktivite:** Kritik bir etkinliğin başlangıcında veya bitişinde herhangi bir gecikme, projenin tamamında gecikmeye neden olur.
- **Kritik Yol:** Bir ağdaki gecikmeler ve kısıtlamalar dâhil baştan sona en uzun yoldur.

Kısaca CPM hesaplamaları Tablo 2.5’te gösterildiği gibi özetlenebilir.

Tablo 2.5: CPM hesaplamaları özet tablosu.

HESAPLANMAK İSTENEN	HESAP ADIMLARI
ES	Hemen önceki faaliyetlerinin en büyük EF değeri
EF	ES+Faaliyet Süresi
LS	LF-Faaliyet Süresi
LF	Hemen sonraki faaliyetlerin en küçük LS değeri
FF	(Hemen sonraki faaliyetlerin en küçük ES değeri)-EF
TF	LS-ES

Tablo 2.6’da faaliyet, öncelik ilişkileri ve süreleri verilen örnek bir projenin CPM hesaplamaları AoA ve AoN tipi şebekelerde adım adım gösterilmiştir (Stevens, 1990).

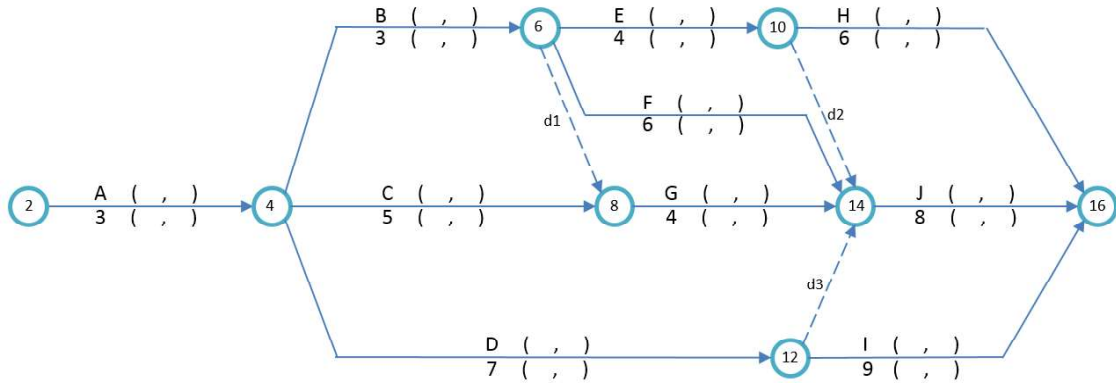
Tablo 2.6: Örnek CPM şebekesi verileri.

FAALİYET	HÖF	SÜRE
A	-	3
B	A	3
C	A	5
D	A	7
E	B	4
F	B	6
G	B, C	4
H	E	6
I	D	9
J	D, E, F, G	8

Tablo 2.6'daki proje verileri göz önüne alınarak Şekil 2.25'te belirtilen şebeke çizilmiştir. Görüldüğü üzere öncülü olmayan A faaliyeti başlangıç faaliyeti olurken ardılı olmayan H, I ve J faaliyetleri bitiş faaliyetleri olarak karşımıza çıkar. Şebeke çizimlerinde öncelikle başlangıç ve bitiş faaliyetlerinin çizilmesi ok tipi diyagramların çizimini kolaylaştıran genel bir kuraldır.

C ve B faaliyetleri aynı olay zamanında başladığından ve aynı zamanda bu iki faaliyet G faaliyetini öncülü olduğundan 4 ve 8 olayları arasında iki faaliyet birden çizmek ancak d1 kukla faaliyet vasıtası ile gerçekleştirilebilir. Burada, kukla faaliyet numaralama kukla faaliyeti olarak kullanılmıştır. J faaliyetinden önce kullanılan d2 ve d3 kukla faaliyetleri ise mantıksal öncelik ilişkisini tanımlamak üzere kullanılmıştır ve lojik kukla faaliyet adını alır.

Olayların isimlendirilmesi yapılırken ardışık sayıların kullanılması uygun olur. Projenin ilerleyen aşamalarında faaliyet eklenmesi olasılığı olduğundan, proje büyüklüğüne göre belirlenecek sayıda (2, 5 gibi) artan ardışık sayılar kullanılabilir. Şebeke faaliyetleri daha önce Şekil 2.16'da belirtilen standart forma göre çizilmiştir.

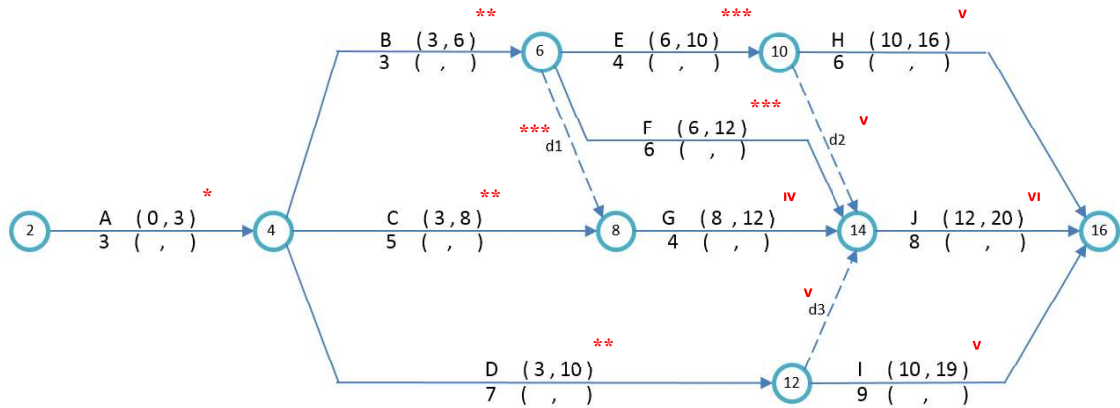
**Şekil 2.25:** Faaliyet ve günlerin belirtildiği CPM şebeke ağı.

İleri hesap (Forward Pass)

Projenin 0 zamanında başladığını varsayarak tüm başlangıç faaliyetlerinin ES değerini 0 olarak başlatabiliriz. Devamında ise projedeki tüm faaliyetlerin ES ve EF değerleri bulunacaktır.

- * A faaliyeti 0. gün sonunda başlayıp 3. gün sonunda bitecektir. $EF=ES+Dur=0+3=3$
- ** B, C ve D, 3. gün sonunda başlayıp 6, 8 ve 10. gün sonunda bitecektir.
- *** E, F ve d1 6. gün sonunda başlayıp, 10, 12 ve 6. gün sonunda bitecektir. d1 faaliyeti kukla faaliyet olduğundan zaman ve kaynak tüketmez.
- ^{iv} G faaliyeti, B ve C faaliyeti bittikten sonra başlayacağından, iki faaliyet de tamamlanmadan başlayamaz. O halde önceki faaliyetlerden $Max(EF)$ değeri, G faaliyetinin ES değeri olur.
- ^v I, d3, H, d2 faaliyetleri 10. günde başlayıp sırasıyla 10, 19, 16 ve 10. günde bitmektedir.
- ^{vi} J faaliyeti d2, F, G, d3 faaliyetlerinden sonra başlamaktadır. Önceki faaliyetlerden $Max(EF)$ değeri J_{ES} değeridir. $J_{ES}=12$ ve $J_{EF}=12+8=20$.

Son J faaliyetinin bitebileceği en erken süre 20 gündür ve bu nedenle projenin tamamlanması için gereken en kısa süre 20 gündür. Şekil 2.26'da ileri hesap sonuçları şebeke üzerinde gösterilmiştir.



Şekil 2.26: CPM şebekesi ileri hesap.

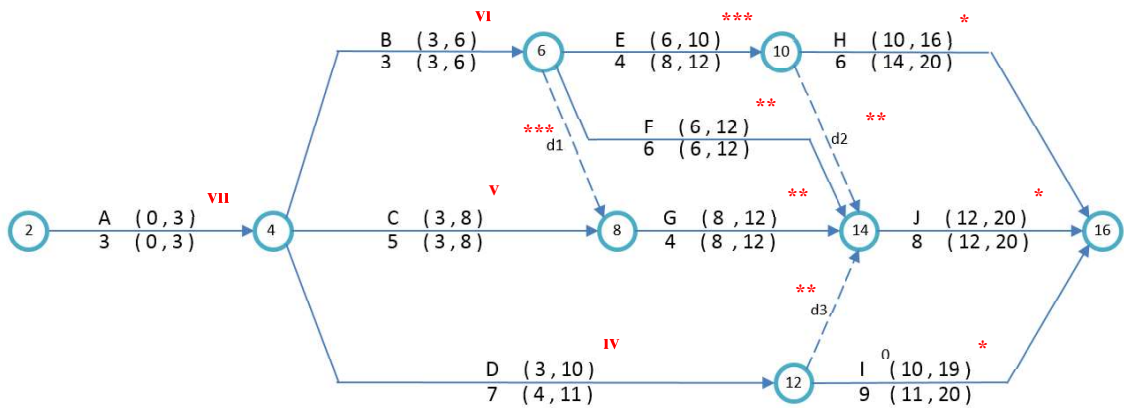
Geri Hesap (Backward Pass)

- * Projenin son faaliyetinin EF değeri 20 olduğundan, H, I ve J faaliyetlerinin en geç bitiş süreleri de proje bitiş süresine denk olmalıdır. $LF_H=20$, $LF_I=20$, ve $LF_J=20$ 'dir. Buna

bağlı olarak bu faaliyetlerin En geç başlama zamanları ($LS=LF-DUR$) sırasıyla 14, 12 ve 11 gün olarak bulunur.

- ****** d2, F, G ve d3 faaliyetlerinden sonra J faaliyeti geldiğinden bu faaliyetlerin en geç bitiş LF değerleri 12 olur. Buna göre bu faaliyetlerin LS değerleri sırasıyla 12, 6, 8, 12 gün olarak hesaplanır.
- ******* E faaliyetinden hemen sonraki faaliyetlerin (H ve d2) $Min(LS)$ değeri E'nin LF değeridir ve buna bağlı olarak $LS_E=12-DUR_E=8$ olarak hesaplanır.
- **IV** D faaliyetinden hemen sonraki faaliyetlerin (d3 ve I) $Min(LS)$ değeri I'nın LF değeridir ve buna bağlı olarak $LS_D=11-DUR_D=4$ olarak hesaplanır.
- **V** C ve d1, en geç 8. günde bitmelidir, bu nedenle en geç 3 ve 8. günde tamamlanmalıdır.
- **VI** B faaliyeti en geç E, F ve d1 faaliyetlerinden önce 6. günde bitmelidir ve buna bağlı olarak en geç 3. günde başlayabilir.
- **VII** A faaliyeti en geç B, C ve D faaliyetlerinden önce 3. günde bitmelidir ve buna bağlı olarak en geç 0. günde başlayabilir.

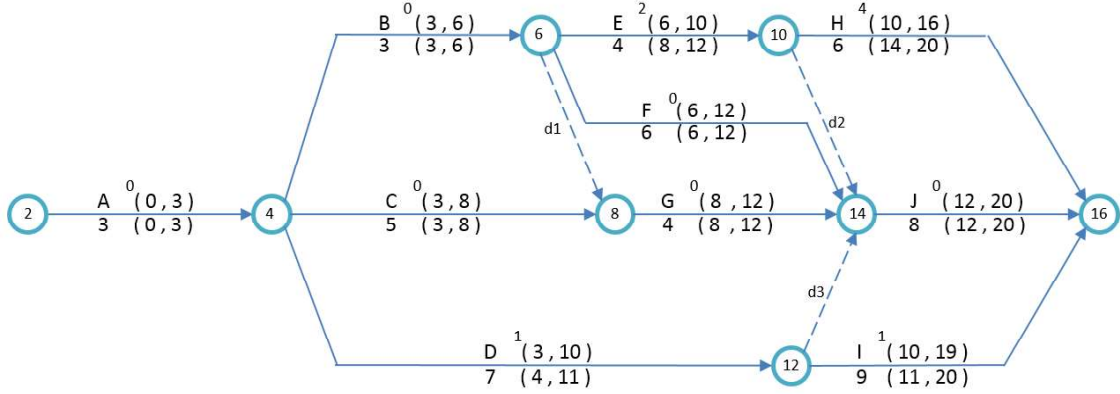
Şekil 2.27'de geri hesap sonucu gösterilmektedir. Geri hesap sonucunda A başlangıç faaliyetinin En erken ve en geç başlama zamanı sıfır olarak hesaplandığından çözümün doğruluğu ispatlanmaktadır.



Şekil 2.27: CPM şebekesi geri hesap.

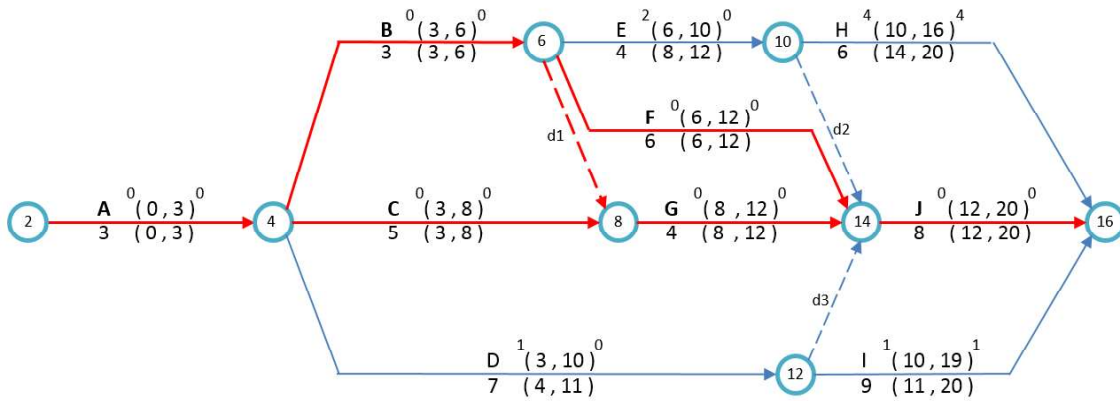
ES, EF, LS, LF değerleri hesaplandıktan sonra faaliyetlerin toplam bolluk hesabı yapılabilir. Toplam bolluk bir faaliyetin toplam proje süresini uzatmadan maksimum ertelenebileceği süredir. Her bir faaliyet için $TF_i=LS_i-ES_i$ veya $TF_i=LF_i-EF_i$ formülizasyonu ile toplam bolluk hesabı yapılabilir.

Tüm faaliyetlerin toplam bollukları bulunduğundan sonra projenin kritik yolu/yolları belirlenmiştir. Şekil 2.28’de faaliyetlerin toplam bolluk hesabı verilmiştir.



Şekil 2.28: Faaliyetlerin toplam bolluk hesabı.

Serbest bolluk değeri bir faaliyetin kendisinden sonra gelen faaliyetleri geciktirmeden maksimum ertelenebileceği süredir. Bir faaliyetin serbest bolluk hesabı için kendisinden sonra gelen aktivitelerin $\text{Min}_{(ES)}$ değerinden, faaliyetin EF değerinin çıkarılması ile bulunur. Şekil 2.29’da hem serbest bolluk değerleri hem de kritik yollar gösterilmiştir.



Şekil 2.29: CPM şebekesinin serbest bolluk ve kritik yolları.

Hesapların doğruluğunu kontrol etmek için aşağıdaki prosedür izlenebilir:

- Başlangıç ve bitiş faaliyetleri kritik yol üzerinde bulunmalıdır.
- Tüm kritik yollar başlangıçtan bitişe kesiksiz olmalıdır.
- Tüm aktiviteler için $TF \geq FF$ olmalıdır.
- $TF=0$ olan tüm aktiviteler kritik yol üzerinde olmalıdır.

Yapılan hesaplar sonucunda proje süresi 20 gün ve **A, B, C, d1, F, G, J** faaliyetleri kritik faaliyetler olarak bulunmuştur. Projede (**A-B-d1-G-J**), (**A-C-G-J**) ve (**A-B-F-J**) olmak üzere 3 adet kritik yol mevcuttur.

Hesaplanan tüm değerler Tablo 2.7’de sunulmuştur.

Tablo 2.7: CPM Şebekesinin sonuç değerleri.

FAALİYET	HÖF	SÜRE	ES	LS	EF	LF	TF	FF
A	-	3	0	3	0	3	0	0
B	A	3	3	6	3	6	0	0
C	A	5	3	8	3	8	0	0
D	A	7	3	10	4	11	1	0
E	B	4	6	10	8	12	2	0
F	B	6	6	12	6	12	0	0
G	B, C	4	8	12	8	12	0	0
H	E	6	10	16	14	20	4	4
I	D	9	10	19	11	20	1	1
J	D, E, F, G	8	12	20	12	20	0	0

AoN tipi diyagramların kritik yol hesabı, AoA tipi diyagramlarla aynıdır. Aralarındaki temel fark aktivitelerin gösterim biçimlerinden ileri gelir. Tablo 2.6’da verilen örnek projenin PDM hesabı için öncelikle düğüm noktalarının nereye çizilmesi gerektiğini belirlemek adına sıralama sayılarının (sequence step) hesabı Tablo 2.8’de gösterilmiştir.

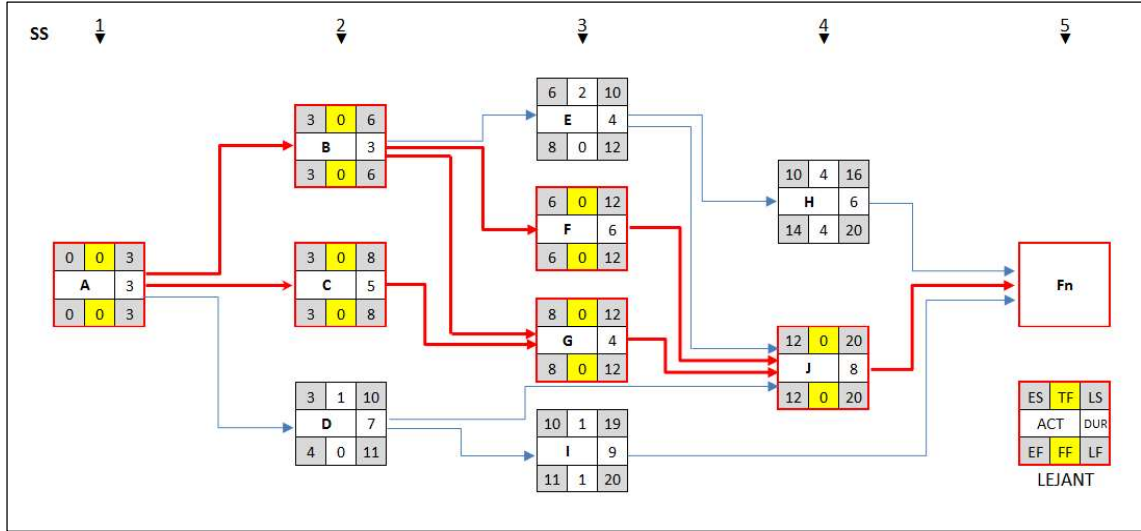
Tüm faaliyetler kendisinden önce gelen aktiviteden 1 sıra sonra başlamaktadır.

Tablo 2.8: Sıralama sayısı hesabı.

HÖF	FAALİYET	SS	SS
-	A	1	1
A	B	$SS_A+1=2$	2
A	C	$SS_A+1=2$	2
A	D	$SS_A+1=2$	2
B	E	$SS_B+1=3$	3
B	F	$SS_B+1=3$	3
B, C	G	$\text{Max} [SS_B+1=3 ; SS_C+1=3]$	3
E	H	$SS_E+1=4$	4
D	I	$SS_D+1=3$	3
D, E, F, G	J	$\text{Max} [SS_D+1=3 ; SS_E+1=4 ; SS_F+1=4 ; SS_G+1=4]$	4
H, J, I	Fn		5

Sıralama sayısına göre dizilen faaliyet düğümleri arasında öncelik ilişkilerine göre tanımlı oklar çizilmiş ve CPM ileri ve geri hesap adımları uygulanmıştır. Fn adlı süre ve kaynak gerektirmeyen aktivite de şebekeye eklenmiştir. Proje süresi CPM şebekesinde olduğu gibi 20 gün ve kritik yollar da, A-B-F-J | A-B-G-J | A-C-G-J olmak üzere 3 adettir. Tablo 2.7’de belirtilen değerlerin hepsinin PDM şebekesi için de geçerli olduğu aşikârdır.

Tablo 2.6’da verilen CPM şebekesinin öncelik diyagramı metodu (PDM) ile çizimi Şekil 2.30’da verilmiştir.



Şekil 2.30: PDM şebekesi.

2.3.5. Matris Yöntem

Ok veya kutu tipi şebeke çizelgeleme metotlarının yanı sıra matris metotta diyagramların çizilmesine gerek yoktur. Bütün CPM hesaplamaları oluşturulan bir matris üzerinden yapılır. Bu metotta hesaplamalara başlayabilmek için bir başlangıç ve bir bitiş aktivitesine ihtiyaç vardır. PDM metodunda olduğu gibi eğer birden çok başlangıç veya bitiş faaliyeti varsa St ve Fn kukla faaliyetleri şebekeye eklenir.

Matris metodun kritik yolun bulunmasında bazı ara işlemlerin yapılmasına gerek kalmaması ve şebekenin bilgisayar ortamına uyarlanması gibi avantajları vardır (Manisalı ve diğ., 2000).

Hesap Adımları

- Matrisin ilk satırına aktiviteler ve ikinci satıra faaliyet süreleri, sıralama sayılarına göre düzenli olarak sıralanır. Üçüncü satır daha sonra ES değerlerinin yazılması için boş bırakılır.
- En sağ sütuna sıralama sayısına göre sıralanmış aktiviteler yazılır ve bir önceki sütun sonradan EF değerlerini yazmak üzere boş bırakılır.

- Aktivite satır ve sütunları birleştirilerek Tablo 2.10'da belirtildiği üzere diyagonal hücreler ikiye ayrılarak TF ve FF değerlerini yazmak üzere boş bırakılır. Bu yolla başlangıç (setup) matrisi oluşturulmuş olur.

Tablo 2.9'da gösterilen proje verilerine göre matris metotla çözümlene yapılmış ve hesap adımları çözümün içinde belirtilmiştir.

Tablo 2.9: Matris metot uygulaması için proje verileri.

FAALİYET	SÜRE	HÖF	SS
A	3	-	1
B	3	A	2
C	5	A	2
D	7	A	2
E	4	B	3
F	6	B	3
G	4	B, C	3
I	9	D	3
H	6	E	4
J	8	D, E, F, G	4
Fn	0	H, J, I	5

Hesap adımlarına göre oluşturulan başlangıç matrisi Tablo 2.10'da verilmiştir.

Tablo 2.10: Matris metot başlangıç (setup) matrisi.

ACT	A	B	C	D	E	F	G	I	H	J	Fn	EF	ACT
DUR	3	3	3	7	4	6	4	9	6	8	0		
ES													
													A
													B
													C
													D
													E
													F
													G
													I
													H
													J
													Fn

- Birbiri ile ilişkili aktivitelerin kesiştiği her kutunun içine bir çizgi çizilerek aktiviteler arasında öncelik ilişkisi oluşturulur. A aktivitesi B'nin hemen önceki faaliyeti olduğundan A ve B aktivitelerinin kesiştiği hücre 2 kısıma ayrılır. Bütün ilişkili aktiviteler için bu yapıldığında Tablo 2.11'de belirtilen Lojik ilişki matrisi elde edilmiş olur.

Tablo 2.11: Lojik ilişki matrisi.

ACT	A	B	C	D	E	F	G	I	H	J	Fn		
DUR	3	3	3	7	4	6	4	9	6	8	0		
ES												EF	ACT
													A
													B
													C
													D
													E
													F
													G
													I
													H
													J
													Fn

- A faaliyeti ilk faaliyet olduğundan $ES_A=0$ olduğu aşikârdır. $EF_i=ES_i+DUR_i$ ve $ES_i=\text{Max}(EF_{HÖF})_i$ formülizasyonu ile tüm aktivitelerin ES ve EF değerleri bulunup yerlerine yazılarak ES-LS Matrisi oluşturulur.

Tablo 2.12: ES(Early start) –LS (Late start) matrisi.

ACT	A	B	C	D	E	F	G	I	H	J	Fn		
DUR	3	3	3	7	4	6	4	9	6	8	0		
ES	0	3	3	3	6	6	8	10	10	12	20	EF	ACT
												3	A
												6	B
												8	C
												10	D
												10	E
												12	F
												12	G
												19	I
												16	H
												20	J
												20	Fn

- ES-LS matrisi oluşturulduktan sonra $FF_i=ES_i-EF_i$ işlemi ile her aktivitenin FF değeri bulunarak bağımlılık çizgisinin üst kısmına yazılır. Her bağımlılık satırındaki MIN_{FF} değerleri satırın solundaki üçgensel alanın üst kısmına yazılarak Serbest Bolluk (Free Float) matrisi oluşturulmuş olur.

Tablo 2.13: Serbest bolluk (Free Float) matrisi.

ACT	A	B	C	D	E	F	G	I	H	J	Fn	EF	ACT
DUR	3	3	3	7	4	6	4	9	6	8	0		
ES	0	3	3	3	6	6	8	10	10	12	20		
	0	0	0	0								3	A
		0			0	0	2					6	B
			0				0					8	C
				0				0		2		10	D
					0				0	2		10	E
						0				0		12	F
							0			0		12	G
								1			1	19	I
									4		4	16	H
										0	0	20	J
											0	20	Fn

- Son aktivite bitiş aktivitesi olduğundan ve kritik yol üzerinde bulunma zorunluluğu olduğundan $TF_{Fn}=0$ olduğu aşikârdır. TF değerleri sondan başa doğru hesaplanır. Her öncelik çizgisinin altına gelecek değer, öncelik çizgisinin üzerindeki FF değeri ile aynı sütunun en altında bulunan TF değeri toplanarak bulunur. Öncelik çizgisinin altında kalan TF değerlerinin minimum olanı her satırın en solundaki TF üçgensel alanına yerleştirilir.
- Bu yöntemde LS ve LF değerlerine ihtiyaç bulunmamakla beraber eğer hesaplanmak istenirse $LS=ES+TF$ ve $LF=LS+TF$ formülizasyonu ile bulunabilir.

Tablo 2.14: Matris metot sonuç tablosu.

ACT	A	B	C	D	E	F	G	I	H	J	Fn	EF	ACT
DUR	3	3	3	7	4	6	4	9	6	8	0		
ES	0	3	3	3	6	6	8	10	10	12	20		
	0	0	0	0								3	A
		0			0	0	2					6	B
			0				0					8	C
				0				0		2		10	D
					0				0	2		10	E
						0			4	2		12	F
							0			0		12	G
								1			1	19	I
									4		4	16	H
										0	0	20	J
											0	20	Fn

2.4. ZAMAN MALİYET ÖDÜNLEŞİM PROBLEMİ

Kritik yol, bir proje şebekesinde planlanan aktivitelerin en uzun yoludur. Kritik yolun hesabıyla bir projeyi tamamlamak için asgari, aynı zamanda azami süre belirlenebilir. Proje yönetiminde kritik yol önemlidir çünkü kritik yol üzerindeki bir faaliyetin gecikmesi projenin gecikmesine sebep olur. Basit bir proje şebekesinde her faaliyetin süresi önceden belirlenmiş olmasına rağmen, aynı zamanda maliyetin bir fonksiyonu olarak değişebilir. Bir kazı işinde işçi sayısını artırmak süreyi kısaltabilir, bu durumda maliyetin artması kaçınılmaz olmaktadır. Eğer bir faaliyet kritik ise, zaman ve maliyet arasında bir ödünleşim yapılarak faaliyette değişiklik yapılabilir. Genel anlamda bu iki amacın dengelenmesi problemi, zaman-maliyet ödünleşim problemi olarak tanımlanır (Geem, 2010).

Bir proje için optimum bir zaman-maliyet dengesi kurma kabiliyeti, rekabet ortamında en büyük kârlılığı elde ederken rakiplerini eleme şansını artırabileceğinden, proje süresinin ve maliyetinin etkin bir şekilde planlanması ve tahmin edilmesi, müteahhit ve işverenler için çok önemlidir (Park & Chapin, 1992). Aslında, zaman ve maliyet karmaşık bir şekilde ilişkilidir. Klasik zaman maliyeti tahmin kavramları, doğrudan maliyet ve etkinlik süresi arasında ters bir ilişki kurar ve proje süresinin kısaltılması durumunda işgücü ve malzeme maliyetlerinde bir artışa işaret eder (Adrian, 1979).

Günümüzün rekabet ortamında, en az zaman ve maliyetle bir inşaat projesinin tamamlanması, bununla birlikte kalite ve performans düzeyinin korunmasında maliyet yönetimi kritik öneme sahiptir. Maliyet, proje yönetiminin farklı aşamalarında, karar verme sürecinde belirleyici etkidir. Yapılan çalışmalarda zayıf maliyet tahminlerinin projenin başarısızlığının altında yatan ana faktör olduğu görülmüş ve maliyet kontrolü önemli ölçüde dikkat çekmektedir (Raoofpahah & Hassanlou, 2013; Zheng ve diğ., 2002).

Proje faaliyetleri, ekip büyüklükleri, ekipman ve malzemeler gibi mevcut kaynaklar kapsamında planlanmalıdır. Faaliyet süresi kaynak kullanılabilirliğinin bir fonksiyonu olarak incelenebilir. Ayrıca, farklı kaynak kombinasyonlarının kendi maliyetleri vardır. Proje planı, direkt maliyetler ile proje tamamlanma süresi arasında bir denge sağlamalıdır. Bir inşaat faaliyetinin süresinin kısaltılması genellikle ek işgücü ve tesisin devreye sokulmasını veya alternatif yapım tekniklerinin benimsenmesini gerektirir ve bunlar genellikle yükleniciye ek maliyet getirir. Örneğin, verimi yüksek ekipmanların kullanılması veya daha fazla işçi

çalıştırılması zaman kazandırabilir, ancak bu durum proje maliyetini artırabilir (Azaron, ve diğ., 2005). Program sıkıştırmasıyla ilişkili toplam maliyetleri en aza indirmek için, yükleniciler genellikle planlama kararları alındığında inşaat faaliyetleri için en uygun zaman-maliyet ilişkilerini kurmaya çağılmaktadır (Zheng ve diğ., 2002).

Faaliyet süresi ve ilgili faaliyet maliyeti birbiriyle ilişkilidir. Aslında, çoğu gerçek projede, karar vericiler her zaman toplam proje maliyeti ile proje tamamlama zamanı arasındaki dengeyi değerlendirmek durumundadırlar. Bazen proje maliyetini azaltmak için, proje tamamlanma süresinden fedakârlık edilebilir. Bazı durumlarda ise projeyi zamanında bitirebilmek için proje maliyetini artırmak bir çözüm olabilir. Örneğin, daha fazla işçi çalıştırma kararı, projenin gerçekleştirilme hızını, projenin bitiş tarihinden önce bitirilebileceği şekilde artırabilir, bununla birlikte toplam proje maliyeti artar. Dolayısıyla proje süresi kısıtını karşılamak için, proje yöneticilerinin bir projeyi minimum maliyetle tamamlamak için bir zamanlama bulmaları gerekmektedir (Ke ve diğ., 2009).

Zaman maliyet ödünleşim problemi geleneksel olarak matematiksel programlama modellerinden lineer programlama, dinamik programlama ve bunların birleşimi olan lineer dinamik hibrit programlama teknikleriyle ele alınmıştır. Lineer programlamanın, ilgili fonksiyonlar için spesifik varsayımlar gerektirmesi, dinamik programlamanın çok fazla deneme ihtiyacı ve bunlara rağmen yerel optimum değerlerine takılıp kalmaları deterministik yaklaşımı verimsiz kılmaktadır (Geem, 2010).

Zaman–maliyet ödünleşim probleminin çözümünde, Genetik Algoritmalar yoğun bir şekilde kullanılmıştır. Genetik Algoritmalar dışında farklı doğal süreçlerden esinlenen, Karınca Kolonisi Optimizasyonu (ACO), Memetik Algoritmalar (MA), Parçacık Sürü Optimizasyonu (PSO), Sıçrayan Kurbağa Algoritması (SFL) gibi algoritmalar da belirtilen problemin çözümünde kullanılmıştır (Azeez & Alsaffar, 2014). Elbeltagi ve diğ., (2005) beş farklı evrimsel tabanlı optimizasyon algoritması kullanarak zaman-maliyet ödünleşim problemi modeli geliştirmişlerdir. Bunun yanı sıra algoritma verimliliklerini; işlem zamanı, yakınsama hızı ve sonuçların kalitesi yönünden karşılaştırarak, her algoritma için en iyi operatörleri belirlemeye yönelik ilkeler geliştirmişlerdir. Yine de bu çalışmalar genel olarak projenin en kısa sürede tamamlanabilmesi için gereken maliyet minimizasyonu ile ilgilidir ki bu farklı süreler için ortaya çıkabilecek maliyet avantajlarını da bilmek isteyen müteahhit ve işverenler için yetersiz kalmaktadır (Afshar, ve diğ., 2007).

Feng ve diğ., (1997) zaman-maliyet ödünleşim problemi ile ilgili Genetik Algoritma prensiplerine dayanan bir algoritma ve algoritmayı efektif bir şekilde çalıştırabilen bir bilgisayar programı sunmuşlardır. Program algoritmanın uygulaması için pratik bir araç olmuştur (Azeez ve Alsaffar, 2014). Feng ve diğ., (2000) önceki çalışmalarında, proje ağlarının olasılıklı yapısını da gözönünde bulundurarak optimum çözümler aramışlardır. Zaman - maliyet ödünleşim problemi, belirsizlik altında daha reel sonuçlar sunmuştur.

Li ve Love, (1997) , Feng ve diğ., (1997)'nin problemlerini daha kısa sürede çözümleyebilen bir GA modeli sunmuşlardır. Hegazy, (2002) zaman-maliyet ödünleşim probleminin sonucunu daha az iterasyon ile bulmayı amaçlamıştır.

Bu sebeplerden dolayı Zaman Maliyet Ödünleşim Probleminde (TCTP) Genetik Algoritma (Feng ve diğ., 1997 ; Zheng ve diğ., 2004), Karınca Kolonisi Optimizasyonu (Xiong ve Kuang, 2008), Parçacık Sürü Optimizasyonu (Yang I. T., 2007) gibi optimum veya yakın optimum çözümlere makul sürelerde ulaşan metasezgisel algoritmalara yönelim gerçekleşmiştir (Geem, 2010).

Bu tez çalışması kapsamında evrimsel tabanlı metasezgisellerden en bilinen, Genetik Algoritma yaklaşımı kullanılarak, çok amaçlı, ayrık, çok modlu zaman maliyet ödünleşim problemi incelenmiş, farklı mod bileşenlerinden elde edilen sürelerle karşılık gelen minimum maliyet değerleri bulunarak birbirine baskın çözümler değerlendirilmiş ve pareto cephesi elde edilmiştir.

2.5. OPTİMİZASYON

Optimizasyon bir sonucu daha iyi yapma sürecidir. Bir mühendis veya bilim insanı, yeni bir düşünceyi uyandırır ve bu fikir üzerinde optimizasyon geliştirir. Bir başlangıç konsepti üzerinde farklı varyasyonları deneyerek edinilen bilgilerle fikir üzerinde gelişmeler kaydedilir. Terminolojideki "en iyi" terimi, birden fazla çözüm bulunduğunu ve çözümlerin eşit değerde olmadığını gösterir. "En iyi"nin tanımı, eldeki soruna, çözüm yöntemine ve izin verilen toleranslara göre yapılır. Başka bir deyişle optimum çözüm, problemi formülize eden kişiye bağlıdır (Haupt ve Haupt, 2004).

2.5.1. Çok Amaçlı Optimizasyon

Bir optimizasyon problemine birden fazla amaç eklemek problemi karmaşıklaştırır. Örneğin bir yapı tasarımını optimize ederken, maliyet ve güvenlik açısından optimum bir tasarım istenebilir. Böyle bir durumda bu iki amaç birbiriyle çatışır ve çözüm geliştirmek için amaçlar arasında bir ödünleşim olması gerekir. Böyle bir durumda en ucuz, en güvenli tasarımlar ve bunların yanı sıra bu iki amacın farklı bileşimlerinden oluşan sonsuz sayıda tasarım da mümkün olacaktır. Bu farklı tasarımların oluşturduğu çözüm grubu, pareto seti olarak tanımlanabilir. Tasarımların güvenlik-maliyet grafiğindeki en iyi çözümlerde oluşan eğri ise pareto cephesi olarak bilinir. Bir tasarım diğer tasarımlar tarafından baskılanmadığı sürece en uygun pareto çözümdür. Pareto optimum çözümü en az bir açıdan başka bir çözüme göre daha iyi olmalıdır(Shrivastava ve diğ., 2012).

Gerçek hayatta karşılaşılan birçok problem, aynı anda birden fazla amaç fonksiyonunun optimizasyonu şeklindedir. Birden fazla hedefin eşzamanlı olarak optimize edilmesi, çok amaçlı optimizasyonda, problemi daha karmaşık hale getirir. Gerçek hayatta birçok çok amaçlı ve karmaşık problemlerin varlığı ve meta sezgisel prosedürlerin bu problemlerle başa çıkabilme kabiliyeti son yıllarda bu araştırma alanını güçlü bir şekilde geliştirmektedir (Goldberg & Richardson, 1987).

Uzun süredir tek amaçlı optimizasyon için kullanılan evrimsel tekniklerin, birden fazla amaç fonksiyonuyla birleştirilmesi ile uygunluk fonksiyonu araştırmaları popülerlik kazanmıştır (Coello C. C., 2000).

Optimizasyon ve arama problemleri genelde tek bir kritere indirgenmiştir, amaç fonksiyonu olarak nitelendirilen bu kriter, uygunluk fonksiyonuna dönüştürülerek üreme planları ve operatörler yardımıyla çözüm ilerletilir. Bu yaklaşım pek çok problemde iyi sonuç verirken aynı anda birkaç kriterin mevcut olduğu zamanlarda farklı birkaç amacı tek bir sayıya indirmek mümkün olmamaktadır. Bu durumda, problem çok amaçlı veya çok kriterli optimizasyon problemi olarak ele alınır (Mendes, 2015).

Prensipte çok amaçlı optimizasyonda (MOP) “optimum” kelimesinin net bir tanımı yoktur; hatta amaçlar arasındaki uyumsuzluk ve çelişki nedeniyle tüm amaçlara karşılık gelen kesin bir üstün çözüm olmak zorunda değildir. Çözümler birbirleriyle basitçe karşılaştırılamayacağından, optimizasyondan elde edilen “en iyi” çözüm, insan karar

vericilerinin olası bir çözüm havuzundan yaptıkları öznel seçime karşılık gelecektir (Zheng ve diğ., 2004).

Klasik yöntemler, MOP'u çok amaçlı ağırlıklandırma (MOW) veya bir uygunluk fonksiyonu (çok amaçlı fayda analizi) kullanarak bir skaler optimizasyona indirgemektedir. Çok amaçlı ağırlıklandırma, karar vericilerin her bir hedefin önceliğini karar verme sürecine dahil etmelerini sağlar. Matematiksel olarak, tüm hedeflere eşit ağırlık verilmesiyle elde edilen çözümler en az amaç karışıklığı sağlamaktadır; ancak çoğu durumda, her bir amaç ilk olarak ayrı ayrı optimize edilir ve genel amaç değeri, ağırlık faktörlerine bağlı olarak değerlendirilir. MOW'un zayıflığı, global optimumun genellikle sadece baskın amaçta olmasıdır (Feng ve diğ., 1997).

Zadeh'in (1965) çalışmasının, her bir amaç fonksiyonuna ağırlıkların atanmasını savunan ve bunları tek amaçlı fonksiyona dönüştüren ilk çalışma olduğunu söylenebilir. Daha yakın zamanlarda, Gen ve Cheng (2000) inşaat TCO probleminde adaptif ağırlık yaklaşımını (AWA) benimsemiştir (GC yaklaşımı olarak da bilinir).

GC yaklaşımında Gen ve Cheng (2000) aşağıdaki formülleri önermiştir:

$$Z^+ = \{Z_c^{max}, Z_t^{max}\} \quad (2.2)$$

$$Z^- = \{Z_c^{min}, Z_t^{min}\} \quad (2.3)$$

Z_c^{max} , mevcut popülasyonda maksimum toplam maliyet değeri, Z_t^{max} , mevcut popülasyonda maksimum süre değeri, Z_c^{min} , mevcut popülasyonda minimum toplam maliyet değeri, Z_t^{min} , mevcut popülasyonda minimum süre değeri olmak üzere:

$$w_c = 1/(Z_c^{max} - Z_c^{min}), w_t = 1/(Z_t^{max} - Z_t^{min}) \quad (2.4)$$

$$f(x) = w_c(Z_c^{max} - Z_c) + w_t(Z_t^{max} - Z_t) \quad (2.5)$$

2004 yılında, Zheng ve diğ. (2004) çok amaçlı problemle başa çıkmak için modifiye edilmiş ağırlık yaklaşımını (MAWA) önermişlerdir. MAWA'ya göre, adaptif ağırlıklar aşağıdaki dört koşulda formüle edilir:

1) $Z_t^{\max} \neq Z_t^{\min}$ ve $Z_c^{\max} \neq Z_c^{\min}$ için:

$$v_c = \frac{Z_c^{\min}}{Z_c^{\max} - Z_c^{\min}} \quad (2.6)$$

$$v_t = \frac{Z_t^{\min}}{Z_t^{\max} - Z_t^{\min}} \quad (2.7)$$

$$v = v_c + v_t \quad (2.8)$$

$$w_c = v_c/v \quad (2.9)$$

$$w_t = v_t/v \quad (2.10)$$

$$w_c + w_t = 1 \quad (2.11)$$

2) $Z_t^{\max} = Z_t^{\min}$ ve $Z_c^{\max} = Z_c^{\min}$ için:

$$w_c = w_t = 0,5 \quad (2.12)$$

3) $Z_t^{\max} = Z_t^{\min}$ ve $Z_c^{\max} \neq Z_c^{\min}$ için:

$$w_c = 0,1 ; w_t = 0,9 \quad (2.13)$$

4) $Z_t^{\max} \neq Z_t^{\min}$ ve $Z_c^{\max} = Z_c^{\min}$ için:

$$w_c = 0,9 ; w_t = 0,1 \quad (2.14)$$

Zheng ve diğ. (2004) önerilen adaptif ağırlığa göre denklem 2.15'te belirtilen uygunluk fonksiyonunu önermiştir:

$$f(i) = w_t \frac{(Z_t^{\max} - Z_t) + \gamma}{(Z_t^{\max} - Z_t^{\min}) + \gamma} + w_c \frac{(Z_c^{\max} - Z_c) + \gamma}{(Z_c^{\max} - Z_c^{\min}) + \gamma} \quad (2.15)$$

Denklem 2.15'te belirtilen semboller aşağıdaki gibi ifade edilir.

- γ : 0 ile 1 arasında küçük pozitif bir rassal sayıdır.
- Z_c : Mevcut popülasyonda i. çözüme karşılık gelen toplam maliyettir.
- Z_t : Mevcut popülasyonda i. çözüme karşılık gelen süredir

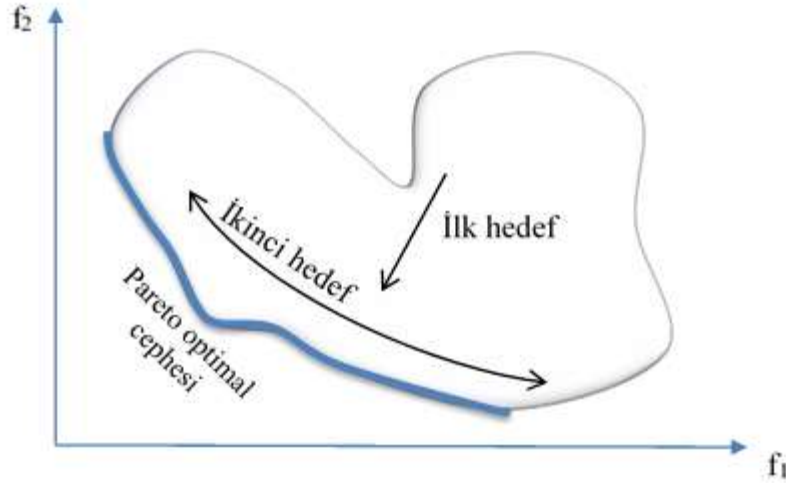
2.5.2. Pareto Cephesi

Birden fazla amaç fonksiyonuna sahip olmak “optimumluk” kavramını deęiřtirir, çünkü çok amaçlı problemlerde hedef global optimizasyondaki gibi tek bir çözüm bulmak deęil, çözümler arasında bir uzlaşma (ödünleşim) sağlamaktır (Coello, Veldhuizen, & Lamont, 2002).

Çok amaçlı optimizasyonda birden fazla hedef bulunur:

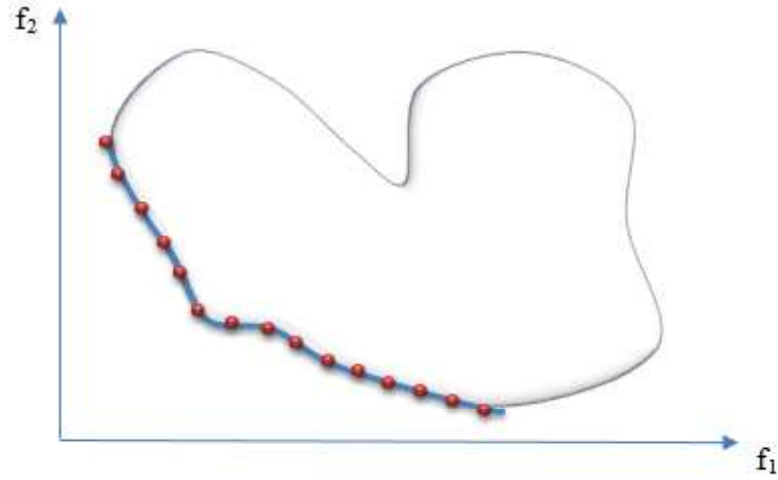
- Pareto optimal çözümlerine mümkün olduğunca yakın çözümler keşfetmek
- Pareto cephesinde önceden elde edilmemiş mümkün olduğunca çeşitli çözümler bulmak.

İlk hedef pareto cephesine doğru bir arama gerektirirken ikinci hedef Şekil 2.31’de görüldüğü gibi pareto optimal cephesi boyunca arama gerektirir (Deb, 2001).



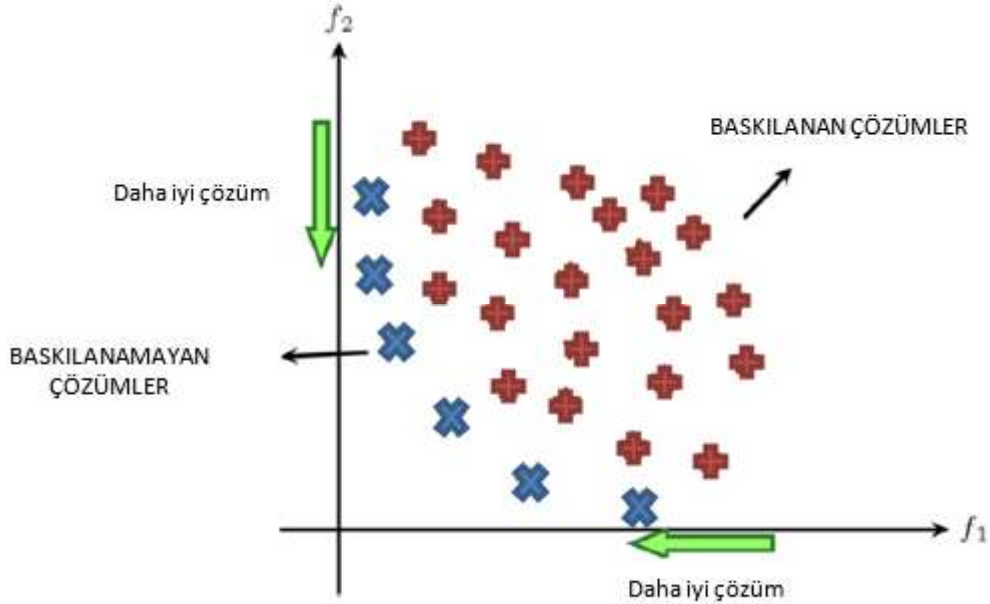
Şekil 2.31: Pareto optimal cephede arama hedefleri (Deb, 2001).

Çeşitlilik içeren bir çözüm kümesi tüm pareto cephesini tekdüze olarak kaplayan bir çözümü temsil eder. Çeşitliliğin ölçüsü, derece (aşırı çözümlerin yayılması) ve dağılım (çözümler arasındaki görel mesafe) olarak ayrılabilir (Zitzler, Deb, & Thiele, 2000). İdeal bir pareto grafięi Şekil 2.32’te gösterilmiştir.



Şekil 2.32: İdeal pareto çözümleri (Deb, 2001).

Daha önce de belirtildiği üzere, çok amaçlı optimizasyon problemlerinin amacı çoklu ve çelişen hedefler arasında en iyi ödünleşimi bulmaktır. Bu problemlerin tüm amaçlarını aynı anda en iyi duruma getiren birden fazla çözüm olacaktır ve bu çözümler arasında belirgin bir üstünlük yoktur. Bu nedenle optimum çözüm gibi tek bir en iyi değer seçmek mümkün değildir. Dolayısıyla pareto cephesi (pareto front) olarak nitelendirilen, diğer çözümlerden üstün olan daha iyi çözümler dizisi oluşturulabilir. Uygun çözümler arasından pareto cephesine ait olan çözümler diğer çözümler tarafından baskılanamayan (non-dominated) çözümler iken diğer çözümlerin baskın geldiği ve elenen çözümler baskılanmış (dominated) çözümler olarak adlandırılır. Pareto optimal cephe üzerindeki hiçbir çözüm bir diğerine baskınlık sağlamadığından, bu çözümlerin amaç fonksiyonunu eşit derecede gerçekleştirdiği kabul edilebilir (Afshar ve diğ., 2007).



Şekil 2.33: Baskılanan ve baskılanamayan çözümler (Afruzi ve diğ., 2013).

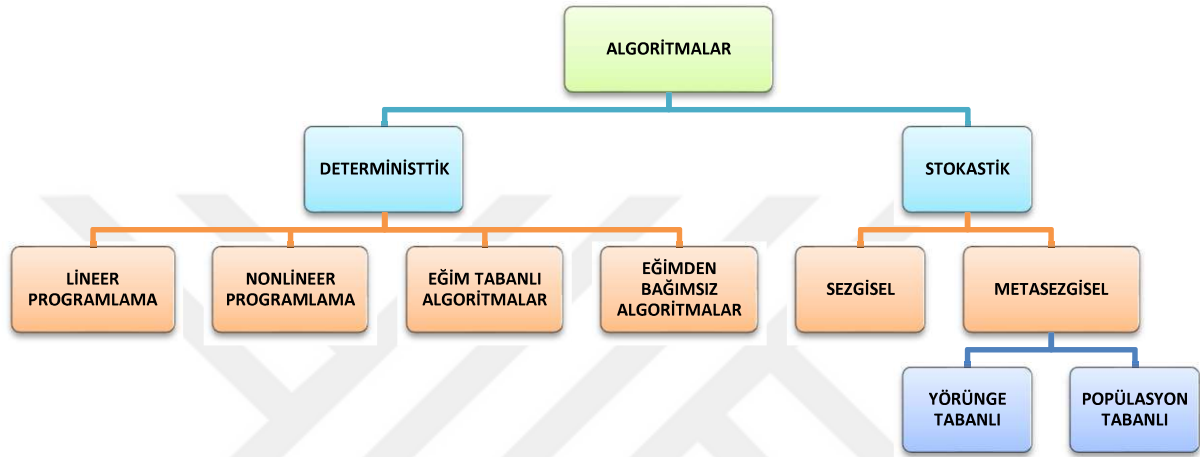
Şekil 2.33'te f_1 'in süreyi f_2 'nin maliyeti temsil ettiği varsayıldığında, aynı süreye ait çözümler arasında minimum maliyetli çözümlerin, aynı maliyete sahip çözümler arasında ise daha düşük süreli çözümlerin diğerlerine baskın geldiği ve pareto çözümü olarak nitelendirildiği görülmektedir. Pareto çözümlerinin birbirleri arasında ise optimallik açısından bir kıyaslama yapmak mümkün değildir. İnşaat projelerinde projenin bitim süresi veya nakit akışlarına göre, projeye özel bir gün ve maliyet seçimi yapılabilir. Uygulanan proje özelinde belirlenen süre maliyet çözümü optimumdur denebilir.

2.5.3. Deterministlik ve Stokastik Yaklaşım

Genel olarak, optimizasyon algoritmaları deterministlik algoritmalar ve stokastik algoritmalar olmak üzere iki kategoriye ayrılabilir. Deterministlik algoritmalar titiz bir prosedür izler, problem modelinin değişken ve fonksiyonları ve izlediği yol tekrarlanabilir bir yapıdadır. Öyle ki, deterministlik bir algoritma aynı başlangıç noktası ile her zaman aynı yolu izleyecektir (lineer programlama, dal sınır algoritması, tepe tırmanma tekniği vb).

Stokastik algoritmalar daima rassallık içerir. Literatürde en yaygın kullanım alanı bulan genetik algoritmada, program her çalıştırıldığında, algoritmada kullanılan rassal sayılar dolayısıyla popülasyondaki dizeler veya çözümler farklı olacaktır. Nihai çözümde büyük farklar olmasa da her bir bireyin izlediği yol tekrarlanabilir değildir.

Üçüncü bir yöntem olarak hibrit algoritmalar örnek verilebilir. Hibrit algoritmalar oluşturularak deterministik bir algoritmaya, rassal başlangıç değerleri atanarak çözüm aranabilir, bu sayede algoritmanın yerel optimum değerlere takılması engellenmiş olacaktır. Yine de içinde rassallık barındıran bu tür algoritmaları stokastik algoritmalara dahil etmek daha anlamlıdır. Şekil 2.34'te algoritmaların sınıflandırılmasına yönelik genel bir grafik verilmiştir.



Şekil 2.34: Algoritmaların sınıflandırılması.

2.6. SEZGİSEL VE METASEZGİSEL METOTLAR

Sezgisel yöntemler, karmaşık bir probleme deneme yanılma yoluyla, makul ve uygulanabilir bir zamanda, kabul edilebilir çözümler üreten çözüm stratejileridir. Probleminin karmaşıklığı, mümkün olan her çözümü veya kombinasyonu aramayı imkansız kılar; amaç, kabul edilebilir bir zaman ölçeğinde iyi veya olurlu çözümler bulmaktır. Öyle ki en iyi çözümlerin bulunabileceğinin garantisi yoktur ve bir algoritmanın çalışıp çalışmayacağı, ya da neden çalıştığı sorusunun cevabı yoktur. Ana fikir, çoğu zaman çalışacak ve kaliteli çözümler üretebilecek pratik ve verimli bir algoritma kullanmaktır. Ortaya çıkan kaliteli çözümleri arasından, -bu tür bir optimumluğun garantisi olmasa da bazılarının neredeyse en uygun olduğu düşünülmektedir (Yang, 2010).

Sezgisel algoritma ve metotlar doğal fenomenlerden esinlenerek oluşturulur. Bu tip algoritmalar optimum çözümü garanti etmemekle birlikte optimum çözüme yakınsama ve yakın optimum çözümü bulma eğilimindedir. Kolay anlaşılabilir olması, gerçek hayat problemlerinin

kesin çözüm teknikleriyle tanımlanmasının zor olması nedeniyle sezgisel metotlar tercih edilir. Bazen kesin çözümün bir parçası olarak da kullanılabilir (Karaboğa & Görkemli, 2011).

Genel amaçlı sezgisel yöntemler; biyoloji tabanlı, fizik tabanlı, sürü tabanlı, sosyal tabanlı, müzik tabanlı ve kimya tabanlı olarak gruplandırılabilirler. Ayrıca bunların birleşimi olan hibrit algoritmalar da sıklıkla kullanılmaktadır (Alataş, 2007).

Sezgisel yöntemlerin sınıflandırılması ve terminoloji ile ilgili literatürde bazı karışıklıklar söz konusudur. Genel anlamda bir sınıflandırma problemin amaç sayısı, kısıt durumu, değişkenler, fonksiyon tipi, çözüm uzayının geometrisi ve belirlilik yönünden yapılabilir. Amaç sayısına göre sınıflandırma yapılırsa; optimizasyon, tek amaçlı (single objective) $M=1$ ve çok amaçlı veya çok kriterli (multiobjective) $M>1$ olarak iki kategoriye ayrılabilir (Yang X.-S. , 2010). Gerçek hayat problemleri genellikle çok amaçlı olmakla beraber birbiriyle çelişen amaçlar problemin zorluk derecesini de artırmaktadır (Kaya & Fıglalı, 2016). Örneğin, bir otomobil motoru tasarlarken yakıt verimliliğini en üst düzeye çıkarmak, karbon dioksit emisyonunu en aza indirmek ve gürültü seviyesini düşürmek istersek, problem üç amaçlı olarak değerlendirilebilir. Bu çalışma, farklı sürelerde gerçekleşen maliyetlerin minimize edilmesi amaçlandığından çok kriterli bir problem olarak ele alınabilir. Problem, kısıt yönünden ele alınırsa kısıtlı ve kısıtsız olarak değerlendirmek mümkündür.

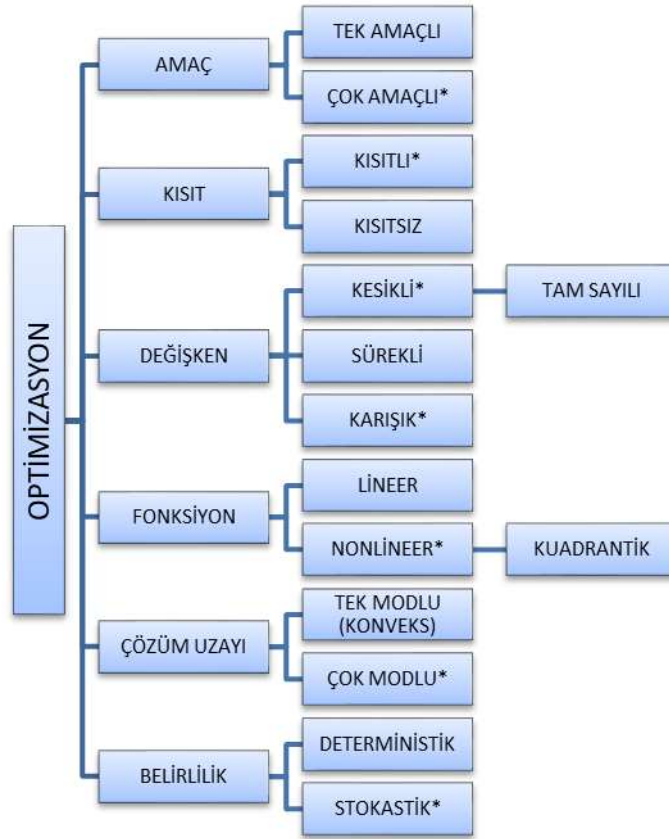
Amaç fonksiyonu lineer ve nonlineer olarak sınıflandırılabilir. Bununla birlikte amaç fonksiyonu lineer olsun olmasın, kısıtlarda lineer ise problem lineer kısıtlı probleme dönüşür. Eğer hem amaç fonksiyonu hem kısıtlar lineer ise problem lineer programlama (planlama ve/veya optimizasyon anlamında kullanılır) problemine dönüşür. Bu çalışmada da incelenmiş olduğu gibi bazı problemler değişkenler açısından ayrık olabilir. Bu durumda değişkenler açısından kesikli ve sürekli olarak bir sınıflandırma yapmak mümkündür.

Literatürde ayrık optimizasyon, kombinatoriyal optimizasyon olarak da kullanılmaktadır. Bazı durumlarda değişkenler hem kesikli hem de sürekli olabilmektedir bu tip problemler karışık tip (mixed type) olarak değerlendirilmektedir.

Problemleri tek modlu ve çok modlu olarak sınıflandırmak da mümkündür. Amaç fonksiyonu konveks bir yapıda ve tek global optimum değerine sahipse tek modlu (unimodal) olarak sınıflandırılır. Birçok amaç fonksiyonu kuadrantik yapıda olup birden fazla moda sahiptir ve çözümü tek modlulara göre çok daha zordur (Yang X.-S. , 2010). Birçok metasezgisel metotta,

programın nonlinear amaç fonksiyonunun yerel optimum noktalarına takılıp kalmasını engellemek adına bir takım yaklaşımlar sergilenmektedir.

Tasarım değişkenlerinin tamamının bilindiği bir problem tipi deterministik olarak değerlendirilebilirken gerçek hayatta bu pek mümkün olmamaktadır. Amaç fonksiyonu ve değişkenlerden herhangi biri rassallık içerdiği anda problem stokastik optimizasyon problemine dönüşür. Bu çalışmada uygulanan genetik algoritmada başlangıç aşamasından itibaren rassal dizi ve değerler kullanılmaktadır. Başlangıç popülasyonu rassal olarak oluşturulmakta, yeni nesiller oluşturulurken çaprazlama sırasında rassal değerler kullanılmakta, problemin yerel optimum değerlere takılmasını engellemek adına yapılan mutasyon işlemi de yine rassallık içermektedir. Bu sebeplerle incelenen problem de stokastik optimizasyon problemi olarak sınıflandırılmaktadır. Şekil 2.35’te optimizasyon problemlerinin sınıflandırılmasına ilişkin bir grafik verilmiştir. “*” ile belirtilen özellikler eklendikçe ileride anlatılacağı üzere problemin zorluk derecesi artmaktadır.



Şekil 2.35: Optimizasyon Problemlerinin Sınıflandırılması (Yang X.-S. , 2010).

2.6.1. Metasezgisel Algoritmalar

Meta sezgisel algoritmaların çoğu doğanın soyutlanmasına dayanarak geliştirilmiştir. Doğanın problem çözme becerisinden esinlenilerek sezgisel ve meta sezgisel algoritmalar oluşturulmuştur. Daha spesifik olarak, bazı algoritmalar Darwin'in evrim teorisinden esinlenilerek geliştirilmiş ve biyolojik tabanlı (biyo-inspired) algoritmalar olarak adlandırılır (Yang X.-S. , 2010).

En iyi çözümlerin seçimi ve rassallık meta sezgisel algoritmaların temel iki bileşenidir. En iyi seçim, çözümlerin optimalliğe yakınsamasını sağlarken, rasgele seçme, çözümlerin yerel optimuma takılmasını önler ve aynı zamanda çözümlerin çeşitliliğini artırır. Bu iki bileşenin iyi kombinasyonu genellikle global optimalliğe ulaşmayı sağlar (Yang X.-S. , 2010).

Meta sezgisel algoritmalar birçok açıdan sınıflandırılabilir. Bunlardan biri popülasyon ve yörunge tabanlı olarak sınıflandırmaktır. Örneğin genetik algoritmaların bireyleri bir dizi seti kullandıklarından popülasyona dayalı iken birden fazla parçacık veya ajan kullanan parçacık sürü optimizasyonu (PSO) ajan tabanlı algoritmalar olarak sınıflandırılabilir (Yang X.-S., 2010).

2.6.2. Evrimsel Tabanlı Optimizasyon ve Evrimsel Algoritmalar

Evrimsel tabanlı algoritmalar işleme süresini kısaltmak ve çözümlerin kalitesini artırmak; özellikle de yerel optimum noktalarına takılmayı engellemek amacıyla son 15 yılda uygulamaya konmuştur (Elbeltagi ve diğ., 2005).

Evrimsel tabanlı algoritmalar doğal biyolojik evrim metaforunu ve / veya türlerin sosyal davranışlarını taklit eden stokastik arama yöntemleridir. Evrimsel algoritmalar, genetik algoritmalar, evrim stratejileri, evrimsel programlama ve genetik programlama; mimari tasarım, mühendislik optimizasyonu, fabrika iş planlaması, elektronik devre tasarımı, sinyal işleme, ağ konfigürasyonu, robotik kontrol gibi çeşitli uygulamalarda büyük avantaj sağladığı için giderek daha fazla kullanılmaktadır. (Dasgupta & Michalewicz, 1997)

Evrimsel algoritmalar, evrimsel mekanizmayı tasarımlarında ve uygulamalarında kullanan bilgisayar tabanlı çözüm sistemleridir. Evrim mekanizmasının kullanımı 1950'li yıllarda başlamış 1960'lı yıllara gelindiğinde bu konuda Evrimsel Programlama, Evrimsel Stratejiler,

Genetik Algoritmalar olmak üzere üç farklı uygulama geliştirilmiştir. Daha sonra bu çalışmalar Evrimsel Hesaplamalar çatısı altında toplanmıştır ve yine 1990'lı yılların başlangıcında dördüncü bir yöntem de Genetik Programlama adı ile bu çatı altına ilave edilmiştir. En önemli evrimsel algoritmaları kombinatoriyal problemlerin optimizasyonunda kullanılan Genetik Algoritmalar; üreme süreci kullanılmadan sürekli fonksiyonların optimizasyonunda kullanılan Evrimsel Programlama, üreme süreci kullanılarak sürekli fonksiyonların optimizasyonunda kullanılan Evrimsel Stratejiler, Sınıflandırma Sistemleri (Classifier Systems) ve program geliştirmede kullanılan Genetik Programlamadır (Sarı, 2008).

Optimizasyon süreci boyunca genellikle en iyi çözüm aranır. Küçük arama uzayları için klasik metotlar yeterliyken daha büyük arama uzayları ve karmaşık problem tipleri için özel yapay zeka teknikleri kullanılmalıdır. Evrimsel algoritmalar (EAs) bu tür teknikler arasındadır. Arama yöntemleri, hayatta kalmak için genetik kalıtım ve Darwinci çatışma gibi bazı doğal fenomenleri modelleyen stokastik algoritmalarlardır. Daha öncede belirtildiği üzere evrimsel algoritmaların çeşitli türevleri vardır, bunun yanı sıra belirtilen paradigmalardan çeşitli özelliklerini içeren birçok hibrit sistem vardır. Ancak herhangi bir evrimsel metodun yapısı çok benzerdir. Evrimsel Hesaplamalar olarak tanımlanan bu algoritmalar aşağıda açıklanmıştır.

2.6.2.1. Evrimsel Strateji

Evrimsel stratejileri (ES) 1960'ların başlarında Ingo Rechenberg ve Hans-Paul Schwefel tarafından önerilmiştir. Başlangıçta pratik problemleri çözmek için bir optimizasyon aracı olarak geliştirilen evrimsel stratejileri, her bireyi genellikle tamsayı veya gerçek sayı olarak kodlanmış bir dizi parametre olarak tanımlar.

Mutasyon operatörü, seçilen bir bireyin tüm parametrelerini, küçük değişikliklere büyük olasılık, büyük modifikasyonlara küçük olasılık vermek kaydıyla aynı anda değiştirir. rekombinasyon operatörü için her bir ebeveynin parametrelerinin bir alt kümesini kopyalamak veya tüm sayıların ortalamasını hesaplamak gibi yöntemler mümkündür.

Evrimsel stratejilerinin özelliklerini tanımlamak için özgün bir biçim geliştirilmiştir. Genel olarak nüfusun büyüklüğü " μ " ve her nesilde üretilen yavruların büyüklüğü için ise " λ " kullanılır. $(\mu + \lambda)$ -ES, gelecek nesil için hayatta kalma seçiminden önce; yavruların mevcut nüfusa eklendiği bir evrimsel stratejisidir. Bu durumda iyi çözümler, kararlı durum genetik algoritmasında veya evrimsel programlamada olduğu gibi birkaç nesil boyunca hayatta

kalabilmektedir. Diğer bir yandan $(\mu + \lambda)$ – ES formülizasyonu seçim işleminden önce yeni neslin mevcut popülasyonu tamamen değiştirdiği bir evrim stratejisini ifade etmektedir. Bir diğer yaklaşım da genetik algoritma ve evrimsel programlamada olduğu gibi en uygun çözümün çalışma sırasında atılabilmesi olasılığıdır. Yaygın olarak kullanılan bir gösterimde, iki yaklaşım sırasıyla artı (+) ve virgül (,) olarak adlandırılır. Virgül seçimi kullanıldığında, $\mu < \lambda$ tutulmalıdır. Bununla birlikte, evrim stratejilerinin neredeyse tüm uygulamalarında, yavruların büyüklüğü, her adımdaki nüfusun boyutundan çok daha büyüktür.

Çaprazlama operatörü uygulandığında gerekli ebeveyn sayısı “ ρ ” ve algoritma $(\mu / \rho, \lambda)$ -ES algoritması ile gösterilir. Ebeveynlerin sayısı her zaman nüfus büyüklüğünden daha küçüktür $\rho < \mu$. Evrim stratejilerine özgü bir yaklaşım da iç içe geçme (nestling) yaklaşımıdır. Yeni nesil geleneksel operatörlerle değil evrimsel bir alt strateji ile üretilerek ana strateji için yavru olarak kullanılmasıdır. İç strateji aslında yerel optimizasyonlar için bir araç olarak hareket etmekte ve yaygın olarak dış stratejinin parametreleri ile ilgisi olmayan parametreleri benimsemektedir. Farklı uygulamalarda, bu teknik, iç içe evrim stratejileri, hiyerarşik evrim stratejileri ve meta evrim stratejileri olarak adlandırılmıştır. Alt stratejilerin oluşturduğu kuşakları üreten algoritma $(\mu / \rho, (\mu / \rho, \lambda)^{\gamma})$ -ES ile gösterilebilir. “ γ ” izolasyon süresi olarak adlandırılır. Böyle bir teknik, evrimsel programlamada veya genetik algoritmalarda neredeyse hiç kullanılmamaktadır.

Evrimsel stratejilerinin yeni nesil üretiminin temeli mutasyonlara dayanır. Mutasyonun ne derece yapılacağını yani değişikliklerin hangi genlikte yapılacağını belirlemek gerekmektedir. Gerçek arama uzaylarında mutasyon genellikle sıfır etrafında normal bir olasılık dağılımını izleyen rassal bir değişiklik olarak tanımlanır. Bu şekilde, mutasyonların derecesinin küçük olma olasılığı büyük değişiklikler yapılması olasılığından daha muhtemel olmaktadır (Sanchez ve diğ., 2012).

2.6.2.2. Evrimsel Programlama

Lawrence J. Fogel tarafından 1960 yılında tasarlanmıştır. Genetik Algoritmalara benzeyen stokastik bir optimizasyon yöntemidir. Çaprazlama operatörünün kullanılmaması genetik algoritmadan ayrılan özelliğidir. Ebeveyn ile çocuk birey arasındaki ilişki üzerine tasarlanmıştır. Üreme süreci, her ebeveyn bireyin sadece bir çocuk bireyi mutasyon operatörü kullanılarak üretmesi ile sağlanır (Sarı, 2008).

Fogel zeki davranışların, ortamdaki modifikasyonları tahmin etme yeteneği gerektirdiğini savunduğu için çalışmalarının odak noktası öngörü kabiliyetinin evrimidir. Otomat olarak adlandırdığı makineleri evrimleşen varlıklar olarak kullanmış ve bir sonraki sembolün kendilerine sağlanan bir giriş dizisinde tahmin edebilecek bireyleri evrimleştirmeye çalışarak bir öngörü kabiliyeti göstermiştir. Orijinal algoritma, bir dizi P otomatını göz önünde bulundurmıştır. Dizi içindeki her bir birey, girişteki bir dizi çevre olarak tanımlanan sembollerle test edilmiştir.

Tahmin kabiliyeti, çok karmaşık otomatlar için bir ceza puanı olarak kabul edilen farklı kazanç fonksiyonları ile uygunluk değerine indirgenmiştir. Bireyler uygunluk değerlerine göre sıralanarak, popülasyonun yarısı kadar birey uygunluk değerleriyle ilişkili bir olasılıkla rasgele seçilip mutasyona uğratarak P bireyli yeni jenerasyon oluşturulmuş olur (Sanchez ve diğ., 2012).

Fogel'in Evrimsel Programlamayı makine öğrenme problemini çözmek için kullanması ve kromozomu temsilen otomatları kullanması EP ile optimizasyon problemini çözmek için bazı zorluklara neden olmuştur (Sanchez ve diğ., 2012).

1990'lı yıllarda birçok araştırmacı EP'yi bir optimizasyon alanı haline getirmiş ve birçok EP türü oluşturmuştur. En çok alıntı yapılan EP, değişkenleri temsil etmek için gerçek sayıların kullanıldığı aşağıdaki algoritma ile gösterilmiştir (Yu & Gen, 2010).

- > λ , μ ve σ gibi EP için parametreler atayın.
- > Başlangıçtaki popülasyonu oluşturmak ve uygunluk değerlerini değerlendirmek için uniform dağılımlı bireyleri (μ) rasgele oluşturun. $gen = 0$.
- > **while** $gen > maxgen$
- > μ bireyli nüfus oluşuncaya kadar, bireylerin her bir genine mutasyon uygula
- > Her yeni birey için uygunluk değerlerini hesapla
- > Mevcut ve yeni bireyleri birleştirip yeni nüfus için en iyileri seç $gen = gen + 1$
- > **end while**
- > EP sonuçlarını yazdır.

2.6.2.3. Genetik Programlama

Genetik Programlama (GP), Koza, (1992) tarafından bilgisayarların problemleri otomatik olarak çözmesi için bir araç olarak önerilmiştir.

Metodolojinin amacı, Neo-Darwinist kavramlardan yola çıkarak oluşturulan genetik operasyonlarla otomatik bilgisayar programları üretmektir. Genetik programlamanın ilk versiyonu, 1950'lerin sonlarına kadar uzanan yorumlanmış bir bilgisayar dili olan Lisp'de geliştirilmiştir. Lisp programının karakteristik özelliklerinden biri, kod parçalarını veri olarak ele almasından dolayı kodları değerlendirmeden önce alt programlar oluşturmayı mümkün kılmasıdır. Değişkenler ve sabitler hariç, Lisp'deki her şey bir önek ifadesi olarak kabul edilir, genetik programlamadaki ilk bireyler Lisp kodunun blokları olduğundan, bunlar aynı zamanda önek ifadeleridir. Lisp'in esnek olmasının dışında dilin verimsiz olması araştırmacıları derlenmiş diller içeren alternatif çözümlere itmiştir (Yu & Gen, 2010).

Genetik programlamadaki bireyler, benimsenen bilgisayar dillerindeki farklılıklara rağmen neredeyse her zaman ağaç olarak temsil edilmektedir. En basit formlarda, iç düğümler işlemleri açıklarken, yapraklar (terminal düğümleri) sayıları kodlar. Değişkenler, fonksiyonlar ve programlama yapıları daha karmaşık varyasyonlarda görünür.

Yeni nesil çaprazlama veya mutasyon yoluyla üretilebilir. Çaprazlama ebeveynler arasında alt ağaçların bir takası olarak modellenir. Mutasyon ise yalnızca son uygulamalarda ortaya çıkar ve genellikle ağacın rasgele bir modifikasyonunu, yeni bir bireye bir alt ağacın tanıtımını veya bir alt ağacın tek bir terminal düğümüne çökmesini içerir. Büyük popülasyonlarla yapılan ilk genetik programlama denemelerinde çoğunlukla çaprazlama ve nadiren mutasyon kullanılmıştır. Bir alt ağacın ikame edilmesi yıkıcı bir işlemdir ve kuvvetli bir yenilik sağlayabilir. Popülasyonun çok büyük olması da gen havuzunda mümkün olan tüm olası durumların mevcut olmasını sağlamaktadır (Yu & Gen, 2010).

2.7. GENETİK ALGORİTMALAR

Genetik algoritma, NP-Zor tipi problemlerin optimizasyonunu sağlayan sezgisel bir arama yaklaşımıdır. Bu prosedür, doğayı taklit eden bazı operatörler aracılığıyla güçlü olanın hayatta kalması prensibine dayanır (Goldberg D. E., 1989).

Genetik algoritmalar doğal seçim mekaniği ve doğal genetiğe dayalı arama algoritmalarıdır. Yenilikçi bir dizaynla bir arama algoritması oluşturmak üzere, dizilerden oluşan yapılar arasında en iyi olanların hayatını sürdürebilmesi, rassal bilgi alışverişi ile birleştirilmiştir. Oluşturulan her yeni nesilde en yetenekli ve hayatta kalmaya elverişli bireylerin özellikleri kullanılarak yeni bir yapay popülasyon oluşturulur ve zaman zaman daha iyi bir sonuç için

genetik bilgide rassal deęişikliklere gidilebilir. Popülasyondan elde edilen geçmiş bilgi yeni arama noktalarında verimli bir şekilde kullanılarak problem optimize edilir.

Genetik Algoritmalar genetik ve evrim ilkelerine dayanır. Evrim doğaya uyum sağlama ve genlerin gelecek kuşaklara aktarılmasını sağlayan kolektif bir olgudur. Eğer genlerin bir kombinasyonu hayatta kalma ve eş bulma konusunda makul bir şans veriyorsa, bu gen grubu optimal olmasa bile bir sonraki kuşağa aktarılır. (Goldberg D. E., 1989) (Miettinen, Neittaanmaki, Makela, & Periaux, 1999)

Örneğin biyolojik evrim ve ikili kodlama yapısında verilmiş Genetik Algoritma arasındaki benzerliği ortaya koymak adına yüksek ve düşük seslerle havlayan köpeklerin bulunduğu bir popülasyondan daha yüksek sesle havlayacak bir köpeğin nasıl evrilebileceği araştırılmıştır. Şekil 2.36’da biyolojik evrim ve binary (ikili kod) GA arasındaki analogi gösterilmiştir (Haupt & Haupt, 2004).



Şekil 2.36: Nümerik GA ve biyolojik evrim analogisi.

Genetik algoritmalar Michigan Üniversitesi öğretim üyelerinden John Holland, meslektaşları ve öğrencileri tarafından geliştirilmiştir. Araştırmalarının iki temel amacı; doğal sistemlerin

adaptif süreçlerini açıklamak, doğal sistemlerin önemli mekanizmalarını koruyan yapay sistemler yazılımını tasarlamaktır. Bu girişim doğal ve yapay sistem bilimlerinde önemli bulgulara yol açmıştır (Goldberg D. E., 1989).

Goldberg (1989)' e göre genetik algoritmayı diğer algoritmalarından ayıran başlıca özellikler aşağıdaki gibidir.

- Genetik Algoritmalar parametrelerle değil, parametre setlerinin kodlanmasıyla ilgilenir.
- Genetik Algoritmalar, tek noktadan değil, noktalar popülasyonundan arama yapar.
- Genetik Algoritmalar, türev veya yardımcı bilgileri değil, nesnel fonksiyon bilgisini kullanır.
- Rastlantısal geçiş kuralları uygular, determinist değildir.

Genetik algoritma, proje planlama ve kontrolünde kullanılan arama ve optimizasyon araçlarıdır. Geniş arama uzayına sahip problemlerde optimum ya da yakın optimum çözümlerin tanımlanmasına yardımcı olur (Sorrentino, 2013) .

Genetik algoritmaların geleneksel optimizasyon algoritmalarına kıyasla pek çok avantajı vardır ve en dikkat çekici iki avantajı, karmaşık optimizasyon problemleri, paralellik ile baş etme becerisidir. Genetik algoritmalarla, uygunluk fonksiyonunun durağan (stationary) veya durağan olmayan (non-stationary), doğrusal veya doğrusal olmayan, sürekli veya kesikli gibi çeşitli optimizasyon türleri ele alınabilir. Bir popülasyondaki nesiller bağımsız ajan gibi davrandıkları için, popülasyon (veya herhangi bir alt grup) aynı anda birçok yönden arama alanını keşfedebilir. Bununla birlikte genetik algoritmaların bazı dezavantajları da vardır. Uygunluk fonksiyonunun formülizasyonu, popülasyon büyüklüğünün kullanımı, mutasyon ve çaprazlama oranı gibi önemli parametrelerin seçimi ve yeni popülasyonun seçim kriterleri dikkatle belirlenmelidir. Herhangi uygun olmayan bir seçim, algoritmanın bir araya gelmesini zorlaştırabilir veya anlamsız sonuçların üretilmesine sebep olur. Bu dezavantajlara rağmen, genetik algoritmalar modern doğrusal olmayan optimizasyonda en yaygın kullanılan optimizasyon algoritmalarından biri olmaya devam etmektedir (Yang X.-S. , 2010).

Genetik algoritmaların bazı avantajlarını aşağıdaki gibi sıralamak mümkündür (Haupt & Haupt, 2004).

- Sürekli veya ayrık değişkenlerle optimizasyon yapılabilir
- Türev bilgisi gerektirmez
- Arama uzayının geniş bir örnekleminde arama yapılabilir
- Çok sayıda değişkenle çözümlenebilir
- Paralel bilgisayarlar için uygundur
- Karmaşık arama uzaylarında verileri optimize eder ve yerel minimumdan kurtulabilir
- Sadece tek bir çözüm değil, optimum değişken listesi sunabilir
- Değişkenleri kodlayabilir, böylece optimizasyon kodlanmış değişkenlerle yapılır
- Sayısal olarak üretilen veriler, deneysel veriler veya analitik fonksiyonlarla çalışır.

Bu avantajlarla, genetik algoritma geleneksel optimizasyon yöntemlerinin başarısız olduğu noktalarda çok çarpıcı sonuçlar ortaya koyar. Bu demek değildir ki genetik algoritma her problemi çözenin en iyi yoludur. Örneğin geleneksel metotlar, sadece birkaç değişkenli iyi tanımlanmış konveks analitik bir fonksiyonun çözümünü hızlı bir şekilde bulabilir. Bu gibi durumlarda, genetik algoritma ilk popülasyonun uygunluk değerlerini hesaplarken, cebirsel metotlar fonksiyonun minimum değerini kısa sürede bulmuş olacaktır. Buna rağmen çoğu gerçek hayat problemi bu kategorinin dışındadır (Haupt & Haupt, 2004).

GA, optimizasyon problemlerinde optimum veya yakın optimum çözümler bulmak için kullanılan bir arama tekniğidir. GA, mayoz bölünmeye dayalı bir algoritma ile global optimumu arar. İlk popülasyon rasgele üretilir ve yeni genler çaprazlama ile çoğaltılır. Genetik farklılıklar mutasyon ile oluşur ve uygun olmayan genler doğal seleksiyon işlemleri ile son bulur. (Bettemir Ö. H., 2010)

Genetik algoritma prosedürü, kromozom biçiminde kodlanan rasgele çözümlerin oluşturduğu başlangıç popülasyonu ile başlar. Bir kromozom problemin karar değişkenlerini temsil eden genlerden oluşur. Popülasyondaki kromozom sayısı popülasyon büyüklüğü olarak adlandırılır. Popülasyondaki kromozomların performansları, uygunluk fonksiyonu üzerinden değerlendirilir ve en uygun kromozomlar seçim operatörleri ile belirlenir. En iyi kromozomlar yeni bir popülasyon oluşturmak üzere, bilgi (gen) alışverişi yaparak ve mutasyona uğrayarak popülasyondaki daha zayıf bireylerin elenmesini amaçlar. Bu yolla en iyi çözümler ardışık nesiller aracılığıyla gelişir (Parveen & Saha, 2012).

Kısaca genetik algoritma, bir optimizasyon fonksiyonunun, kromozomları temsil eden bit dizileri ya da karakter dizileri, dizelerin genetik operatörler tarafından değiştirilme operasyonları ve soruna bir çözüm bulmak amacıyla kendi uygunluğuna göre seçilmesini içerir. Bu genellikle aşağıdaki prosedür ile yapılır (Yang X.-S. , 2010).

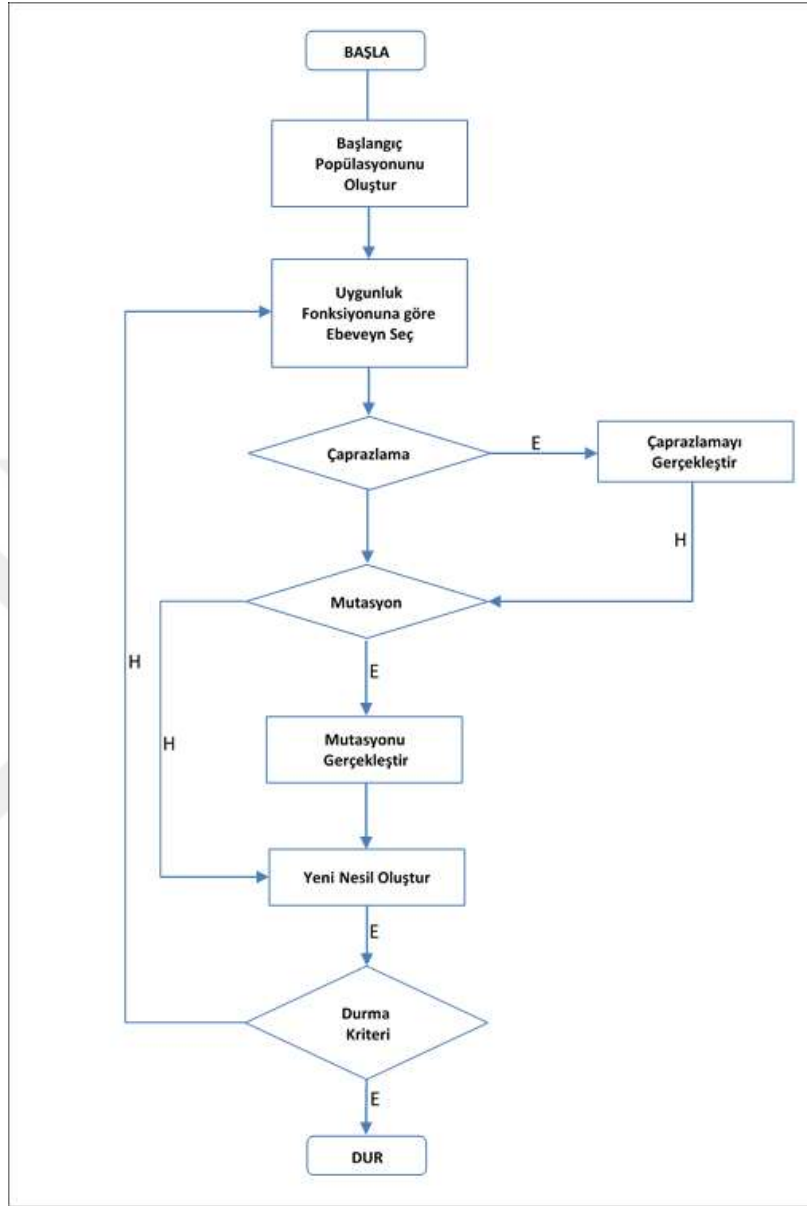
- Amaç veya optimizasyon fonksiyonlarının kodlanması
- Bir uygunluk fonksiyonu veya seçim kriterinin tanımlanması
- Bireylerin popülasyonunun tanımlanması
- Popülasyondaki tüm bireylerin uygunluğunun değerlendirilmesi
- Çaprazlama, mutasyon, elitizm, üreme vb. yöntemlerle yeni bir nüfusun oluşturulması
- Belli durdurma kriterleri sağlanıncaya kadar nüfusun gelişimi
- Sorunun çözümünü elde etmek için sonuçların değerlendirilmesi

Bu adımlar Tablo 2.15'te genetik algoritmaların sözde kodu (pseudo code) ile gösterilmiştir:

Tablo 2.15: Genetik algoritmanın sözde kodu.

-
- > Amaç fonksiyonu $f(x)$, $x = (x_1, \dots, x_n)^T$
 - > Kromozomları oluştur (ikili, permütasyon, gerçel sayılı vb.)
 - > Uygunluk fonksiyonunu tanımla (Genelde bireyin amaç fonksiyonu Karşılığı)
 - > Başlangıç popülasyonunu üret
 - > Çaprazlama(p_c) ve Mutasyon(p_m) oranlarını belirle
 - > while ($t < \text{Max nesil sayısı}$)
 - > Çaprazlama ve mutasyon yoluyla yeni çözümler üret
 - > if $p_c > \text{rassal sayı}$, Çaprazlama Uygula; end if
 - > if $p_m > \text{rand}$, Mutasyon Uygula; end if
 - > Uygunluk değerleri artarsa yeni çözümleri kabul et
 - > Yeni çözüm için mevcut eniyiyi seç (Elitizm varsa) ;
 - > end while
 - > Sonuçları yazdır ve görselleştir.
-

Genetik algoritmanın akış şeması Şekil 2.37’de gösterilmiştir (Haque & Hasin, 2012).



Şekil 2.37: Genetik algoritmanın akış diyagramı.

Genetik algoritmaya ait kavramlar ve operatörlerle ilgili tanımlamalar aşağıda kısaca özetlenmiştir:

Genetik algoritmaya ait kavramlar ve operatörlerle ilgili tanımlamalar aşağıda kısaca özetlenmiştir:

- **Gen:** Bir çözümün parametrelerinin herbiri. Organizmanın karakterini ortaya koyan kalıtsal birimlerdir.

- **Alel:** Genlerin taşıdığı değerdir.
- **Locus:** Kromozomdaki genin pozisyonudur.
- **Kromozom:** Birden fazla genin bir araya gelmesi ile oluşan dizi, olası çözümleri gösteren herbir bireydir.
- **Popülasyon:** Bir problemin olası çözümlerini gösteren kromozomların oluşturduğu topluluktur.
- **Uygunluk değeri:** Herbir kromozomun, uygunluk fonksiyonu kullanılarak (çoğu zaman amaç fonksiyonu) değerlendirilmesi elde edilen performans skorudur.
- **Seçim:** Uygunluk değerine bağlı olarak bir popülasyonda bulunan en başarılı çözümlerin bir seçim algoritması ile üreme işlemi için seçilmesidir.
- **Çaprazlama:** İki farklı çözümün parçalarının rassal şekilde karıştırılarak yeni çözümlerin türetilmesidir.
- **Mutasyon:** Rassal olarak aday bir çözümün gen yapısının değiştirilmesidir.

2.7.1. Gen, Kromozom ve Kodlama Yapısı

Standart bir GA'da aday çözümler (bireyler) eşit boyutlu ikilik düzende (binary) vektörler olarak ifade edilir. Başlangıçta bu vektörlerden bir grup rastlantısal olarak seçilerek belirli büyüklükte bir popülasyon oluşturulur. Kromozom adı verilen vektörler evrimsel mekanizmalar neticesinde yeni popülasyonların gelişimini sağlar. Kromozom üzerindeki genler 'n' boyutlu vektörün bir boyutunu temsil eder (Sarı, 2008).

2.7.2. Popülasyon

GA'nın ilk basamağı başlangıç popülasyonunun üretilmesidir. Popülasyon büyüklüğünün belirlenmesi son derece önemlidir. Büyük popülasyonlar ağır hesaplamalara ihtiyaç duysa da küçük popülasyonlar çözümü ciddi derecede gizleme riski içerir (Bettemir Ö. H., 2010).

2.7.3. Uygunluk Değeri (fi) ve Uygunluk Fonksiyonu

Uygunluk fonksiyonu, dizi şeklindeki giriş değişkenlerinden (kromozomdan) bir çıktı üretir. Uygunluk fonksiyonu matematiksel bir fonksiyon, bir deney veya bir oyun olabilir. Amaç girdiler içerisindeki en uygun değerleri bularak çıktıyı istenilen şekilde modifiye etmektir. GA bir kromozom veya optimize edilecek değişken değerlerinden oluşan bir dizinin tanımlanmasıyla başlar.

Eğer kromozom $p_1, p_2, \dots, p_{N_{\text{değ}}}$ şeklinde $N_{\text{değ}}$ adet değişkene sahipse, kromozom elemanları bir satır vektörü olarak yazılır:

$$\text{kromozom} = [p_1, p_2, p_3, \dots, p_{N_{\text{değ}}}]$$

Örneğin, bir topografik harita üzerinde maksimum yükseklik aramak, boylam (x) ve enlem (y) girdi değişkenleri ile bir uygunluk fonksiyonu gerektirir. Bu durumda $N_{\text{değ}}=2$ olan kromozom aşağıdaki şekilde tanımlanabilir:

$$\text{kromozom} = [x, y]$$

Oluşturulan her kromozomun, uygunluk fonksiyonu, f 'den $p_1, p_2, \dots, p_{N_{\text{değ}}}$ değişkenleriyle elde edilen bir uygunluk değeri vardır.

$$\text{uygunluk değeri} = f(\text{kromozom}) = f(p_1, p_2, \dots, p_{N_{\text{değ}}})$$

2.7.4. Seçim Operatörü

Seçim işlemi, “yetenekli olanın yaşaması” ilkesine uygun olarak çalışmaktadır. Başlangıç toplumundaki kromozomlar arasından, en iyi uygunluk değerine sahip olanlardan bazıları yeni topluma aktarılmakta, bazıları da ebeveyn olarak seçilmektedir. Yeni toplumun diğer kromozomları, ebeveyn olarak seçilen bireylere uygulanan genetik işlemler ile oluşturulmaktadır. Uygunluk değeri en iyi olanların seçilme ve yeni topluma aktarılma şansı daha yüksektir (Paksoy, 2007).

Doğal seleksiyon, çaprazlama öncesinde var olan popülasyon büyüklüğünün, çaprazlama sonucunda da aynı kalmasını sağlar. Bununla birlikte uygunluk değerleri düşük olan bireyleri eleyerek popülasyonun genel gen kalitesini de geliştirir. Öte yandan, düşük uyum gösteren kromozomlar, belirli yerlerinde çok önemli genler taşıyabilir ve bu kısımları korumak için doğal seleksiyon aşamasında bazı önlemler alınır.

2.7.4.1. Rulet Çarkı (Tekerleği) Yöntemi

Holland tarafından bulunmuş bir yöntemdir. Öncelikle her bir kromozomun uygunluk değeri, uygunluk fonksiyonu ile hesaplanır. Uygunluk fonksiyonu çoğu zaman amaç fonksiyonudur. Popülasyondaki tüm uygunluk değerleri toplanarak her bir değer uygunluk değerleri toplamına bölünerek seçilme olasılıkları hesaplanır.

Denklem 2.16'da popülasyondaki bireylerin uygunluk değerleri toplamı gösterilmiştir. Burada n , popülasyon büyüklüğünü ve $f(x_i)$, i . bireyin uygunluk değerini simgeler.

$$F = \sum_{i=1}^n f(x_i) \quad (2.16)$$

Denklem 2.17'de her bir bireye ait seçilme olasılıklarının hesabı gösterilmektedir. Her bir bireye karşılık gelen seçilme olasılığı hesaplanır.

$$P(x_i) = \frac{f(x_i)}{F} \quad (2.17)$$

Olasılık hesabından sonra her bir bireyin olasılık değeri kümülatif şekilde toplanır. Değerlerin toplamı 1 olmaktadır. Denklem 2.18'de birikimli olasılık hesabı gösterilmiştir.

$$q_i = \sum_{j=1}^i P_j \quad (2.18)$$

2.7.4.2. *Elitizm*

Çaprazlama ve mutasyon sırasında uygunluk değeri en yüksek birey bir sonraki popülasyona aktarılamamışsa, belirli bir yüzdeye karşılık gelen sayıda en iyi birey yeni nesildeki en kötü bireylerle yer değiştirilir. Algoritmadan ziyade en iyi çözümlerin korunmasına yönelik bir stratejidir.

2.7.5. **Çaprazlama Operatörü**

Çaprazlama, genetik yenilenme açısından gerekli bir işlemdir. Yeni genler rastgele seçilmiş genlerden çoğaltılır. Çiftler (ebeveynler) rassal sayılar ile belirlenir ve yeni iki gen, çaprazlama işlemi ile ebeveynlerden çoğaltılır. Çaprazlama yeri de rastgele bir sayı üretilerek belirlenir. Çaprazlamadan sonra, popülasyonda var olan iki gen kombinasyonundan, yeni iki gen kombinasyonu türetilmiş olur (Bettemir Ö. H., 2010).

Çaprazlama oranının düşük olması yeterli genetik farklılığın oluşturulamaması ve çözüme yakınsamanın azalmasına neden olduğundan hayati öneme sahiptir. Çok yüksek çaprazlama

oranı ise genleri çok fazla karıştırır ki bu durumda iyi uygunluk gösteren genleri taşıyan kromozomların global optimuma ulaşmasını engeller (Bettemir Ö. H., 2010).

Çaprazlama oranı, popülasyon içindeki herhangi bir bireyin çaprazlanma olasılığıdır. Genellikle 0,6 ile 1,0 arasında değerler alır. Her nesil için çaprazlama olasılığı değişebilir. Çaprazlama olasılığının küçük olması sonuca yavaş ulaşılmasına, büyük olması en iyi çözüm noktasının kaçırılmasına neden olabilir (Goldberg D. E., 1989).

Rulet çarkı veya elitist strateji ile seçilen ebeveynler ikili olarak gruplandırıldıktan sonra her bir ebeveyn çifti için bir rassal sayı üretilir, bu rassal sayılardan hangileri çaprazlama oranından küçük ise o ebeveyn çifti için çaprazlama uygulanır.

Çaprazlama işlemi için, tek noktalı çaprazlama, iki noktalı çaprazlama, çok noktalı çaprazlama ve uniform çaprazlama teknikleri sıklıkla kullanılmaktadır.

2.7.5.1. Tek Noktalı Çaprazlama

Tek noktalı çaprazlamada, rastgele seçilen kromozom çiftinde, çaprazlama yapılacak bölge rastgele seçilerek çaprazlama yapılmaktadır (Paksoy, 2007). N sayıda genden oluşan bir kromozomda, rasgele N-1 sayıdan biri belirlenerek çaprazlama konumu belirlenir ve bu noktanın sağında kalan değerler yer değiştirir.

Rassal olarak belirlenen sayının 3 olması durumunda, Tablo 2.16'da gösterildiği gibi 3. aralıktan sonra gelen genler ebeveynler arasında değiştirilir.

Tablo 2.16: Tek noktalı çaprazlama.

KROMOZOM 1	0	1	1	1	0	1	0
KROMOZOM 2	1	1	0	0	1	1	1

TEK NOKTALI ÇAPRAZLAMA

KROMOZOM 1'	0	1	1	0	1	1	1
KROMOZOM 2'	1	1	0	1	0	1	0

2.7.5.2. İki Noktalı – Çok Noktalı Çaprazlama

Tek noktalı çaprazlamaya benzer, kromozom üzerinde rassal olarak 2 sabit nokta belirlenir ve bu iki nokta arasındaki genler yer değiştirir. Tablo 2.17'de rassal olarak belirlenmiş 3 ve 5. çaprazlama konumları üzerinden çaprazlama gösterilmiştir.

Tablo 2.17: İki noktalı çaprazlama.

KROMOZOM 1	0	1	1	1	0	1	0
KROMOZOM 2	1	1	0	0	1	1	1
İKİ NOKTALI ÇAPRAZLAMA							
KROMOZOM 1'	0	1	0	0	1	1	0
KROMOZOM 2'	1	1	1	1	0	1	1

Çok noktalı çaprazlama da, iki noktalı çaprazlamaya benzer bir şekilde, belirlenen 2 den fazla rassal çaprazlama noktasından gen alışverişi yapılarak uygulanır. İlk bölümde kalan genler sabit olmak üzere sırasıyla her bölümdeki genler arasında gen alışverişi yapılır.

2.7.5.3. Uniform Çaprazlama

Çok noktalı çaprazlamanın özel bir uygulamasıdır. N genlik bir birey için aynı sayıda gene sahip bir maske dizi oluşturulur. Çocuk 1 oluşturulurken maske dizinin ilgili lokusundaki değer 1 ise gen Ebeveyn 1'den, değer 0 ise gen Ebeveyn 2 den alınır. Çocuk 2'nin genleri ise eğer maskede o genin lokusundaki değer 0 ise Ebeveyn 1'den, 1 ise Ebeveyn 2'den gelir (Sakawa, 2002). Tablo 2.18'de ikili kodlama yapısı için uniform çaprazlama gösterilmiştir.

Tablo 2.18: Binary kodlama yapısında, uniform çaprazlama.

EBEVEYN 1	0	1	1	1	0	1	0
EBEVEYN 2	1	1	0	0	1	1	1
MASKE DİZİ	1	1	0	1	0	0	1
ÇOCUK 1	0	1	0	1	1	1	0
ÇOCUK 2	1	1	1	0	0	1	1

Permütasyon tipi kodlama yapısında genlerin her birinden bir adet bulunma zorunluluğundan dolayı (hiçbir şehre iki kez uğramadan en kısa yolun bulunması problemi gibi) uniform çaprazlamada, Çocuk 1 için maske dizide 1 olan genler Ebeveyn 1'den alınır, boş kalan yerlere ise sırasıyla Ebeveyn 2'den kullanılmayan genler yerleştirilir. Çocuk 2 için bunun tam tersi uygulanır. Tablo 2.19'da permütasyon tipi kodlama için uniform çaprazlama uygulaması gösterilmiştir.

Tablo 2.19: Permütasyon tipi kodlamada uniform çaprazlama.

EBEVEYN 1	1	3	2	6	4	5	7
EBEVEYN 2	3	1	7	4	2	6	5
MASKE DİZİ	1	1	0	1	0	1	1
ÇOCUK 1	1	3	7	6	4	5	7
ÇOCUK 2	1	4	7	6	2	4	5

2.7.6. Mutasyon Operatörü

Mutasyon hayatta kalma olasılığı yüksek olan belirli bir genin hâkimiyetini önler. İyi uyum sağlamış olan bazı kromozomlar yerel minimuma takılmış olabilir. Bu durumda daha iyi çözümler gözden kaçır. Mutasyon operatörü muhtemel yerel optimum değerlerine takılmayı önler. Yüksek mutasyon oranı ise global optimuma giden iyi uyum gösteren genlerin bozulmasına yol açabilir (Bettemir Ö. H., 2010).

2.7.7. Durdurma Kriteri

Optimizasyon algoritmalarında aşağıdaki durumlar durdurma kriteri olarak uygulanabilir:

- Max. iterasyon (nesil) sayısı,
- Belirlenen amaç fonksiyonu değeri,
- Yakınsama ya da çözümün gelişmediği ardışık iterasyon sayısı,
- Belirlenen bir zaman

3. MALZEME VE YÖNTEM

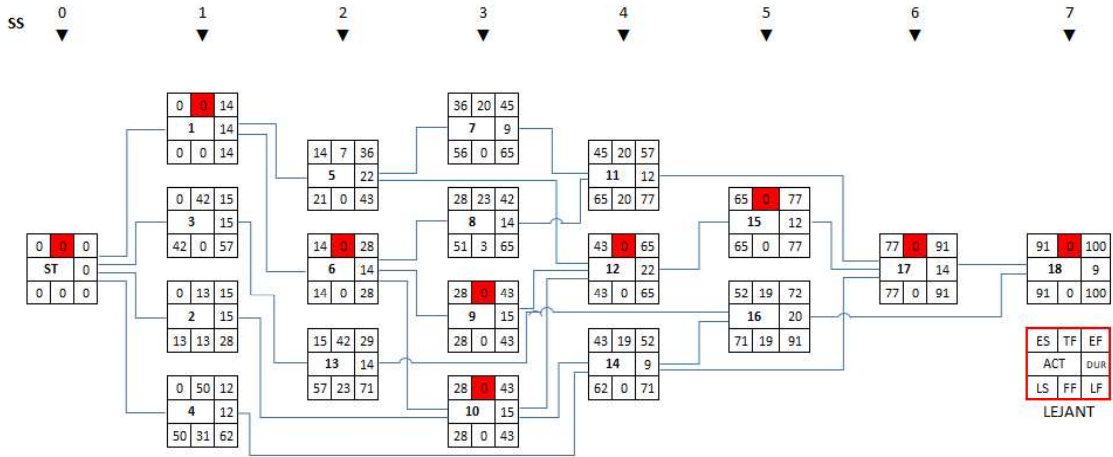
3.1. PROBLEM TANIMI

Bu çalışmada öncelikle Feng ve diğ. (1997) ile Hegazy (1999)'nin çalışmalarından türetilen çok modlu test problemine yönelik algoritma geliştirilmiştir. Problem çözümü için yazılan Matlab kodu öncelikle test probleminde kullanılmış ardından gerçek bir projeye ait veriler değerlendirilmiştir. Probleme ait aktivite numaraları, hemen önceki faaliyet listeleri ve her bir faaliyet numarası için maksimum beş mod içeren farklı süre-maliyet değerleri ile AoN gösterimi için kullanılan sıralama sayıları da Tablo 3.1'de verilmiştir.

Tablo 3.1: 18 faaliyetli test problemi verileri.

AKTİVİTE	HÖF	1. MOD		2. MOD		3. MOD		4. MOD		5. MOD		SS
		SÜRE	MALİYET	SÜRE	MALİYET	SÜRE	MALİYET	SÜRE	MALİYET	SÜRE	MALİYET	
1	-	14	2400	15	2150	16	1900	21	1500	24	1200	1
2	-	15	3000	18	2400	20	1800	23	1500	25	1000	1
3	-	15	4500	22	4000	33	3200	-	-	-	-	1
4	-	12	45000	16	35000	20	30000	-	-	-	-	1
5	1	22	20000	24	17500	28	15000	30	10000	-	-	2
6	1	14	40000	18	32000	24	18000	-	-	-	-	2
7	5	9	30000	15	24000	18	22000	-	-	-	-	3
8	6	14	220	15	215	16	200	21	208	24	120	3
9	6	15	300	18	240	20	180	23	150	25	100	3
10	2, 6	15	450	22	400	33	320	-	-	-	-	3
11	7, 8	12	450	16	350	20	300	-	-	-	-	4
12	5, 9, 10	22	2000	24	1750	28	1500	30	1000	-	-	4
13	3	14	4000	18	3200	24	1800	-	-	-	-	2
14	4, 10	9	3000	15	2400	18	2200	-	-	-	-	4
15	12	12	4500	16	3500	-	-	-	-	-	-	5
16	13, 14	20	3000	22	2000	24	1750	28	1500	30	1000	5
17	11, 14, 15	14	4000	18	3200	24	1800	-	-	-	-	6
18	16, 17	9	3000	15	2400	18	2200	-	-	-	-	7

Tablo 3.1'de verilen bilgiler ışığında problemin CPM şebekesi çizilerek Şekil 3.1'de verilmiştir. Şebeke daha kolay anlaşılması için kutu tipi diyagram olarak verilmiştir ve literatürde çok sık karşılaşılmayan bir düğüm gösterimi ile sunulmuştur. Örnek olması açısından 5 milyar çözüm arasından sadece 1. modlar kullanılarak elde edilen şebekenin diyagramı çizilmiştir.



Şekil 3.1: 18 faaliyetli test probleminin AoN tipi şebeke diyagramı.

Her bir aktiviteye karşılık rasgele birer mod seçildiğinde farklı süre ve maliyet bileşenleri ortaya çıkmaktadır. Problemdeki modlar göz önüne alındığında 5.904.900.000 olası sonuç arasından hangi sonuçların optimum olduğuna ilişkin bir çözüm deterministik metotlarla kabul edilebilir süre sınırları dahilinde üretilememektedir. Problem çözme süresini üstel çözüm zamanından polinomial çözüm zamanına indirmek adına sezgisel çözüm algoritmalarının kullanılması neredeyse bir zorunluluk haline almıştır.

Çözümü için polinomial zaman fonksiyonu bulunabilen problemler p (polinomial) tipi, çözümü için polinomial zaman fonksiyonu henüz bulunamamış ama bulunamayacak anlamına gelmeyen problemler ise Np (non-polinomial) tipi problemler olarak tanımlanır. NP-Zor tipi problemler ise polinomial zamana bağlı çözüm algoritması olmayan, deterministik olmayan zor problemlerdir ve çözüm için meta sezgisel metotlar kullanılır. Gerçek hayat problemleri genellikle bu sınıfa dahil olmaktadır. Çok karar değişkeni, süresiz (kombinatorial) veri setleri, bağımlı değişkenler, birden fazla tepe noktası, çok kısıt olması gibi durumlar problemin zorluk derecesini artırmaktadır.

Bu çalışmada kullanılan test problemine ait veri seti ve zaman maliyet optimizasyonu problemi, çok amaçlı, kaynak kısıtlı, kesikli, nonlinear, çok modlu ve stokastik olma gibi özellikler içerdiğinden NP-Zor tipi optimizasyon problemi sınıfına girmektedir.

18 faaliyetli çok modlu, çok amaçlı problemin çözümü için Evrimsel tabanlı meta sezgisel metotlardan (EAs) en bilinen ve literatürde sıklıkla kullanılan Genetik Algoritma kullanılmıştır.

Zaman- maliyet optimizasyon probleminde hem zaman, hem de maliyetin eşit derecede önemli olması çok amaçlı optimizasyon problemini doğurmaktadır. Çok amaçlı optimizasyon problemlerinde tek bir optimum çözüm elde etmek mümkün değildir. Proje yöneticisi, projede ortaya çıkan değişiklikler ve çevresel faktörlerden dolayı, projeyi farklı sürelerde bitirmek durumunda kalabilir. Bu durumda her bir süre değeri için minimum maliyetlerin tespit edilmesi bir zorunluluk haline gelmektedir. Yine farklı maliyet durumları için projenin bitebileceği minimum sürelerin tespiti de önemlidir.

Böyle karmaşık bir problem yapısında projenin 100 gün ve 100.000TL maliyetle bitmesi 120 gün ve 80.000 TL maliyetle bitmesinden üstün değildir. Her ikisi de farklı durumlarda proje yöneticisi tarafından seçilebilecek pareto optimal değerleridir, birbirlerine bir üstünlük sağlayamamaktadırlar. Oysa ki projenin 100 gün ve 100.000TL maliyetle bitmesi 100 gün ve 120.000TL maliyetle bitmesinden daha iyi bir sonuçtur ve yakın optimum veya optimum olmaya aday bir çözümdür. Farklı bir durumda projenin 100 gün ve 100.000 TL maliyetle bitmesi 80 gün ve 100.000 TL maliyetle bitmesi durumundan daha kötü bir sonuçtur ve aynı maliyet değerleri için optimum olması beklenemez.

Farklı maliyet ve süre bileşenlerinin bulunduğu çözüm uzayında, pareto değerlerinin elde edilmesini sağlayan algoritma yaklaşık 6 milyar olası süre maliyet çifti arasından birbirine baskın olmayan pareto cephesini elde etmeyi amaçlamaktadır.

Problemin kesikli çözüm uzayında her farklı mod bileşimi için bir proje süresi ve maliyeti ortaya çıkmaktadır. Tablo 3.2’de modlara bağlı kombinasyonlar hesaplandığında yaklaşık 6 milyar olası çözümün olduğu görülmektedir.

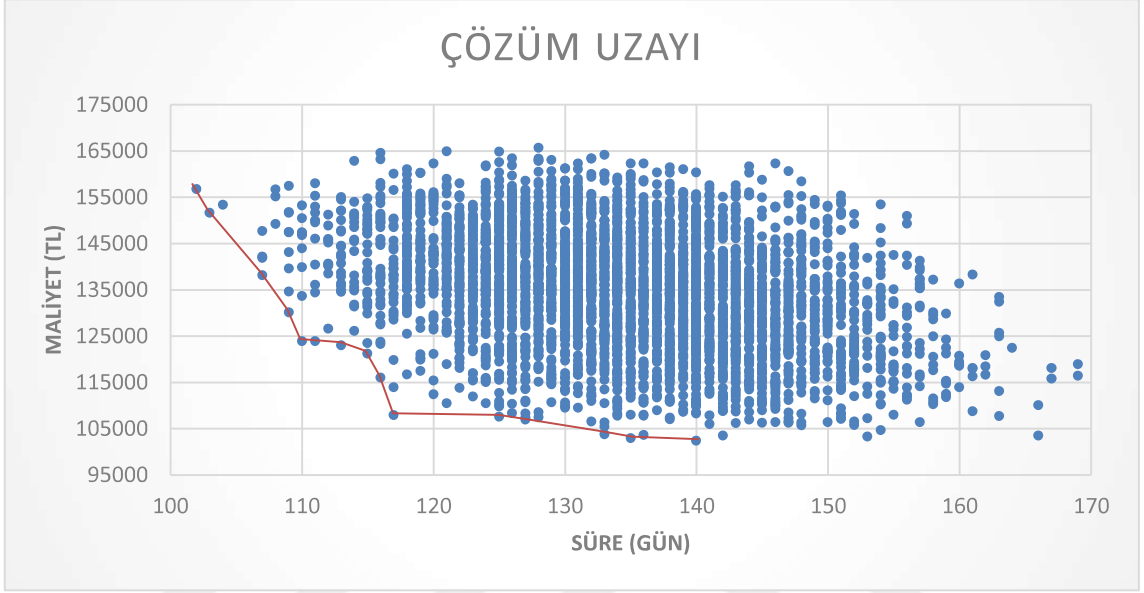
Tablo 3.2: Olası mod bileşimi ve çözüm sayısı.

ACT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	OLASI ÇÖZÜM
MOD	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3	5.904.900.000

Altı milyar çözüm arasından hangi mod bileşenlerinin en iyi çözümü verdiğini bulmak için deterministik yöntemlerin yetersiz kaldığı belirtilmişti. Zaman-maliyet ödünleşim probleminde süre ve maliyeti en küçükleme istediğimizde karşımıza çok amaçlı optimizasyon modeli ortaya çıkar.

İncelenen problemde 100 günden 170 güne kadar her bir güne karşılık yüzlerce maliyet karşımıza çıkmaktadır. Bu maliyetlerin minimumunu ve aynı zamanda minimum maliyetler arasından da birbirine baskın gelemeyen (nondominated) çözümlerin bulunması gerekmektedir.

Şekil 3.2’de altı milyar çözümden sadece 2000 adedi ile çizilmiş grafik çözüm uzayı için fikir vermesi açısından önemlidir. Kırmızı çizgi ile belirtilen çözümlerden her biri bir diğerine baskın gelememekte (yani herhangi birine optimum çözüm demek mümkün değildir) ve pareto cephesi denen nondominated çözümleri içermektedir.



Şekil 3.2: 2000 adet çözüm içeren olası çözüm uzayı.

Genetik algoritma ile amaçlanan, öncelikle çözüm uzayı içerisinde belirlenen popülasyon sayısına göre dağılmış zaman maliyet bileşenlerini MAWA ile değerlendirip en iyilerini seçmektir. Ardından seçilen bireylerin birbirleri arasında gen alışverişi yapmaları için çaprazlama, çeşitliliğin sağlanması ve yerel minimumlara takılmamak için de mutasyon gerçekleştirilerek önceki nesle göre daha iyi nesiller elde etmeye çalışılır. Doğru başlangıç popülasyonu, iterasyon sayısı ve GA parametreleri ile kabul edilebilir çözüm sürelerinde istenen pareto değerlerini elde etmek mümkündür.

3.2. PROBLEM ÇÖZÜMÜ

Bu bölümde, bölüm 3.1.’de ayrıntılı olarak anlatılan, çok amaçlı, çok modlu zaman maliyet ödünleşim problemi optimizasyonu yapılmıştır. Genetik algoritma ile yapılan çözümlenmelerden sonra bir Modifiye Genetik Algoritma önerisi yapılarak bulunan sonuçlar karşılaştırılmıştır.

Yapılan analizlerin daha iyi anlaşılabilmesi için küçük bir başlangıç popülasyonu ile algoritmanın çalışma prensibi ve matlab kodları verilmiştir. Analiz sırasında kullanılan GA parametre ve değişkenleri Tablo 3.3’de verilmiştir:

Tablo 3.3: Örnek çözüm için genetik algoritma parametreleri.

GA PARAMETRELERİ	
Çaprazlama oranı (Pc)	0,75
Mutasyon oranı (Pm)	0,05
Başlangıç Popülasyonu (Psize)	6
Run sayısı	1
İterasyon sayısı (It)	2

Çaprazlama oranı literatürde $0,70 - 1$, mutasyon oranı ise $0,01 - 0,1$ arasında değerler almaktadır. Çaprazlama stratejilerinden genellikle, iki noktalı, tek noktalı, üniform çaprazlama kullanılmaktadır. Seçim kuralı olarak ise en yaygın kullanım rulet çemberi tekniğidir. Örnek çözümleme için çaprazlama oranı 0.75, Mutasyon oranı 0.05 olarak literatürde sıklıkla karşılaşılan değerler seçilmiştir. Genetik algoritma mekanizmasının daha iyi anlaşılabilmesi için başlangıç popülasyonu 6 bireyden oluşan bir nesil olarak belirlenmiştir. Çalışma iki iterasyon ilerletilerek hem GA hemde önerilen MGA modelinin pareto çıktıları üzerinden değerlendirmeler yapılmıştır.

GA mekanizması anlatıldıktan sonra başlangıç popülasyonunun ve iterasyon sayısının genetik algoritma üzerine etkisi incelenerek asıl analizler için uygun olabilecek başlangıç popülasyonu sayısı ve iterasyon sayısı seçilmiştir.

3.2.1. Başlangıç Popülasyonunun Oluşturulması

Başlangıç popülasyonu metasezgisel algoritmalarla yapılan çözümlemelerde genellikle ya random olarak seçilir, ya da başka sezgisellerin çıktıları kullanılarak çözüme başlanır. İncelenen zaman-maliyet optimizasyonu probleminde başlangıç popülasyonu rassal olarak üretilecektir.

Tablo 3.4: Faaliyetlerin opsiyon bileşenleri ve rassal başlangıç popülasyonu kodu.

ACT	IPA	MOD 1		MOD 2		MOD 3		MOD 4		MOD 5		MOD SAYISI
		DUR	COST	DUR	COST	DUR	COST	DUR	COST	DUR	COST	
1	-	14	2400	15	2150	16	1900	21	1500	24	1200	5
2	-	15	3000	18	2400	20	1800	23	1500	25	1000	5
3	-	15	4500	22	4000	33	3200	-	-	-	-	3
4	-	12	45000	16	35000	20	30000	-	-	-	-	3
5	1	22	20000	24	17500	28	15000	30	10000	-	-	4
6	1	14	40000	18	32000	24	18000	-	-	-	-	3
7	5	9	30000	15	24000	18	22000	-	-	-	-	3
8	6	14	220	15	215	16	200	21	208	24	120	5
9	6	15	300	18	240	20	180	23	150	25	100	5
10	2, 6	15	450	22	400	33	320	-	-	-	-	3
11	7, 8	12	450	16	350	20	300	-	-	-	-	3
12	5, 9, 10	22	2000	24	1750	28	1500	30	1000	-	-	4
13	3	14	4000	18	3200	24	1800	-	-	-	-	3
14	4, 10	9	3000	15	2400	18	2200	-	-	-	-	3
15	12	12	4500	16	3500	-	-	-	-	-	-	2
16	13, 14	20	3000	22	2000	24	1750	28	1500	30	1000	5
17	11, 14, 15	14	4000	18	3200	24	1800	-	-	-	-	3
18	16, 17	9	3000	15	2400	18	2200	-	-	-	-	3

%%

%RASSAL BAŞLANGIÇ POPÜLASYONUNUN OPSİYON GENLİKLERİ

range(1)=5; range(10)=3;

range(2)=5; range(11)=3;

range(3)=3; range(12)=4;

range(4)=3; range(13)=3;

range(5)=4; range(14)=3;

range(6)=3; range(15)=2;

range(7)=3; range(16)=5;

range(8)=5; range(17)=3;

range(9)=5; range(18)=3;

%RASSAL BAŞLANGIÇ POPÜLASYONUNUN OLUŞTURULMASI

for j=1:psize

for i=1:18

pop(j,i)=randi([1,range(i)],1,1);

end

end

%%

Tablo 3.4'te verilen opsiyon bileşenlerine göre matlab kodunda opsiyonlar için genlikler belirlenmiştir. Bu genlikler arasında kalmak kaydıyla rassal olarak her bir faaliyete karşılık bir mod seçilmesi sağlanmıştır. Tablo 3.5'te üretilen rassal başlangıç popülasyonu verilmiştir.

Tablo 3.5: Rassal başlangıç popülasyonu ve 1. bireye karşılık gelen opsiyonlar.

ACT NO	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
E1	2	5	3	3	3	3	1	3	5	1	1	1	1	2	1	4	2	3
E2	2	5	3	1	3	2	3	4	5	3	3	2	3	3	1	3	3	2
E3	1	5	1	3	2	3	3	3	1	2	2	3	2	3	1	1	1	3
E4	2	1	3	2	4	3	1	5	5	3	3	4	3	3	2	5	1	3
E5	3	4	3	1	2	1	2	4	1	1	3	4	3	1	1	5	1	3
E6	5	1	3	2	1	3	1	3	4	3	3	2	3	2	2	4	1	3

KROMOZOM 1	ACT	HÖF	1. MOD		2. MOD		3. MOD		4. MOD		5. MOD	
			DUR	COST	DUR	COST	DUR	COST	DUR	COST	DUR	COST
	1	-	14	2400	15	2150	16	1900	21	1500	24	1200
	2	-	15	3000	18	2400	20	1800	23	1500	25	1000
	3	-	15	4500	22	4000	33	3200	-	-	-	-
	4	-	12	45000	16	35000	20	30000	-	-	-	-
	5	1	22	20000	24	17500	28	15000	30	10000	-	-
	6	1	14	40000	18	32000	24	18000	-	-	-	-
	7	5	9	30000	15	24000	18	22000	-	-	-	-
	8	6	14	220	15	215	16	200	21	208	24	120
	9	6	15	300	18	240	20	180	23	150	25	100
	10	2, 6	15	450	22	400	33	320	-	-	-	-
	11	7, 8	12	450	16	350	20	300	-	-	-	-
	12	5, 9, 10	22	2000	24	1750	28	1500	30	1000	-	-
	13	3	14	4000	18	3200	24	1800	-	-	-	-
	14	4, 10	9	3000	15	2400	18	2200	-	-	-	-
	15	12	12	4500	16	3500	-	-	-	-	-	-
	16	13, 14	20	3000	22	2000	24	1750	28	1500	30	1000
	17	11, 14, 15	14	4000	18	3200	24	1800	-	-	-	-
	18	16, 17	9	3000	15	2400	18	2200	-	-	-	-

Tablo 3.5'te görüldüğü üzere her bir kromozom permütasyon tipi kodlama gibi görünse de aslında seçilen moda karşılık 1, diğer modlara 0 yazıldığında binary kodlamaya karşılık gelen bir kodlama yapısına sahiptir. CPM hesabının kolaylaşması açısından kromozomlar vektör olarak tanımlanarak popülasyon matrisi elde edilmiştir.

Aynı mantıkla rassal başlangıç popülasyonundaki her bir gene karşılık gelen maliyet ve süre matrisleri de ayrı ayrı oluşturulmaktadır. Maliyet matrisi Tablo 3.5'te, süre matrisi Tablo 3.6'da gösterilmiştir.

Tablo 3.6: Bireylerin genlerine karşılık gelen maliyetler.

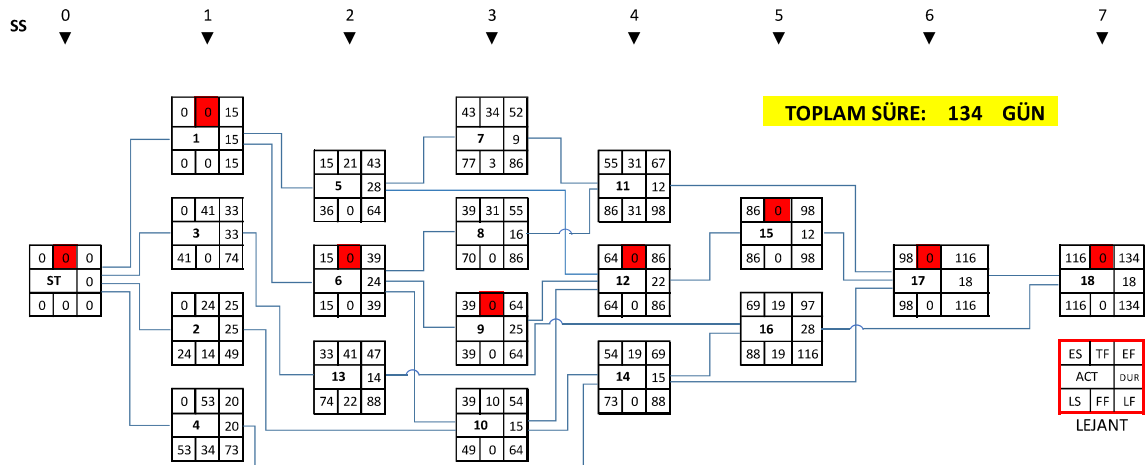
cost: KROMOZOMLARA AİT MODLARA KARŞILIK GELEN MALİYET (TL) MATRİSİ																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
E1	2150	1000	3200	30000	15000	18000	30000	200	100	450	450	2000	4000	2400	4500	1500	3200	2200
E2	2150	1000	3200	45000	15000	32000	22000	208	100	320	300	1750	1800	2200	4500	1750	1800	2400
E3	2400	1000	4500	30000	17500	18000	22000	200	300	400	350	1500	3200	2200	4500	3000	4000	2200
E4	2150	3000	3200	35000	10000	18000	30000	120	100	320	300	1000	1800	2200	3500	1000	4000	2200
E5	1900	1500	3200	45000	17500	40000	24000	208	300	450	300	1000	1800	3000	4500	1000	4000	2200
E6	1200	3000	3200	35000	20000	18000	30000	200	150	320	300	1750	1800	2400	3500	1500	4000	2200

Tablo 3.7: Bireylerin genlerine karşılık gelen süreler.

dur: KROMOZOMLARA AİT MODLARA KARŞILIK GELEN SÜRE (GÜN) MATRİSİ																		
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
E1	15	25	33	20	28	24	9	16	25	15	12	22	14	15	12	28	18	18
E2	15	25	33	12	28	18	18	21	25	33	20	24	24	18	12	24	24	15
E3	14	25	15	20	24	24	18	16	15	22	16	28	18	18	12	20	14	18
E4	15	15	33	16	30	24	9	24	25	33	20	30	24	18	16	30	14	18
E5	16	23	33	12	24	14	15	21	15	15	20	30	24	9	12	30	14	18
E6	24	15	33	16	22	24	9	16	23	33	20	24	24	15	16	28	14	18

3.2.2. CPM Hesaplamaları

Başlangıç popülasyonunun oluşturulmasının ardından, Psize kadar kromozomun her biri belirli mod bileşenlerine sahip birer proje olarak karşımıza çıkar. Her bir projenin kendine özgü toplam süre ve toplam maliyet değerleri vardır. Şekil 3.3'te E1 kromozomuna ait şebeke diyagramı görsel açıdan fikir vermesi amacıyla gösterilmiştir.

**Şekil 3.3:** Kromozom 1'e ait CPM şebekesi diyagramı.

Her bir kromozom için CPM hesaplamaları probleme özel yazılan statik matlab kodu ile hesaplanmaktadır.

Tablo 3.8: Bireylerin toplam süre ve toplam maliyet hesabı ile matlab kodu.

ACT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	CP	TC
E1	2	5	3	3	3	3	1	3	5	1	1	1	1	2	1	4	2	3	134	120350
E2	2	5	3	1	3	2	3	4	5	3	3	2	3	3	1	3	3	2	141	137478
E3	1	5	1	3	2	3	3	3	1	2	2	3	2	3	1	1	1	3	132	117250
E4	2	1	3	2	4	3	1	5	5	3	3	4	3	3	2	5	1	3	150	117890
E5	3	4	3	1	2	1	2	4	1	1	3	4	3	1	1	5	1	3	119	151858
E6	5	1	3	2	1	3	1	3	4	3	3	2	3	2	2	4	1	3	153	128520

```

%Random değerlere karşılık gelen cost ve dur değerleri
for j=1:18
    for i=1:psize
        cost(i,j)= c(j, pop(i,j));
        dur(i,j)=s(j, pop(i,j));
    end
end

%% CPM HESAPLAMALARI

for i= 1:psize
    es(i,1)=0 ; ef(i,1)=dur(i,1) ;
    es(i,2)=0 ; ef(i,2)=dur(i,2) ;
    es(i,3)=0 ; ef(i,3)=dur(i,3) ;
    es(i,4)=0 ; ef(i,4)=dur(i,4) ;
    es(i,5)=ef(i,1) ; ef(i,5)=es(i,5)+dur(i,5) ;
    es(i,6)=ef(i,1) ; ef(i,6)=es(i,6)+dur(i,6) ;
    es(i,13)=ef(i,3) ; ef(i,13)=es(i,13)+dur(i,13) ;
    es(i,7)=ef(i,5) ; ef(i,7)=es(i,7)+dur(i,7) ;
    es(i,8)=ef(i,6) ; ef(i,8)=es(i,8)+dur(i,8) ;
    es(i,9)=ef(i,6) ; ef(i,9)=es(i,9)+dur(i,9) ;
    es(i,10)=max(ef(i,6),ef(i,2)) ; ef(i,10)=es(i,10)+dur(i,10) ;
    es(i,11)=max(ef(i,8),ef(i,7)) ; ef(i,11)=es(i,11)+dur(i,11) ;
    ab=[ef(i,5) ef(i,9) ef(i,10)] ; es(i,12)=max(ab) ;
    ef(i,12)=es(i,12)+dur(i,12) ;
    es(i,14)=max(ef(i,4),ef(i,10)) ; ef(i,14)=es(i,14)+dur(i,14) ;
    es(i,15)=ef(i,12) ; ef(i,15)=es(i,15)+dur(i,15) ;
    es(i,16)=max(ef(i,13),ef(i,14)) ; ef(i,16)=es(i,16)+dur(i,16) ;
    bc=[ef(i,11) ef(i,14) ef(i,15)] ; es(i,17)=max(bc) ;
    ef(i,17)=es(i,17)+dur(i,17) ;
    es(i,18)=max(ef(i,16),ef(i,17)) ; ef(i,18)=es(i,18)+dur(i,18) ;

    CP(i,1)=ef(i,18) ;
    TC(i,1)=sum(cost(i,:));

```

Tablo 3.8’de verilen matlab kodu her bir kromozomun oluşturduğu projelere ait toplam proje süresi (CP) ve toplam maliyet (TC) değerlerini hesaplamaktadır. Bir sonraki adımda en iyi bireylerin seçilmesi işlemi yapılacaktır.

3.2.3. Seçim Operatörü

Toplam süre ve toplam maliyet değerleri hesaplanan popülasyonda en iyi bireylerin seçimi sadece minimum süre veya sadece minimum maliyet değerlerine göre yapılması problemi tek amaçlı probleme döndürmesinin yanı sıra bazı iyi çözümlerin kaybedilmesine sebep olabilir.

Seçim işlemi süre ve maliyet değerlerine göre ayrı ayrı yapılamayacağından bu iki değeri de aynı anda temsil eden tek bir değer elde etmek gerekmektedir. Bölüm 2.5.1’de gösterilen modifiye edilmiş ağırlık yaklaşımı ile mevcut popülasyonun maksimum ve minimum süre ve maliyet değerleri kullanılarak ağırlıklar bulunup tek bir uygunluk fonksiyonu değeri elde edilmiştir.

Tablo 3.9: Ebeveynlerin uygunluk değerleri tablosu.

ACT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	CP	TC	UYGUNLUK
(E1)	2	5	3	3	3	3	1	3	5	1	1	1	1	2	1	4	2	3	134	120350	0,73
(E2)	2	5	3	1	3	2	3	4	5	3	3	2	3	3	1	3	3	2	141	137478	0,38
(E3)	1	5	1	3	2	3	3	3	1	2	2	3	2	3	1	1	1	3	132	117250	0,80
(E4)	2	1	3	2	4	3	1	5	5	3	3	4	3	3	2	5	1	3	150	117890	0,53
(E5)	3	4	3	1	2	1	2	4	1	1	3	4	3	1	1	5	1	3	119	151858	0,50
(E6)	5	1	3	2	1	3	1	3	4	3	3	2	3	2	2	4	1	3	153	128520	0,33

Tablo 3.9’da gösterilen uygunluk değerleri denklem 2.15’deki uygunluk fonksiyonu ve MAWA yönteminden elde edilen ağırlıklar kullanılarak hesaplanmıştır. E1 bireyi için uygunluk değeri hesaplanacak olursa, Tablo 3.9’daki popülasyon için zaman değerleri ve maliyet değerleri birbirinden farklı olduğundan ($Z_t^{\max} \neq Z_t^{\min}$ ve $Z_c^{\max} \neq Z_c^{\min}$) MAWA yöntemindeki 1. Koşul doğrulanacaktır. Denklem 2.6 – 2.11 kullanılarak w_c ve w_t ağırlıkları aşağıdaki gibi bulunur.

$$v_c = \frac{Z_c^{\min}}{Z_c^{\max} - Z_c^{\min}} = \frac{117250}{151858 - 117250} = 3,3879$$

$$v_t = \frac{Z_t^{\min}}{Z_t^{\max} - Z_t^{\min}} = \frac{119}{153 - 119} = 3,5$$

$$v = v_c + v_t = 3,3879 + 3,5 = 6,8879$$

$$w_c = \frac{v_c}{v} = \frac{3,3879}{6,8879} = 0,49$$

$$w_t = \frac{v_t}{v} = \frac{3,5}{6,8879} = 0,51$$

Bu ağırlıklar denklem 2.15’te yerine yazılarak her bir birey için uygunluk değerleri bulunabilir.

$$f(E_1) = 0,51 \frac{(153 - 134) + 0,106}{(153 - 119) + 0,446} + 0,49 \frac{(151858 - 120350) + 0,048}{(151858 - 117250) + 0,137} = 0,73$$

Bu yaklaşım herhangi bir ek parametre kullanmadan, tek seferde çoklu aramayı amaçlamaktadır. Bu yaklaşımın temel amacı basit bir uygulama olması ve uygunluk atamasında tek bir amaç kullanıldığı için GA'da minimum modifikasyon gerektirmesidir. Bunun yanında hesap açısından oldukça verimlidir. Bu yaklaşımın ana dezavantajı, Pareto cephesinin konveks olmadığı durumlarda tüm Pareto-optimal çözümlerinin araştırılmamasıdır (Konak, Coit, & Smith, 2006). Tablo 3.9'da E5 ebeveyninin pareto çözümü olmasına rağmen diğer bireylere göre belirgin yüksek bir uygunluk değeri almaması bundan kaynaklanmaktadır.

Bu çalışma kapsamında önerilen modifiye GA yaklaşımında ağırlıklandırma yönteminden kaynaklanan gözden kaçabilecek pareto değerleri, önceki jenerasyonların da pareto hesabına katılması ile sisteme dâhil edilerek, elitist bir yaklaşım sağlanmıştır.

Genetik algoritma yaklaşımında seçim operatörü kullanılırken, yüksek uygunluk değerine sahip bireylerin bir sonraki jenerasyona aktarılması ve bu yolla iyi çözümlerden daha iyi çözümler elde edebilme ihtimali üzerinde durulmaktadır.

Seçim işlemi yapılırken iyi çözüm veren bireylere yüksek yüzdelik dilimler atayan rulet çemberi yöntemi literatürde sıklıkla kullanılmıştır. Bu çalışma kapsamında da rulet çemberi tekniği kullanılarak seçim yaptırılmıştır.

Tablo 3.10: Seçim için uygunluk değerlerinin yüzde cinsinden ifadesi.

BİREY	CP	TC	UYGUNLUK f(i)	Probs g(i)	CProbs
(E1)	134	120350	0,73	0,22	0,22
(E2)	141	137478	0,38	0,12	0,34
(E3)	132	117250	0,80	0,24	0,58
(E4)	150	117890	0,53	0,16	0,74
(E5)	119	151858	0,50	0,15	0,90
(E6)	153	128520	0,33	0,10	1,00
TOPLAM			3,28		

```
%Uygunluk değerlerinin yüzde cinsinden ifadesi
probs=fitness/(sum(fitness)) ;
```

```
%Uygunluk değerlerinin kümülatif toplamları
cprobs=probs;
for k=2:psize
cprobs(k)=cprobs(k-1)+probs(k) ;
end
```

Tablo 3.10'da gösterilen her bir uygunluk değeri, denklem 3.2'de verildiği gibi toplam uygunluk değerine bölünerek yüzde cinsinden ifade edilmektedir.

$$g(i) = \frac{f(i)}{\sum_1^n f(i)} \quad (3.2)$$

Hesaplanan yüzdeler kümülatif olarak toplandığında her bir ebeveyn için bir olasılık aralığı (range) oluşmaktadır.

Tablo 3.11: Ebeveynlerin seçim için olasılık aralıkları ve seçim operatörü kodu.

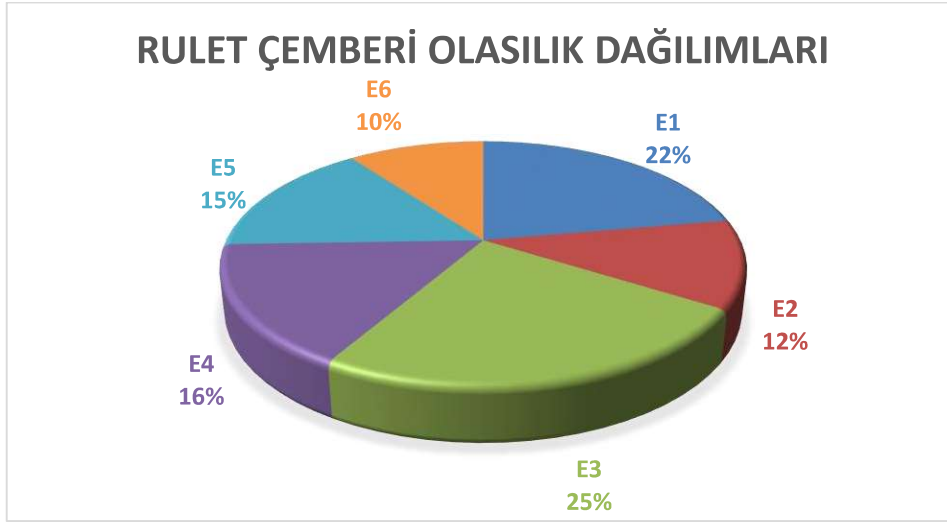
BİREY	CP	TC	FITNESS	Probs	CProbs	Range	RS
(E1)	134	120350	0,73	0,22	0,22	0 - 0,22	0,26
(E2)	141	137478	0,38	0,12	0,34	0,22 - 0,34	0,38
(E3)	132	117250	0,80	0,24	0,58	0,34 - 0,58	0,2
(E4)	150	117890	0,53	0,16	0,74	0,58 - 0,74	0,45
(E5)	119	151858	0,50	0,15	0,90	0,74 - 0,90	0,08
(E6)	153	128520	0,33	0,10	1,00	0,90 - 1,00	0,26

%ÇAPRAZLAMA İÇİN EN YÜKSEK OLASILIKLI BİREYLERİN SEÇİLMESİ

```
for i=1:psize
    jdx=find(rs(i)<cprobs,1);
    arapop(i,:)=pop(jdx,:);
end
```

Rulet çemberi tekniğinde, normal dağılıma uygun rassal sayılar üretilerek bu rassal sayıların denk geldiği olasılık aralığına sahip bireylerin seçilmesi esastır. Şekil 3.3'te bireylerin olasılık aralıkları grafik üzerinde gösterilmiştir. Üretilen 0-1 arasında normal dağılımlı rassal bir sayının yüksek olasılık aralığına sahip bir bireyin olasılık aralığına denk gelme şansı daha yüksektir. İlk bakışta kromozom 3, kromozom 4 ve kromozom 1'in seçilme olasılığının diğerlerine göre daha yüksek olduğu anlaşılmaktadır.

Genetik algoritmanın çalışma prensipleri arasında çeşitliliğin sağlanması son derece önemlidir. Çeşitlilik algoritmanın yerel optimum noktalarına takılmasını engeller. Seçim operatörü kullanılırken yüksek uygunluk değerine sahip bireylerin bir sonraki jenerasyona aktarılma olasılığının yüksek olması, düşük olasılıklı bireylerin seçilemeyeceği anlamına gelmez. Üretilen rassal sayılar ihtimal az da olsa düşük uygunluk değerli bireylerin seçilmesine imkan tanır. Bu nedenle jenerasyonlar boyunca sadece iyi bireylerin çaprazlanması sonucu elde edilen jenerasyonlar değil, kötü çözümlerle elde edilen yeni bireylerin de incelenerek çeşitliliğin artırılması sağlanmış olur.



Şekil 3.4: Rulet çemberi tekniğinde bireylerin olasılık dağılımları.

Tablo 3.11’de üretilen rassal sayılar ve bu rassal sayıların denk geldiği olasılık aralıkları verilmiştir. Tablo 3.12’de gösterilen seçilmiş ebeveynler çaprazlama operatörü için bir sonraki aşamaya taşınmışlardır.

Tablo 3.12: Çaprazlama için seçilen ebeveynler.

RS	SEÇİLEN BİREYLER	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0,26	E2	2	5	3	1	3	2	3	4	5	3	3	2	3	3	1	3	3	2
0,38	E3	1	5	1	3	2	3	3	3	1	2	2	3	2	3	1	1	1	3
0,2	E1	2	5	3	3	3	3	1	3	5	1	1	1	1	2	1	4	2	3
0,45	E3	1	5	1	3	2	3	3	3	1	2	2	3	2	3	1	1	1	3
0,08	E1	2	5	3	3	3	3	1	3	5	1	1	1	1	2	1	4	2	3
0,26	E2	2	5	3	1	3	2	3	4	5	3	3	2	3	3	1	3	3	2

3.2.4. Çaprazlama Operatörü

Çaprazlama, evrimsel süreçte iyi olanın hayatta kalması ve iyi bireylerden iyi soylar elde edilmesi prensibine dayanır. Seçim operatörü ile belirlenen ebeveynler arasında gen alışverişi yapılarak daha yüksek uygunluk değerine sahip yeni jenerasyonlar üretilmesi amaçlanmıştır.

Çaprazlamanın tek noktalı, çok noktalı, uniform gibi türleri vardır. Bu çalışma kapsamında iyi çözümlerden uzaklaşmamak ve çeşitliliği alt düzeyde tutmak için tek noktalı çaprazlama kullanılmıştır.

Altı ebeveyn çaprazlanarak altı yeni birey (yeni jenerasyon) elde edilmesi amaçlanmaktadır. Popülasyonda bulunan bireyler ikili olarak değerlendirilmekte ve her bir ikili için birer rassal

sayı üretilerek bu rassal sayıların çaprazlama oranı ile kıyaslanarak çaprazlamanın yapılıp yapılamayacağına karar verilir.

Tablo 3.13: Çaprazlama gerçekleşme olasılığının incelenmesi.

ACT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	RS	Pcross	Çaprazlama
E2	2	5	3	1	3	2	3	4	5	3	3	2	3	3	1	3	3	2	0,18	0,7	VAR
E3	1	5	1	3	2	3	3	3	1	2	2	3	2	3	1	1	1	3			
E1	2	5	3	3	3	3	1	3	5	1	1	1	1	2	1	4	2	3	0,63	0,7	VAR
E3	1	5	1	3	2	3	3	3	1	2	2	3	2	3	1	1	1	3			
E1	2	5	3	3	3	3	1	3	5	1	1	1	1	2	1	4	2	3	0,84	0,7	YOK
E2	2	5	3	1	3	2	3	4	5	3	3	2	3	3	1	3	3	2			

Tablo 3.13'te görüldüğü gibi oluşturulan rassal sayılar, çözümün başında literatürde sıklıkla kullanılan değerlerden yola çıkarak belirlenen çaprazlama oranı ile kıyaslanarak çaprazlamanın yapılıp yapılmayacağı kararı irdelenmiştir. Bu durumda ilk iki ve sonraki iki ebeveyn aralarında gen alışverişi yapacak, son iki ebeveyn ise bir sonraki jenerasyona aynen aktarılacaktır. Tek noktalı çaprazlama 1 – 17 arasında rassal bir değer belirlenerek kromozomun o noktadan bölünüp, iki kromozom arasında yer değiştirmesi yoluyla gerçekleştirilir.

Tablo 3.14'te çaprazlama sonucu oluşan yeni bireyler ve yeni nesil gösterilmiştir. Yeni nesildeki her bir birey ebeveynlerden türeyen çocuklar olarak nitelendirilebilir.

Tablo 3.14: Çaprazlama sonucu oluşan yeni nesil ve matlab kodu.

	ACT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
KROMOZOM 7 (Ç1)		2	5	3	1	3	2	3	4	5	2	2	3	2	3	1	1	1	3
KROMOZOM 8 (Ç2)		1	5	1	3	2	3	3	3	1	3	3	2	3	3	1	3	3	2
KROMOZOM 9 (Ç3)		2	5	3	3	3	3	3	3	1	2	2	3	2	3	1	1	1	3
KROMOZOM 10 (Ç4)		1	5	1	3	2	3	1	3	5	1	1	1	1	2	1	4	2	3
KROMOZOM 11 (Ç5)		2	5	3	3	3	3	1	3	5	3	1	1	1	2	1	4	2	3
KROMOZOM 12 (Ç6)		2	5	3	1	3	2	3	4	5	3	3	2	3	3	1	3	3	2

```
%%TEK NOKTALI ÇAPRAZLAMA
```

```
pairs=randperm(psize);
```

```
for i=1:psize/2
parent1idx=pairs(2*i-1);
parent2idx=pairs(2*i);
parent1=arapop(parent1idx,:);
parent2=arapop(parent2idx,:);
```

```
%ÇAPRAZLAMA İÇİN RASSAL SAYI
```

```
rs=unifrnd(0,1);
```

```
if (rs<pcross)
cpoint=unidrnd(18-1);
dummy=parent1(cpoint+1:end);
```



```

parent1(cpoint+1:end)=parent2(cpoint+1:end);
parent2(cpoint+1:end)=dummy;
arapop(parent1idx,:)=parent1;
arapop(parent2idx,:)=parent2;
end
end
%%

```

Çaprazlama işleminin ardından yine rassal bazı işlemler uygulanarak mutasyon gerçekleştirilir. Kromozom 11 ve Kromozom 12'nin önceki popülasyondan aynen aktarılması çeşitliliği engellediğinden mutasyon operatörü ile bir veya birkaç gende değişiklikler yapmak farklı projelerinde değerlendirilmesine olanak tanıyacaktır.

3.2.5. Mutasyon Operatörü

Mutasyon algoritmanın yerel minimum noktalarına takılmasını engellemek için evrimin adaptasyon prensibinden yola çıkılarak türetilmiştir. Bireyler üzerindeki her bir gen için bir rassal sayı üretilir ve bu rassal sayılar mutasyon oranı ile karşılaştırılarak hangi gende değişiklik yapılacağı belirlenir. Mutasyon oranı çaprazlama sonucunda daha iyi sonuç verme olasılığı yüksek olan bireylerin olasılığını düşürmemek adına çok küçük bir değer olarak belirlenir (0,05 – 0,1 arası). Yeni jenerasyondaki her bir gen için üretilen rassal sayılar Tablo 3.15'te gösterilmiştir.

Tablo 3.15: Her bir gen için üretilen rassal sayılar.

Ç1	0,10	0,46	0,78	0,75	0,28	0,41	0,22	0,36	0,76	0,86	0,63	0,25	0,29	0,44	0,29	0,36	0,32	0,55
Ç2	0,96	0,61	0,31	0,30	0,13	0,73	0,72	0,17	0,21	0,68	0,21	0,94	0,52	0,93	0,53	0,61	0,08	0,34
Ç3	0,15	0,96	0,27	0,69	0,54	0,84	0,41	0,58	0,61	0,12	0,68	0,15	0,22	0,95	0,10	0,85	0,01	0,02
Ç4	0,74	0,28	0,15	0,64	0,33	0,41	0,92	0,69	0,70	0,94	0,55	0,22	0,57	0,85	0,45	0,85	0,64	0,64
Ç5	0,99	0,93	0,98	0,12	0,06	0,70	0,64	0,61	0,85	0,01	0,78	0,57	0,36	0,10	0,30	0,55	0,48	0,55
Ç6	0,47	0,26	0,06	0,44	0,09	0,43	0,38	0,95	0,49	0,16	0,27	0,67	0,49	0,83	0,62	0,64	0,046	0,41

Tablo 3.15'te işaretlenen rassal sayılar 0,05 olan mutasyon oranından küçük olduğundan bu sayılara denk gelen genlerde rassal değişiklikler yapılmıştır.

Tablo 3.16: Genlerde gerçekleşen mutasyon ve matlab kodu.

ACT	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	CP	TC	FITNESS
(Ç1)	2	5	3	1	3	2	3	4	5	2	2	3	2	3	1	1	1	3	130	142008	0,48
(Ç2)	1	5	1	3	2	3	3	3	1	3	3	2	3	3	1	3	3	2	146	112720	0,66
(Ç3)	2	5	3	3	3	3	3	3	1	2	2	3	2	3	1	1	3	1	134	111800	0,85
(Ç4)	1	5	1	3	2	3	1	3	5	1	1	1	1	2	1	4	2	3	133	124400	0,69
(Ç5)	2	5	3	3	3	3	1	3	5	3	1	1	1	2	1	4	2	3	142	120220	0,61
(Ç6)	2	5	3	1	3	2	3	4	5	3	3	2	3	3	1	3	2	2	135	138878	0,45

```

%%MUTASYON İÇİN RASSAL SAYI
rs=unifrnd(0,1,[psize,18]);

```



```

%MUTASYON OPERATÖRÜ
for i=1:psize
    for j=1:18
        if (rs(i,j)<pmutation)
            arapop(i,j)=randi([1,range(j)],1,1);
        end
    end
end
end
%%

```

Mutasyon gerçekleştirildikten sonra tekrar süre ve maliyet değerleri hesaplanıp uygunluk fonksiyonu kullanılarak yeni neslin uygunluk değerleri belirlenmiş olur. Tablo 3.16'da görüldüğü üzere bir önceki jenerasyona göre uygunluk değeri daha yüksek bir bireyin (Ç3) elde edilmesi sağlanmıştır.

Bu değerler elde edildikten sonra durdurma kriterini sağlayıp sağlamadığı kontrol edilerek analizin durması sağlanabilir. Durdurma kriteri olarak, iterasyon sayısı, belirlenen bir amaç fonksiyonu değerine yakınsama, belirli bir sürenin dolması, daha iyi çözüm üretmeyen iterasyon sayısı kullanılabilir. Bu çalışmada iterasyon sayısı durdurma kriteri olarak kullanılmıştır.

Yukarıda belirtilen işlemler iterasyon sayısı boyunca yapıldıktan sonra elde edilen değerler yakın optimum veya optimum değerler olabilir. Sezgisel metotlar optimum sonucu garanti etmese bile makul sürelerde yakın optimum sonuçlar vermeyi garanti eder.

İki iterasyon ilerletilen örnek problem üzerinden Genetik algoritmanın verdiği sonuçlarla elde edilen pareto değerleri (2. iterasyon sonucu elde edilen zaman maliyet bileşenleri) tablo 3.17'de verilmiştir.

Tablo 3.17: GA yaklaşımı ile 2. iterasyonda elde edilen pareto çözümleri.

BİREY	CP	TC
Ç1	130	142008
Ç2	133	124400
Ç3	134	111800
Ç4	135	138878
Ç5	142	120220
Ç6	146	112720

Genetik algoritma tek amaçlı minimizasyon veya maksimizasyon problemlerinde bulunduğu çözümleri her iterasyonda daha iyileyerek veya bir önceki iterasyondaki iyi çözümleri sonraki iterasyona elitizm yoluyla aktararak optimum sonuca ulaşmaya çalışır. Bu noktada başlangıç

popülasyonunun önemi çok amaçlı optimizasyondaki kadar büyük değildir. Her bir jenerasyon başlangıç popülasyonu kadar yeni birey üretir ve sonuçlar bu popülasyonun içinde aranır.

Bu çalışmada kullanılan test problemi 100 ila 170 gün arasında her bir gün için maliyet değerleri üretebilmektedir. Amacımız her gün için elde edilebilecek minimum maliyet değerlerine ulaşım bu değerlerin arasından pareto değerlerini seçmek olduğundan GA yaklaşımıyla min 70 başlangıç popülasyonu ile çözüme başlamak elzemdir.

Bu yolla bile 70 farklı gün için optimal değerlerin elde edilmesi pek mümkün değildir. Arama uzayının farklı bölgelerinden aramaya başlayan genetik algoritma operatörleri buldukları sonuçları iterasyonlar boyunca minimize etmeye çalışarak pareto değerlerine indirgemeye çalışırlar. Bunu yaparken bazı günler için hiç sonuç bulunamaması veya bulunan sonuçların çaprazlama ve mutasyon ile kaybedilmesi pareto değerlerinin elde edilmesini zorlaştırmaktadır.

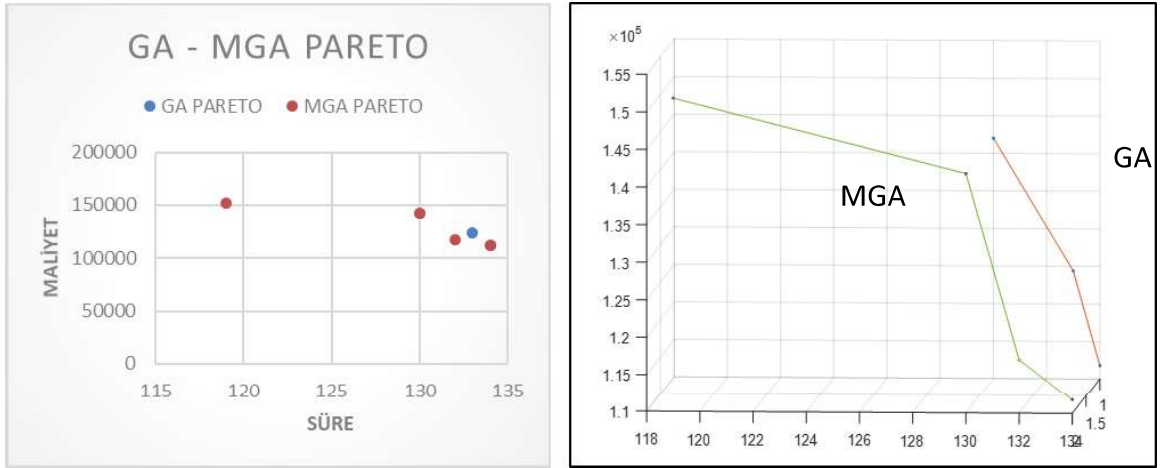
Bu tez kapsamında önerilen Modifiye Genetik Algoritma modelinde iterasyonlar boyunca her bir süreye karşılık gelen minimum maliyet değerleri ayrı bir matriste tutularak daha iyi bir çözüm bulunana kadar saklanması amaçlanmıştır. Bu yolla sadece son jenerasyondan elde edilen sonuçlar değil iterasyonlar boyunca üretilen iyi çözümlerin de sonuca dahil edilmesi sağlanmaktadır.

Tablo 3.18'de çözüme başlangıç popülasyonundaki değerlerin de eklenmesi ile pareto değerlerinin sayısının arttığı ve bir pareto değerinde iyileştiği- görülmektedir.

Tablo 3.18: MGA yaklaşımı ile elde edilen pareto çözümleri.

BİREY	CP	TC
E5	119	151858 *
Ç1	130	142008
E3	132	117250 *
Ç3	134	111800
Ç4	133	124400
E1	134	120350
Ç5	135	138878
E2	141	137478
Ç6	142	120220
Ç2	146	112720
E4	150	117890
E6	153	128520

Yapılan örnek çalışma ile GA yaklaşımının çalışma prensibi üzerinde durulmuş, çok amaçlı optimizasyon için bir modifikasyon uygulanarak MGA yaklaşımı geliştirilmiş ve sonuçları Şekil 3.5'te grafiklerle gösterilmiştir.



Şekil 3.5: GA ile MGA çözümlerinin pareto grafikleri.

İlerleyen bölümlerde öncelikle algoritmanın program kodu verilmiş ardından büyük başlangıç popülasyonu ve farklı iterasyon sayıları için MGA yaklaşımının verdiği sonuçlar değerlendirilmiştir. Makul sürede çözüme ulaşan başlangıç popülasyonu ve iterasyon sayıları belirlenmiş. Bu parametreler kullanılarak GA ve MGA sonuçları karşılaştırmalı olarak değerlendirilmiştir.

3.3. MATLAB PROGRAM KODU

Aşağıda Matlab R2017b versiyonunda yazılmış çok amaçlı çok modlu zaman maliyet ödünleşim problemine yönelik modifiye genetik algoritma kodu verilmiştir.

```

%% MULTIOBJECTIVE MULTIMODE TIME-COST TRADE-OFF ANALYSIS
%% MODIFIED GENETIC ALGORITHM MATLAB CODE
%% GA PARAMETERS
%Crossover Rate, Mutation Rate, İteration Number, Run Number
pcross=0.70; pmutation=0.05; it=500; run=5;

%SÜRE MATRİSİ-----
s= [ 14  15  16  21  24; 15  18  20  23  25;
     15  22  33  0   0; 12  16  20  0   0;
     22  24  28  30  0; 14  18  24  0   0;
     9  15  18  0   0; 14  15  16  21  24;
     15  18  20  23  25; 15  22  33  0   0;
     12  16  20  0   0; 22  24  28  30  0;
     14  18  24  0   0; 9  15  18  0   0;
     12  16  0   0   0; 20  22  24  28  30;
     14  18  24  0   0; 9  15  18  0   0 ];
%-----

```

```

%MALİYET MATRİSİ-----
c=[
    2400 2150 1900 1500 1200; 3000 2400 1800 1500 1000
    4500 4000 3200 0 0 ; 45000 35000 30000 0 0
    20000 17500 15000 10000 0 ; 40000 32000 18000 0 0
    30000 24000 22000 0 0 ; 220 215 200 208 120
    300 240 180 150 100 ; 450 400 320 0 0
    450 350 300 0 0 ; 2000 1750 1500 1000 0
    4000 3200 1800 0 0 ; 3000 2400 2200 0 0
    4500 3500 0 0 0 ; 3000 2000 1750 1500 1000
    4000 3200 1800 0 0 ; 3000 2400 2200 0 0];
%-----

psize =input('Başlangıç popülasyon adedini giriniz=');

%% RUN START
for ii=1:run

%%initializing
pop=zeros(psize,18) ;%Popülasyon
cost=zeros(psize,18) ;%Maliyet matrisi
dur=zeros(psize,18) ;%Süre matrisi
CP=zeros(psize,1) ;%Proje süresi
TC=zeros(psize,1) ;%Proje maliyeti
es=zeros(psize,18) ;%Early start
ef=zeros(psize,18) ;%Early finish
fitness=zeros(psize,1) ;%Uygunluk değerleri

%%MATRİS HAFIZASI
MaliyetMatrisi = zeros(psize,it-1);
SureMatrisi = zeros(psize,it-1);
PopMatrisi = zeros(psize*(it-1),18);
P_CC= zeros(psize*(run-1),20);

%RASSAL BAŞLANGIÇ POPÜLASYONUNUN OPSİYON GENLİKLERİ

range(1)=5 ;range(2)=5 ;range(3)=3 ;range(4)=3 ;range(5)=4 ;range(6)=3;
range(7)=3 ;range(8)=5 ;range(9)=5 ;range(10)=3;range(11)=3;range(12)=4;
range(13)=3;range(14)=3;range(15)=2;range(16)=5;range(17)=3;range(18)=3;

%RASSAL BAŞLANGIÇ POPÜLASYONUNUN OLUŞTURULMASI
for j=1:psize
for i=1:18
pop(j,i)=randi([1,range(i)],1,1);
end
end

iteration=1;
while (iteration<it)

%ÜRETİLEN OPSİYONLARA KARŞILIK GELEN SÜRE VE MALİYET DEĞERLERİ
for j=1:18
for i=1:psize
cost(i,j)= c(j, pop(i,j));
dur(i,j)=s(j, pop(i,j));
end
end

```

```
%% CPM HESAPLAMALARI
```

```
for i= 1:psize
```

```

es(i,1)=0 ; ef(i,1)=dur(i,1) ; es(i,2)=0 ; ef(i,2)=dur(i,2) ; es(i,3)=0 ;
ef(i,3)=dur(i,3) ; es(i,4)=0 ; ef(i,4)=dur(i,4) ; es(i,5)=ef(i,1) ;
ef(i,5)=es(i,5)+dur(i,5) ; es(i,6)=ef(i,1) ; ef(i,6)=es(i,6)+dur(i,6) ;
es(i,13)=ef(i,3) ; ef(i,13)=es(i,13)+dur(i,13) ; es(i,7)=ef(i,5) ;
ef(i,7)=es(i,7)+dur(i,7) ; es(i,8)=ef(i,6) ; ef(i,8)=es(i,8)+dur(i,8) ;
es(i,9)=ef(i,6) ; ef(i,9)=es(i,9)+dur(i,9) ; es(i,10)=max(ef(i,6),ef(i,2)) ;
ef(i,10)=es(i,10)+dur(i,10) ; es(i,11)=max(ef(i,8),ef(i,7)) ;
ef(i,11)=es(i,11)+dur(i,11) ; ab=[ef(i,5) ef(i,9)
ef(i,10)] ; es(i,12)=max(ab) ;
ef(i,12)=es(i,12)+dur(i,12) ; es(i,14)=max(ef(i,4),ef(i,10)) ;
ef(i,14)=es(i,14)+dur(i,14) ; es(i,15)=ef(i,12) ;
ef(i,15)=es(i,15)+dur(i,15) ;
es(i,16)=max(ef(i,13),ef(i,14)) ; ef(i,16)=es(i,16)+dur(i,16) ;
bc=[ef(i,11) ef(i,14) ef(i,15)] ;
es(i,17)=max(bc) ; ef(i,17)=es(i,17)+dur(i,17) ;
es(i,18)=max(ef(i,16),ef(i,17)) ; ef(i,18)=es(i,18)+dur(i,18) ;

```

```

CP(i,1)=ef(i,18) ; %Proje Süresi
TC(i,1)=sum(cost(i,:)) ; %Proje Maliyeti

```

```

PopMatrisi((((iteration-1)*psize)+1):(iteration*psize),:) =
pop(1:psize, :) ;
end

```

```

MaliyetMatrisi(:,iteration) = TC ;
SureMatrisi(:,iteration) = CP ;

```

```
%% MAWA YÖNTEMİ İLE AĞIRLIKLANDIRMA
```

```
Zcmax=max(TC) ; Zcmin=min(TC) ; Ztmin=min(CP) ; Ztmax=max(CP) ;
```

```

if (Ztmax~=Ztmin && Zcmax~=Zcmin)
    Vc=Zcmin/(Zcmax-Zcmin) ; Vt=Ztmin/(Ztmax-Ztmin) ;
    V=Vc+Vt ; Wc=Vc/V ; Wt=Vt/V ;
end

```

```

if (Ztmax==Ztmin && Zcmax==Zcmin)
    Wc=0.5 ; Wt=0.5 ;
end

```

```

if (Ztmax==Ztmin && Zcmax~=Zcmin)
    Wc=0.1 ; Wt=0.9 ;
end

```

```

if (Ztmax~=Ztmin && Zcmax==Zcmin)
    Wc=0.9 ; Wt=0.1 ;
end

```

```
%% FITNESS FUNCTION
```

```
for i=1:psize
```

```

fitness(i)=Wt*((Ztmax-CP(i))+rand)/((Ztmax-Ztmin)+rand)+ Wc*((Zcmax-
TC(i))+rand)/((Zcmax-Zcmin)+rand);
end

% UYGUNLUK DEĞERİ OLASILIKLARI
probs=fitness/(sum(fitness)) ;

% KÜMÜLATİF UYGUNLUK DEĞERİ OLASILIKLARI
cprobs=probs;
for k=2:psize
cprobs(k)=cprobs(k-1)+probs(k);
end

%% SEÇİM
rs=unifrnd(0,1,[psize,1]); %Seçim için rassal sayı
arapop=pop;

for i=1:psize
jdx=find(rs(i)<cprobs,1);
arapop(i,:)=pop(jdx,:);
end

%% ÇAPRAZLAMA
pairs=randperm(psize); %Tek noktalı çaprazlama için rassal sayı

for i=1:psize/2
parent1idx=pairs(2*i-1);
parent2idx=pairs(2*i);
parent1=arapop(parent1idx,:);
parent2=arapop(parent2idx,:);

rs=unifrnd(0,1); %Çaprazlama kararı için rassal sayı

if (rs<pcross)
cpoint=unidrnd(18-1);
dummy=parent1(cpoint+1:end);
parent1(cpoint+1:end)=parent2(cpoint+1:end);
parent2(cpoint+1:end)=dummy;
arapop(parent1idx,:)=parent1;
arapop(parent2idx,:)=parent2;
end
end

%% MUTASYON
rs=unifrnd(0,1,[psize,18]); %Mutasyon için rassal sayı

for i=1:psize
for j=1:18
if (rs(i,j)<pmutation)
arapop(i,j)=randi([1,range(j)],1,1);
end
end
end

pop=arapop;

```

```

iteration=iteration+1;
disp(iteration)

end

%% PARETO CHART
%Tüm maliyetler MaliyetMatrisinde tutuluyor.
%Tüm süreler SureMatrisi nde tutuluyor.
%Single column MaliyetMatrisi:

S_MaliyetMatrisi = reshape(MaliyetMatrisi,[psize.*(it-1),1]);
%single column SureMatrisi:
S_SureMatrisi = reshape(SureMatrisi,[psize.*(it-1),1]);

A = [ S_SureMatrisi S_MaliyetMatrisi PopMatrisi];

v = accumarray( A(:,1), A(:,2), [], @min );
B = [(1:size(v))' v];
BB = A(ismember(A(:,[1 2]),B,'rows'),:);
BB(BB(:,2) == 0,:) = []; %aynı maliyetteki en kısa süreler seçildi

vv = accumarray( BB(:,2), BB(:,1), [], @min );
C = [(1:size(vv))' vv];
CC = BB(ismember(BB(:,[2 1]),C,'rows'),:);
CC(CC(:,2) == 0,:) = []; %aynı süredeki min maliyetler seçildi

end %run end

%% PARETO GRAFİK
Pareto = scatter (CC(:,1),CC(:,2),'MarkerEdgeColor',[0 .5
.5],'MarkerFaceColor',[0 .7 .7],'LineWidth',1.5);
title('GA SONUÇ TABLOSU');
xlabel('Süre (Gün)'); % x-axis label
ylabel('Maliyet (TL)'); % y-axis label

%% SONUÇ YAZDIRMA
xlswrite('cagri_data_mga.xls', CC, 'cagri_data_mga',
sprintf('A%d',ii*70+1));
ii=ii+1;

```

4. BULGULAR

Bu bölümde başlangıç popülasyonunun ve iterasyon sayısının pareto çözümü üzerine etkisi MGA ile incelenecektir. Bulunan sonuçlara göre kabul edilebilir analiz süresine sahip parametreler seçilerek GA ile MGA çözümleri değerlendirilecektir.

Analizler Intel Core i5 – 7200U CPU @ 2.50 GHz işlemci, 4 GB ram, 64 bit işletim sistemli bir bilgisayarda, Matlab’da yazılan kodlarla yapılmıştır.

4.1. POPÜLASYON BÜYÜKLÜĞÜNÜN PARETO ÇÖZÜMÜNE ETKİSİ

Bu bölümde farklı başlangıç popülasyonu değerleri ile elde edilecek pareto çözümü sayısı ile çözümlerin analiz süresi incelenmiştir. İstatistiksel açıdan anlamlı sonuçlar elde edebilmek için analizler 30 koşum çalıştırılmıştır. Tablo 4.1’de gösterile analiz parametreleri ile MGA analizleri yapılmıştır.

Tablo 4.1: Popülasyon büyüklüğünün değerlendirilmesine yönelik parametreler.

MGA PARAMETRELERİ	
Çaprazlama oranı (Pc)	0,75
Mutasyon oranı (Pm)	0,05
Başlangıç Popülasyonu (Psize)	50 100 500 1000 5000
Run sayısı	30
İterasyon sayısı (It)	500

4.1.1. 50 Popülasyon Büyüklüğü ile Çözüm

Başlangıç popülasyonu 50 alınarak 500 iterasyon ileletilen analizin pareto çözümleri Tablo 4.2’de verilmiştir.

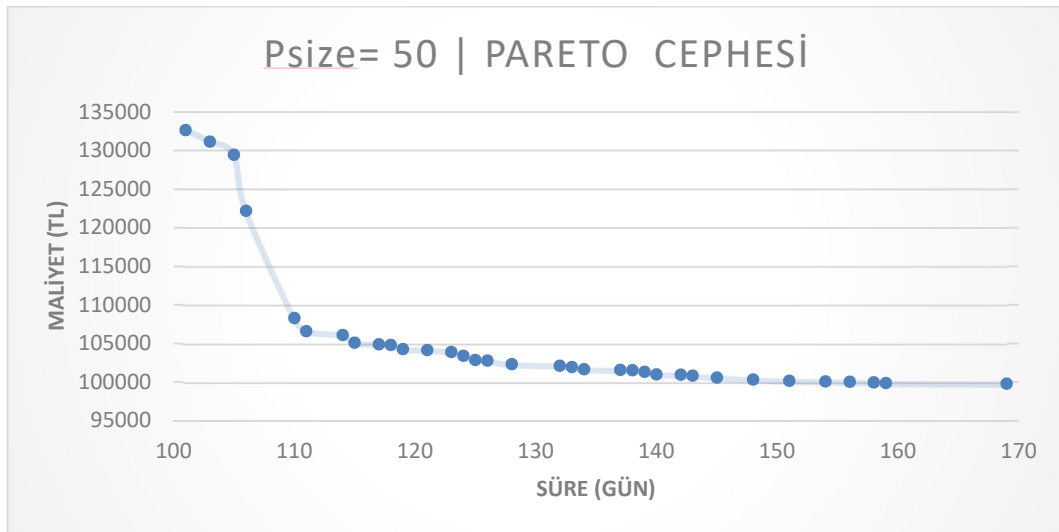
Tablo 4.2: Başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.

Psize=50 Pcross=0.70 Pmut=0.05 Run=30 İter=500 Dur=62.877491 sec																			
SÜRE	MALİYET	AKTİVİTELER																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
101	132650	1	4	3	3	4	1	3	3	1	1	2	1	1	1	1	3	1	1
103	131158	1	1	3	3	4	1	3	4	1	1	2	2	3	3	1	2	1	1
105	129470	1	4	2	3	4	1	3	1	1	1	3	1	3	1	2	5	1	1
106	122200	2	5	1	3	4	2	3	3	1	1	3	1	3	1	1	5	1	1
110	108300	1	3	2	3	4	3	3	3	1	1	1	1	3	2	1	5	1	1
111	106608	2	5	3	3	4	3	3	4	1	1	3	1	3	3	1	4	1	1
114	106100	1	5	3	3	4	3	3	3	1	1	2	1	3	3	1	4	2	1
115	105120	2	5	3	3	4	3	3	1	1	1	3	1	3	3	2	5	1	1
117	104920	2	5	3	3	4	3	3	5	1	1	1	2	3	3	2	5	1	1

Tablo 4.2 (devam): Başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.

118	104820	3	5	3	3	4	3	3	1	1	1	3	2	3	2	2	5	1	1
119	104270	2	5	3	3	4	3	3	5	1	1	2	1	3	3	2	5	2	1
121	104158	2	5	3	3	4	3	3	4	1	1	2	1	3	2	1	5	3	1
123	103910	3	5	3	3	4	3	3	5	2	1	3	1	3	3	2	5	2	1
124	103420	3	5	3	3	4	3	3	1	1	1	3	2	3	3	1	5	3	1
125	102870	2	5	3	3	4	3	3	5	1	1	2	1	3	3	2	5	3	1
126	102800	3	5	3	3	4	3	3	3	1	1	1	1	3	3	2	5	3	1
128	102320	3	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	3	1
132	102120	1	5	3	3	4	3	3	5	1	1	2	4	3	3	2	5	3	1
133	101965	2	5	3	3	4	3	3	2	1	1	2	4	3	3	2	5	3	1
134	101658	3	5	3	3	4	3	3	4	1	1	3	4	3	3	2	5	3	1
137	101570	4	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	2
138	101558	1	5	3	3	4	3	3	4	1	1	3	4	3	3	2	5	3	2
139	101315	2	5	3	3	4	3	3	2	1	1	3	4	3	3	2	5	3	2
140	101020	3	5	3	3	4	3	3	5	1	1	2	4	3	3	2	5	3	2
142	100958	5	5	3	3	4	3	3	4	1	1	3	4	3	3	2	5	3	1
143	100850	3	5	3	3	4	3	3	3	1	1	3	4	3	3	2	5	3	3
145	100570	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
148	100358	5	5	3	3	4	3	3	4	1	1	3	4	3	3	2	5	3	2
151	100158	5	5	3	3	4	3	3	4	1	1	3	4	3	3	2	5	3	3
154	100090	5	5	3	3	4	3	3	3	2	1	3	4	3	3	2	5	3	3
156	10P0030	5	5	3	3	4	3	3	3	3	1	3	4	3	3	2	5	3	3
158	99950	5	5	3	3	4	3	3	5	3	2	2	4	3	3	2	5	3	3
159	99870	5	5	3	3	4	3	3	5	4	2	3	4	3	3	2	5	3	3
169	99790	5	5	3	3	4	3	3	5	5	3	2	4	3	3	2	5	3	3

Yapılan analizler sonucunda 50 başlangıç popülasyonu 500 iterasyon ilerletildiğinde 34 adet pareto değeri bulunmuştur. Analiz süresi 63 saniye sürmüştür. Pareto cephesine ait grafik Şekil 4.1'de gösterilmiştir.

**Şekil 4.1:** 50 başlangıç popülasyonu ile elde edilen pareto cephesi.

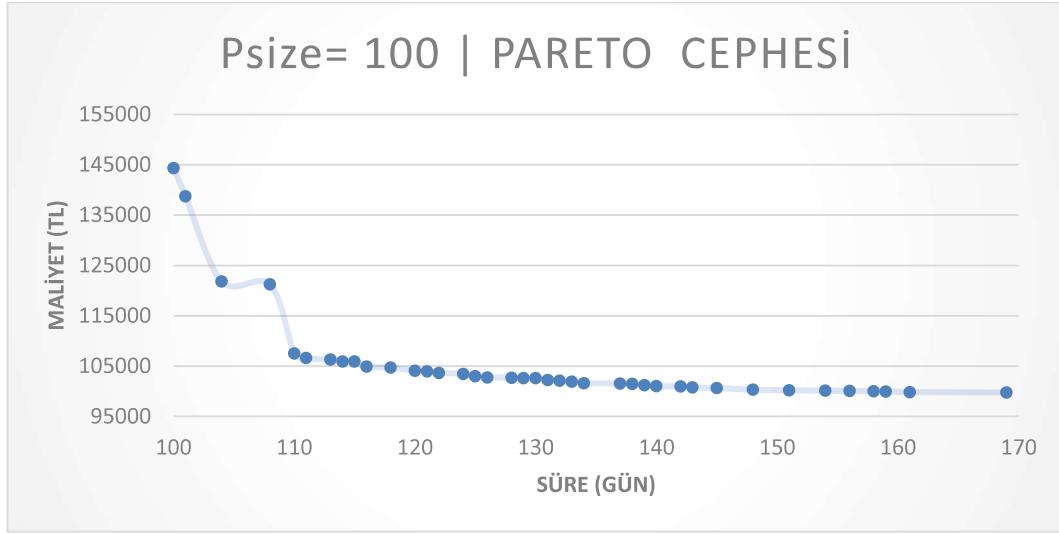
4.1.2. 100 Popülasyon Büyüklüğü ile Çözüm

Başlangıç popülasyonu 100 alınarak 500 iterasyon iletilen analizin pareto çözümleri Tablo 4.3'de verilmiştir.

Tablo 4.3: Başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.

Psize=100 Pcross=0.70 Pmut=0.05 Run=30 İter=500 Dur=79.218286 sec																			
SÜRE	MALİYET	AKTİVİTELER																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
100	144320	1	3	2	2	3	1	2	1	1	1	3	1	2	2	1	3	1	1
101	138715	1	5	2	3	4	1	1	2	1	1	1	1	2	3	1	5	1	1
104	121800	1	4	2	3	4	2	3	3	1	1	1	1	3	3	1	5	1	1
108	121215	3	5	2	3	4	2	3	2	1	1	2	2	3	3	1	3	1	1
110	107508	1	5	2	3	4	3	3	4	1	1	1	1	3	2	1	5	1	1
111	106600	2	4	3	3	4	3	3	3	1	1	3	1	3	3	1	5	1	1
113	106298	1	5	3	3	4	3	3	4	2	1	3	1	3	3	1	5	1	1
114	105915	1	5	3	3	4	3	3	2	1	1	2	1	3	3	2	4	1	1
115	105870	2	3	3	3	4	3	3	5	1	1	2	1	3	3	2	5	1	1
116	104900	3	5	3	3	4	3	3	3	1	1	2	1	3	3	2	5	1	1
118	104658	3	5	3	3	4	3	3	4	1	1	2	2	3	3	2	5	1	1
120	104050	3	5	3	3	4	3	3	3	1	1	3	1	3	3	2	5	2	1
121	103900	2	5	3	3	4	3	3	3	1	1	3	1	3	3	1	5	3	1
122	103620	3	5	3	3	4	3	3	5	1	1	2	1	3	3	1	5	3	1
124	103408	1	5	3	3	4	3	3	4	1	1	2	1	3	2	2	5	3	1
125	102965	2	5	3	3	4	3	3	2	1	1	2	1	3	3	2	5	3	1
126	102720	3	5	3	3	4	3	3	1	1	1	2	1	3	3	2	5	3	1
128	102665	3	5	3	3	4	3	3	2	1	1	2	2	3	2	2	5	3	1
129	102598	3	5	3	3	4	3	3	4	2	1	3	1	3	3	2	5	3	1
130	102570	1	5	3	3	4	3	3	5	1	1	3	3	3	3	2	5	3	1
131	102220	2	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	2
132	102058	3	5	3	3	4	3	3	4	1	1	3	1	3	3	2	5	3	2
133	101915	2	5	3	3	4	3	3	2	1	1	3	4	3	3	2	5	3	1
134	101570	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
137	101520	3	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	3	3
138	101470	1	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
139	101170	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
140	100970	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
142	100950	5	5	3	3	4	3	3	3	1	1	3	4	3	3	2	5	3	1
143	100770	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	3
145	100620	4	5	3	3	4	3	3	5	1	1	2	4	3	3	2	5	3	2
148	100270	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
151	100150	5	5	3	3	4	3	3	3	1	1	3	4	3	3	2	5	3	3
154	100110	5	5	3	3	4	3	3	1	2	1	3	4	3	3	2	5	3	3
156	100030	5	5	3	3	4	3	3	3	3	1	3	4	3	3	2	5	3	3
158	99980	5	5	3	3	4	3	3	3	3	2	3	4	3	3	2	5	3	3
159	99920	5	5	3	3	4	3	3	5	4	2	2	4	3	3	2	5	3	3
161	99820	5	5	3	3	4	3	3	5	5	2	3	4	3	3	2	5	3	3
169	99740	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3

Yapılan analizler sonucunda 100 başlangıç popülasyonu 500 iterasyon iletilildiğinde 39 adet pareto değeri bulunmuştur. Analiz süresi 79 saniye sürmüştür. Pareto cephesine ait grafik Şekil 4.2'de gösterilmiştir.



Şekil 4.2: 100 başlangıç popülasyonu ile elde edilen pareto cephesi.

4.1.3. 500 Popülasyon Büyüklüğü ile Çözüm

Başlangıç popülasyonu 500 alınarak 500 iterasyon ile verilen analizin pareto çözümleri Tablo 4.4'de verilmiştir.

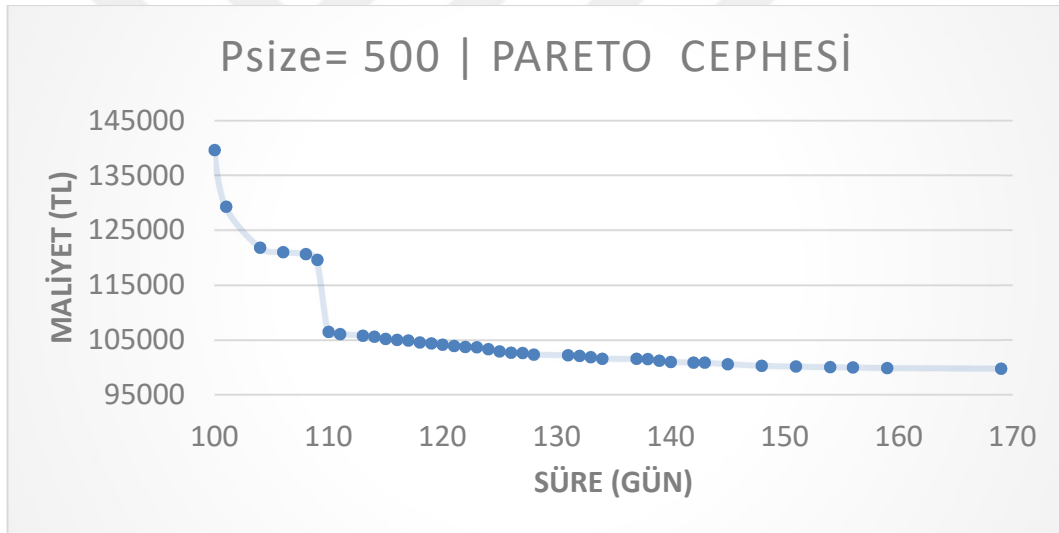
Tablo 4.4: 500 Başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.

Psize=500 Pcross=0.70 Pmut=0.05 Run=30 İter=500 Dur=355.616098 sec																			
SÜRE	MALİYET	AKTİVİTELER																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
100	139608	1	3	1	3	3	1	2	4	1	1	2	1	2	2	1	4	1	1
101	129258	1	5	3	3	4	1	3	4	1	1	1	1	3	3	1	3	1	1
104	121808	1	5	3	3	4	2	3	4	1	1	2	1	2	3	1	5	1	1
106	120970	1	5	3	3	4	2	3	1	1	1	1	2	3	2	1	4	1	1
108	120620	1	5	1	3	4	2	3	5	1	1	2	1	3	3	2	5	1	1
109	119570	1	5	3	3	4	2	3	1	1	1	3	1	3	3	1	5	2	1
110	106470	1	5	3	3	4	3	3	5	1	1	3	1	3	2	1	5	1	1
111	106070	2	5	3	3	4	3	3	5	1	1	2	1	3	3	1	5	1	1
113	105770	2	5	3	3	4	3	3	5	1	1	3	2	3	3	1	5	1	1
114	105600	3	5	3	3	4	3	3	3	1	1	3	2	3	3	1	5	1	1
115	105170	2	5	3	3	4	3	3	1	1	1	2	1	3	3	2	5	1	1
116	104970	3	5	3	3	4	3	3	5	1	1	3	1	3	2	2	5	1	1
117	104858	2	5	3	3	4	3	3	4	1	1	3	2	3	3	2	5	1	1
118	104520	1	5	3	3	4	3	3	5	1	1	2	1	3	3	2	5	2	1
119	104370	2	5	3	3	4	3	3	1	1	1	2	1	3	3	2	5	2	1
120	104108	3	5	3	3	4	3	3	4	1	1	2	1	3	3	2	5	2	1
121	103908	2	5	3	3	4	3	3	4	1	1	3	1	3	3	1	5	3	1
122	103720	3	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	2	1
123	103670	2	5	3	3	4	3	3	1	1	1	3	2	3	3	1	5	3	1
124	103300	1	5	3	3	4	3	3	3	1	1	1	1	3	3	2	5	3	1
125	102908	2	5	3	3	4	3	3	4	1	1	3	1	3	3	2	5	3	1
126	102665	3	5	3	3	4	3	3	2	1	1	3	1	3	3	2	5	3	1
127	102620	2	5	3	3	4	3	3	5	1	1	2	2	3	3	2	5	3	1
128	102320	3	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	3	1
131	102170	4	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1

Tablo 4.4 (devam): 500 Başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.

132	102070	3	5	3	3	4	3	3	1	1	1	3	1	3	3	2	5	3	2
133	101820	2	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
134	101570	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
137	101520	3	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	3	3
138	101470	3	5	3	3	4	3	3	5	1	1	3	3	3	3	2	5	3	2
139	101170	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
140	100970	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
142	100870	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
143	100820	3	5	3	3	4	3	3	5	1	1	2	4	3	3	2	5	3	3
145	100570	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
148	100270	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
151	100120	5	5	3	3	4	3	3	5	1	1	2	4	3	3	2	5	3	3
154	100010	5	5	3	3	4	3	3	5	2	1	3	4	3	3	2	5	3	3
156	99950	5	5	3	3	4	3	3	5	3	1	3	4	3	3	2	5	3	3
159	99870	5	5	3	3	4	3	3	5	4	2	3	4	3	3	2	5	3	3
169	99740	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3

Yapılan analizler sonucunda 500 başlangıç popülasyonu 500 iterasyon ilerletildiğinde 41 adet pareto değeri bulunmuştur. Analiz süresi 355 saniye sürmüştür. Pareto cephesine ait grafik Şekil 4.3'de gösterilmiştir.

**Şekil 4.3:** 500 başlangıç popülasyonu ile elde edilen pareto cephesi.

4.1.4. 1000 Popülasyon Büyüklüğü ile Çözüm

Başlangıç popülasyonu 1000 alınarak 500 iterasyon ilerletilen analizin pareto çözümleri Tablo 4.5'de verilmiştir.

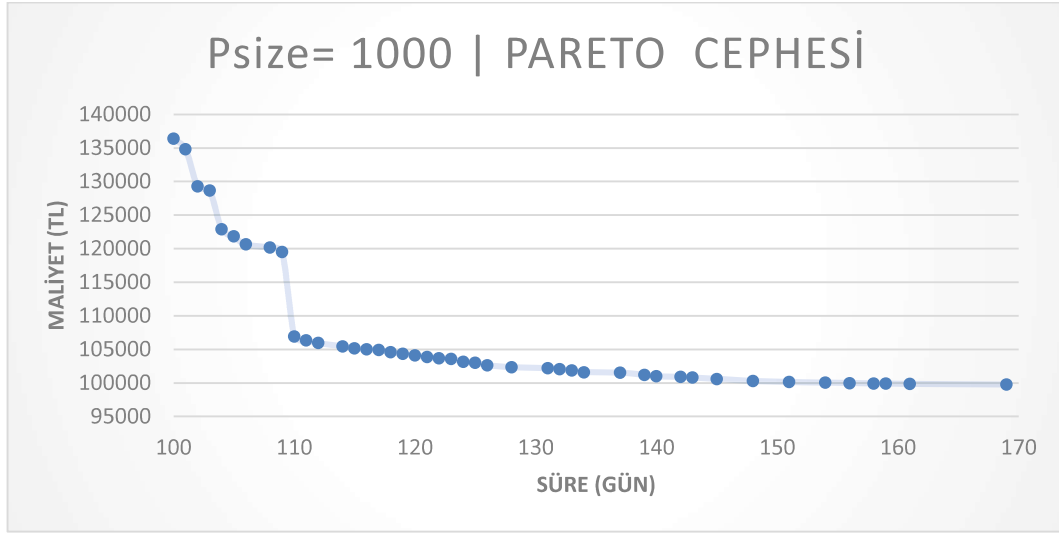
Tablo 4.5: 1000 başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.

Psize=1000 Pcross=0.70 Pmut=0.05 Run=30 İter=500 Dur=899.741472 sec		AKTİVİTELER																	
SÜRE	MALİYET	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
100	136350	1	5	3	3	3	1	2	3	1	1	2	1	3	2	1	3	1	1
101	134808	1	1	3	3	4	1	2	4	1	1	2	1	2	3	1	2	1	1
102	129258	2	5	2	3	4	1	3	4	1	1	1	1	3	2	1	5	1	1

Tablo 4.5 (devam): 1000 başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.

103	128620	3	4	3	3	4	1	3	1	1	1	2	1	3	2	1	5	1	1
104	122858	1	5	3	3	4	2	2	4	1	1	3	1	3	3	1	4	1	1
105	121800	2	5	3	3	4	2	3	3	1	1	1	1	3	1	1	3	1	1
106	120608	3	4	3	3	4	2	3	4	1	1	2	1	3	2	1	5	1	1
108	120150	1	5	3	3	4	2	3	3	1	1	3	1	3	1	2	5	1	1
109	119470	1	5	3	3	4	2	3	5	1	1	3	1	3	3	1	5	2	1
110	106908	1	5	3	3	4	3	3	4	1	1	2	1	3	3	1	4	1	1
p111	106320	2	5	3	3	4	3	3	1	1	1	3	1	3	2	1	5	1	1
112	105915	3	5	3	3	4	3	3	2	1	1	2	1	3	3	1	5	1	1
114	105415	1	5	3	3	4	3	3	2	1	1	2	1	3	3	2	5	1	1
115	105115	2	5	3	3	4	3	3	2	1	1	3	1	3	3	2	5	1	1
116	104970	3	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	2	1
117	104870	2	5	3	3	4	3	3	1	1	1	3	2	3	3	2	5	1	1
118	104565	1	5	3	3	4	3	3	2	1	1	3	1	3	3	2	5	2	1
119	104300	2	5	3	3	4	3	3	3	1	1	3	1	3	3	2	5	2	1
120	104070	1	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	3	1
121	103820	2	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	3	1
122	103658	3	5	3	3	4	3	3	4	1	1	3	1	3	3	1	5	3	1
123	103570	2	5	3	3	4	3	3	5	1	1	3	2	3	3	1	5	3	1
124	103120	1	5	3	3	4	3	3	5	1	1	2	1	3	3	2	5	3	1
125	102965	2	5	3	3	4	3	3	2	1	1	2	1	3	3	2	5	3	1
126	102620	3	5	3	3	4	3	3	5	1	1	2	1	3	3	2	5	3	1
128	102320	3	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	3	1
131	102170	4	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
132	102020	3	5	3	3	4	3	3	5	1	1	2	1	3	3	2	5	3	2
133	101820	2	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
134	101570	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
137	101510	3	5	3	3	4	3	3	5	2	1	3	4	3	3	2	5	3	1
139	101170	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
140	100970	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
142	100870	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
143	100770	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	3
145	100570	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
148	100270	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
151	100120	5	5	3	3	4	3	3	5	1	1	2	4	3	3	2	5	3	3
154	100010	5	5	3	3	4	3	3	5	2	1	3	4	3	3	2	5	3	3
156	99950	5	5	3	3	4	3	3	5	3	1	3	4	3	3	2	5	3	3
158	99900	5	5	3	3	4	3	3	5	3	2	3	4	3	3	2	5	3	3
159	99870	5	5	3	3	4	3	3	5	4	2	3	4	3	3	2	5	3	3
161	99820	5	5	3	3	4	3	3	5	5	2	3	4	3	3	2	5	3	3
169	99740	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3

Yapılan analizler sonucunda 1000 başlangıç popülasyonu 500 iterasyon ilerletildiğinde 44 adet pareto değeri bulunmuştur. Analiz süresi 900 saniye sürmüştür. Pareto cephesine ait grafik Şekil 4.4’de gösterilmiştir.



Şekil 4.4: 1000 başlangıç popülasyonu ile elde edilen pareto cephesi.

4.1.5. 5000 Popülasyon Büyüklüğü ile Çözüm

Başlangıç popülasyonu 5000 alınarak 500 iterasyon ile gerçekleştirilen analizin pareto çözümleri Tablo 4.6'da verilmiştir.

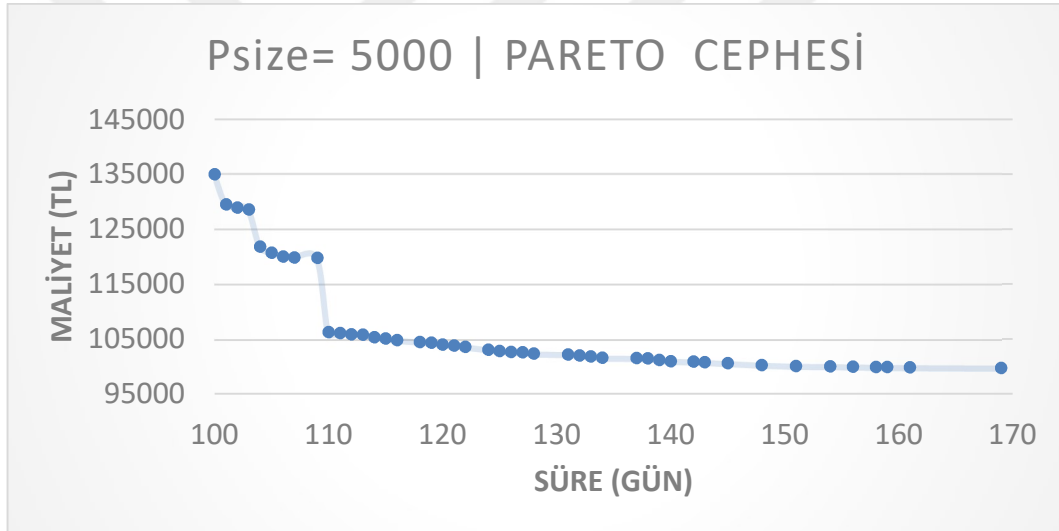
Tablo 4.6: 5000 başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.

Psize=5000 Pcross=0.70 Pmut=0.05 Run=30 İter=500 Dur=16551.899711 sec																			
SÜRE	MALİYET	AKTİVİTELER																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
100	135000	1	5	3	3	3	1	3	3	1	1	2	1	2	2	1	5	1	1
101	129520	1	5	2	3	4	1	3	1	1	1	1	1	3	2	1	5	1	1
102	128970	2	3	3	3	4	1	3	5	1	1	1	1	3	3	1	5	1	1
103	128600	3	4	3	3	4	1	3	3	1	1	2	1	3	2	1	5	1	1
104	121820	1	2	3	3	4	2	3	1	1	1	2	1	3	3	1	5	1	1
105	120758	2	4	3	3	4	2	3	4	1	1	1	1	3	3	1	5	1	1
106	120020	1	5	3	3	4	2	3	5	1	1	3	2	3	3	1	5	1	1
107	119858	2	5	3	3	4	2	3	4	1	1	3	2	3	3	1	5	1	1
109	119770	2	4	3	3	4	2	3	1	1	1	1	1	3	3	2	5	1	1
110	106270	1	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	1	1
111	106108	2	5	3	3	4	3	3	4	1	1	3	1	3	3	1	5	1	1
112	105850	3	5	3	3	4	3	3	3	1	1	3	1	3	3	1	5	1	1
113	105770	2	5	3	3	4	3	3	5	1	1	3	2	3	3	1	5	1	1
114	105350	1	5	3	3	4	3	3	3	1	1	3	1	3	3	2	5	1	1
115	105070	2	5	3	3	4	3	3	5	1	1	2	1	3	3	2	5	1	1
116	104770	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	1	1
118	104470	1	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	2	1
119	104308	2	5	3	3	4	3	3	4	1	1	3	1	3	3	2	5	2	1
120	103970	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	2	1
121	103820	2	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	3	1
122	103570	3	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	3	1
124	103070	1	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
125	102820	2	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
126	102620	3	5	3	3	4	3	3	5	1	1	2	1	3	3	2	5	3	1
127	102570	2	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	3	1
128	102320	3	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	3	1
131	102170	4	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
132	101970	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	2
133	101820	2	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1

Tablo 4.6 (devam): 5000 başlangıç popülasyonu için pareto değerleri ve mod bileşenleri.

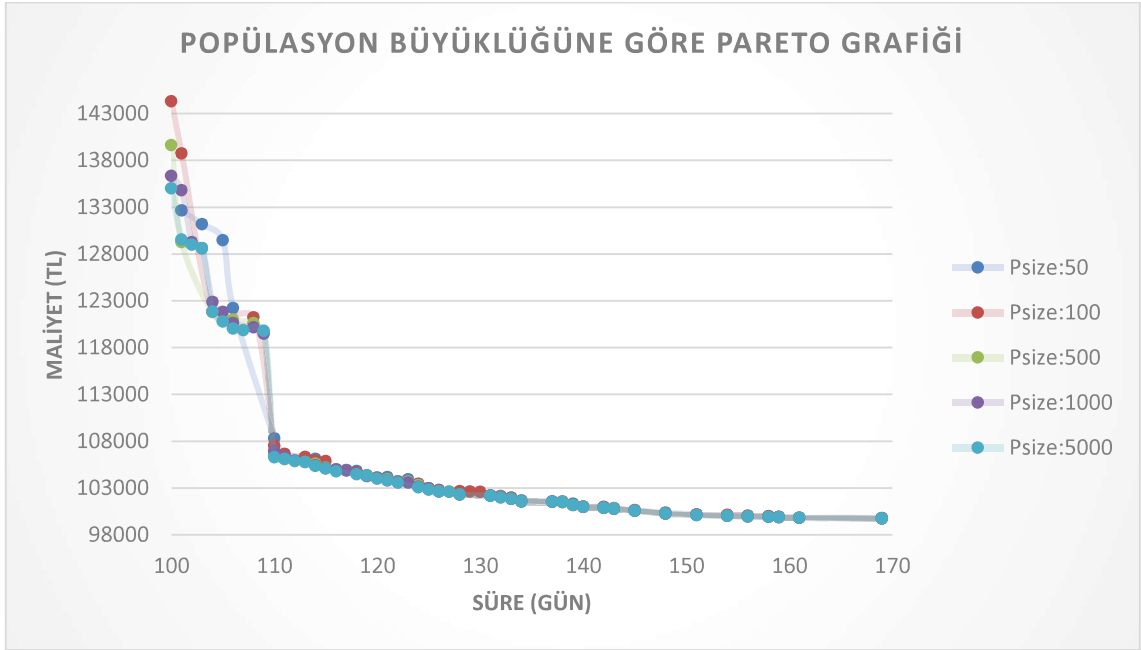
134	101570	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
137	101510	3	5	3	3	4	3	3	5	2	1	3	4	3	3	2	5	3	1
138	101470	3	5	3	3	4	3	3	5	1	1	3	3	3	3	2	5	3	2
139	101170	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
140	100970	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
142	100870	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
143	100770	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	3
145	100570	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
148	100270	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
151	100070	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	3
154	100010	5	5	3	3	4	3	3	5	2	1	3	4	3	3	2	5	3	3
156	99950	5	5	3	3	4	3	3	5	3	1	3	4	3	3	2	5	3	3
158	99900	5	5	3	3	4	3	3	5	3	2	3	4	3	3	2	5	3	3
159	99870	5	5	3	3	4	3	3	5	4	2	3	4	3	3	2	5	3	3
161	99820	5	5	3	3	4	3	3	5	5	2	3	4	3	3	2	5	3	3
169	99740	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3

Yapılan analizler sonucunda 5000 başlangıç popülasyonu 500 iterasyon ilerletildiğinde 45 adet pareto değeri bulunmuştur. Analiz süresi 16552 saniyedir. Pareto cephesine ait grafik Şekil 4.5’de gösterilmiştir.

**Şekil 4.5:** 5000 başlangıç popülasyonu ile elde edilen pareto cephesi.

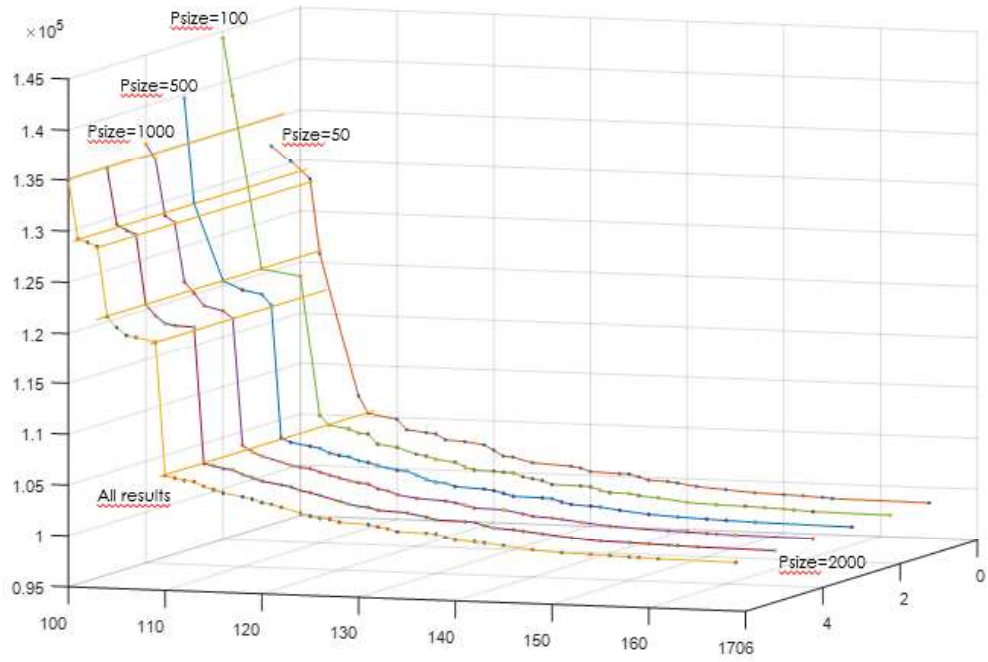
4.1.6. Popülasyon Büyüklüğüne Göre Çözümlerin Karşılaştırılması

Yapılan analizler sonucunda popülasyon büyüklüğünün artması ile iterasyon sayısının sabit kalması durumunda, bulunan pareto çözümü sayısı artış göstermektedir. Bununla birlikte çözüm süresi 5000 iterasyon için makul görünmemektedir. Farklı başlangıç popülasyonu değerlerine göre elde edilen pareto çözümleri Şekil 4.6’da karşılaştırmalı olarak verilmiştir.



Şekil 4.6: Popülasyon büyüklüğüne göre pareto çözümleri grafiği.

Popülasyon büyüklüğünün artışıyla bulunan pareto çözüm değerleri bir önceki sonuçlara göre hem sayı bazında artış göstermiş hem de değer olarak daha iyi çözümler sunmuştur. Çözümlerin daha rahat anlaşılabilmesi için Şekil 4.7’de üç boyutlu grafikler verilmiştir.



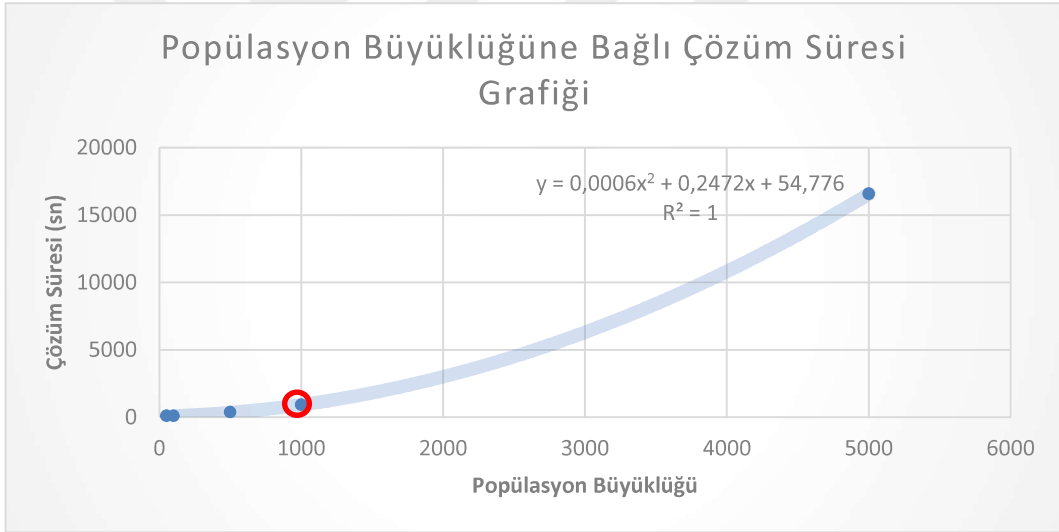
Şekil 4.7: Başlangıç popülasyonuna göre üç boyutlu pareto çözümleri grafiği.

Yapılan analizlerin çözüm süreleri Tablo 4,7’de gösterilmiştir

Tablo 4.7: Başlangıç popülasyonuna göre analiz süreleri.

Pcross: 0,70 Pmut: 0,05		
Psize	Dur (sec)	İter
50	62,877491	500
100	79,218286	500
500	355,616098	500
1000	899,741472	500
5000	16551,89971	500

Sezgisel metotlar, deterministtik metotlarla üstel çözüm sürelerinde çözülen problemlere, polinomial çözüm süreleri içinde çözüm sunan yaklaşımlar geliştirmiştir. Yapılan analizler sonucunda görülmüştür ki nüfus büyüklüğüne bağlı olarak çözüm süresi polinomial bir fonksiyona birebir uygun olarak artmaktadır. Polinomial grafiğe birebir uygunluk Şekil 4.8’de görülmektedir.



Şekil 4.8: Popülasyon büyüklüğüne bağlı çözüm süresi grafiği.

4.2. İTERASYON SAYISININ PARETO ÇÖZÜMÜNE ETKİSİ

Popülasyon sayısının pareto çözümüne olan etkisi incelendikten sonra, Psize sabit tutularak iterasyon sayısındaki artışın pareto çözümü üzerine etkileri incelenmiştir. Analizlerde Tablo 4.8’deki parametreler kullanılmıştır.

Tablo 4.8: İterasyon sayısının etkisini ölçmek için kullanılan MGA parametreleri.

MGA PARAMETRELERİ	
Çaprazlama oranı (Pc)	0,75
Mutasyon oranı (Pm)	0,05
Başlangıç Popülasyonu (Psize)	100
Run sayısı	30
İterasyon sayısı (It)	50 100 500 1000 5000

Yapılan analizler sonucunda bulunan pareto değerleri ve buna bağlı opsiyon bileşenleri aşağıda başlıklar halinde incelenmiştir.

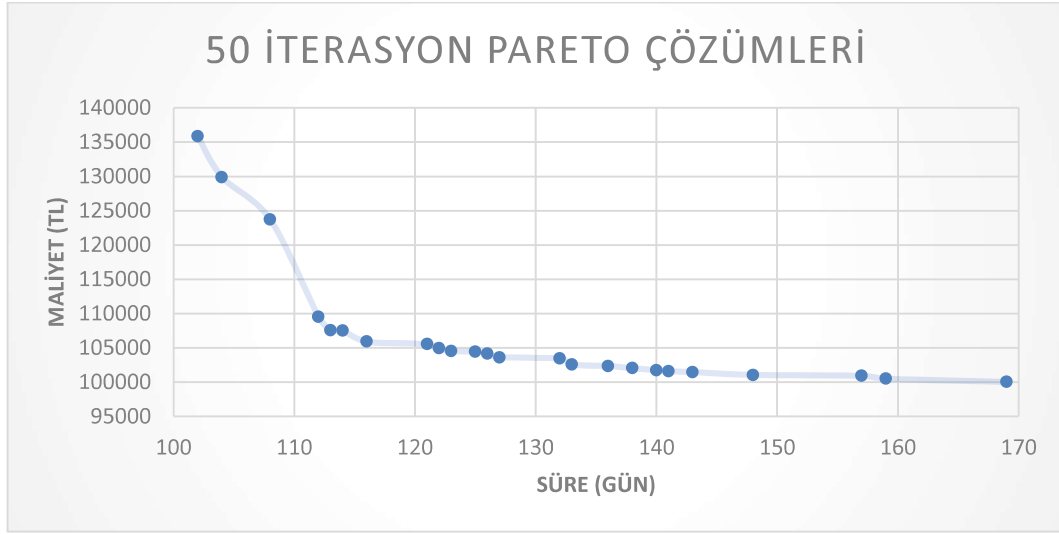
4.2.1. 50 İterasyon ile Çözüm

Başlangıç popülasyonu 100 alınarak 50 iterasyon ilerletilen analizin pareto çözümleri Tablo 4.9'de verilmiştir.

Tablo 4.9: 50 iterasyon sayısı ile bulunan pareto çözümleri.

		Psize=100 Pcross=0.70 Pmut=0.05 Run=30 İter=50 Dur=52,874224 sec																	
SÜRE	MALİYET	AKTİVİTELER																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
102	135870	1	5	3	3	3	1	2	1	1	1	2	2	3	2	1	4	1	1
104	129898	2	3	3	3	4	1	3	4	2	1	2	1	3	3	1	2	1	1
108	123720	3	5	1	3	4	2	2	5	1	1	1	2	3	3	1	3	1	1
112	109520	3	3	3	3	4	3	3	5	1	1	3	1	1	3	1	3	1	1
113	107560	1	4	3	3	4	3	3	5	2	1	2	1	3	1	1	5	1	1
114	107520	1	5	2	3	4	3	3	5	1	1	2	1	2	3	2	5	1	1
116	105920	3	4	3	3	4	3	3	5	1	1	1	1	3	3	2	4	1	1
121	105558	2	5	3	3	4	3	3	4	1	1	1	1	3	2	1	4	2	2
122	104945	2	5	3	3	4	3	3	2	3	2	3	1	3	3	2	5	1	1
123	104520	2	5	3	3	4	3	3	5	1	1	3	2	3	2	1	3	3	1
125	104458	2	2	3	3	4	3	3	4	1	1	1	1	3	3	2	5	3	1
126	104170	1	5	3	3	4	3	3	5	1	1	2	2	3	3	2	4	2	2
127	103610	1	4	3	3	4	3	3	1	2	1	3	1	3	3	2	5	3	1
132	103460	2	5	3	3	4	3	3	1	2	2	1	1	3	3	2	4	3	1
133	102560	1	5	3	3	4	3	3	5	2	1	2	3	3	3	2	5	3	1
136	102360	2	5	3	3	4	3	3	1	2	1	3	4	3	3	2	4	3	1
138	102045	4	5	3	3	4	3	3	2	3	1	1	2	3	3	2	5	3	1
140	101745	2	5	3	3	4	3	3	2	3	2	3	4	3	3	2	5	3	1
141	101610	3	5	3	3	4	3	3	5	2	2	3	2	3	3	2	5	3	2
143	101460	5	5	3	3	4	3	3	5	2	1	2	1	3	2	2	5	3	2
148	101060	4	5	3	3	4	3	3	5	2	1	2	4	3	3	2	4	3	2
157	100940	2	5	3	3	4	3	3	5	4	3	3	4	3	3	2	5	3	2
159	100520	5	5	3	3	4	3	3	5	5	1	1	3	3	3	2	5	3	3
169	100040	5	5	3	3	4	3	3	1	5	3	3	4	3	2	2	5	3	3

Yapılan analizler sonucunda 100 başlangıç popülasyonu 50 iterasyon ilerletildiğinde 24 adet pareto değeri bulunmuştur. Analiz süresi 53 saniye sürmüştür. Pareto cephesine ait grafik Şekil 4.9'da gösterilmiştir.



Şekil 4.9: 50 iterasyon ile elde edilen pareto çözümleri grafiği.

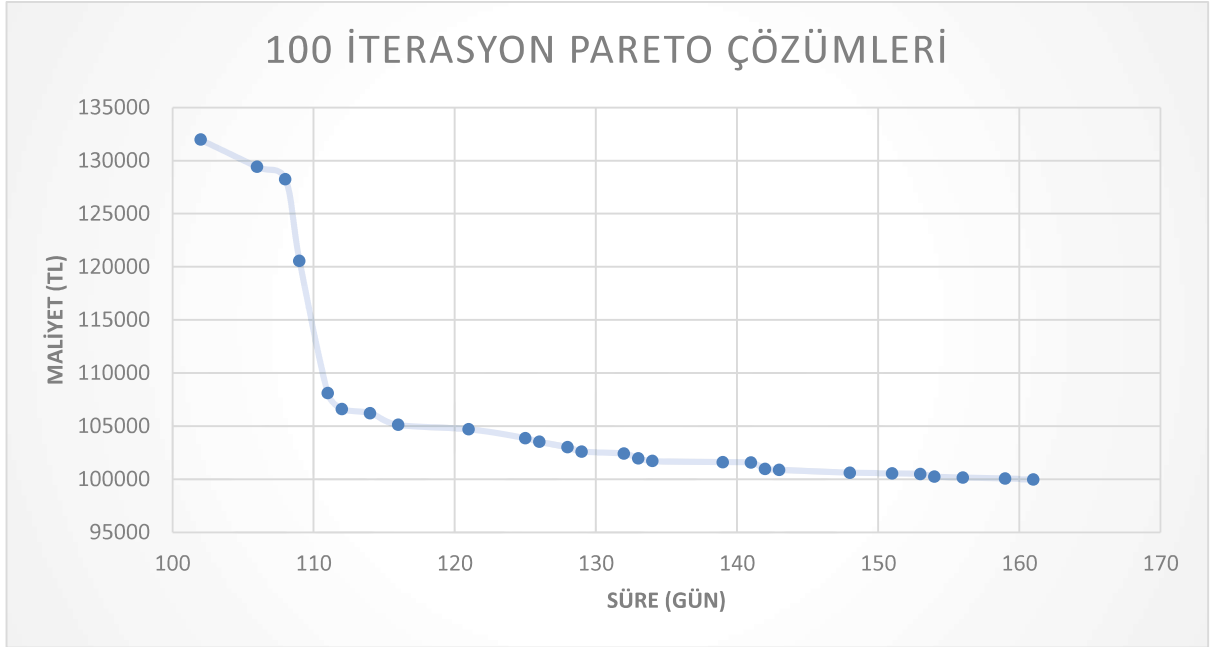
4.2.2. 100 İterasyon ile Çözüm

Başlangıç popülasyonu 100 alınarak 100 iterasyon ile gerçekleştirilen analizin pareto çözümleri Tablo 4.10'da verilmiştir.

Tablo 4.10: 100 iterasyon sayısı ile bulunan pareto çözümleri.

		Psize=100 Pcross=0.70 Pmut=0.05 Run=30 İter=100 Dur=60.768311 sec																	
SÜRE	MALİYET	AKTİVİTELER																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
102	131970	2	5	3	3	4	1	2	1	1	1	2	1	3	1	1	2	1	1
106	129415	3	3	2	3	3	2	3	2	1	1	1	1	3	1	1	1	1	1
108	128250	2	4	3	3	4	1	3	3	1	1	1	3	3	3	1	5	1	1
109	120560	3	5	3	3	4	2	3	1	2	1	2	1	3	2	1	4	1	1
111	108108	2	5	2	3	4	3	3	4	1	1	3	1	3	2	1	2	1	1
112	106608	3	5	3	3	4	3	3	4	1	1	2	1	3	2	1	4	1	1
114	106220	1	5	2	3	4	3	3	5	1	1	1	1	3	3	2	5	1	1
116	105108	3	5	3	3	4	3	3	4	1	1	2	1	3	2	2	5	1	1
121	104700	3	5	3	3	4	3	3	5	3	1	2	1	3	3	2	5	1	1
125	103858	2	5	3	3	4	3	3	4	1	1	3	1	3	2	2	3	3	1
126	103508	3	5	3	3	4	3	3	4	1	1	2	1	3	1	2	5	3	1
128	103005	2	5	3	3	4	3	3	2	2	1	1	1	3	3	2	5	3	1
129	102598	3	5	3	3	4	3	3	4	2	1	3	1	3	3	2	5	3	1
132	102400	1	5	3	3	4	3	3	3	1	1	2	4	3	2	2	5	3	1
133	101950	2	5	3	3	4	3	3	3	1	1	2	4	3	3	2	5	3	1
134	101700	3	5	3	3	4	3	3	3	1	1	2	4	3	3	2	5	3	1
139	101588	3	5	3	3	4	3	3	4	3	1	2	4	3	3	2	5	3	1
141	101550	3	5	3	3	4	3	3	1	3	2	2	4	3	3	2	5	3	1
142	100958	5	5	3	3	4	3	3	4	1	1	3	4	3	3	2	5	3	1
143	100858	3	5	3	3	4	3	3	4	1	1	3	4	3	3	2	5	3	3
148	100600	5	5	3	3	4	3	3	3	1	1	2	4	3	2	2	5	3	2
151	100540	5	5	3	3	4	3	3	3	2	1	2	4	3	2	2	5	3	2
153	100488	5	5	3	3	4	3	3	4	3	1	2	4	3	2	2	5	3	2
154	100240	5	5	3	3	4	3	3	3	2	1	1	4	3	3	2	5	3	3
156	100150	5	5	3	3	4	3	3	5	3	1	3	4	3	2	2	5	3	3
159	100050	5	5	3	3	4	3	3	3	4	1	2	4	3	3	2	5	3	3
161	99970	5	5	3	3	4	3	3	5	5	2	1	4	3	3	2	5	3	3

Yapılan analizler sonucunda 100 başlangıç popülasyonu 100 iterasyon ilerletildiğinde 27 adet pareto değeri bulunmuştur. Analiz süresi 61 saniye sürmüştür. Pareto cephesine ait grafik Şekil 4.10'da gösterilmiştir.



Şekil 4.10: 100 iterasyon ile elde edilen pareto çözümleri grafiği.

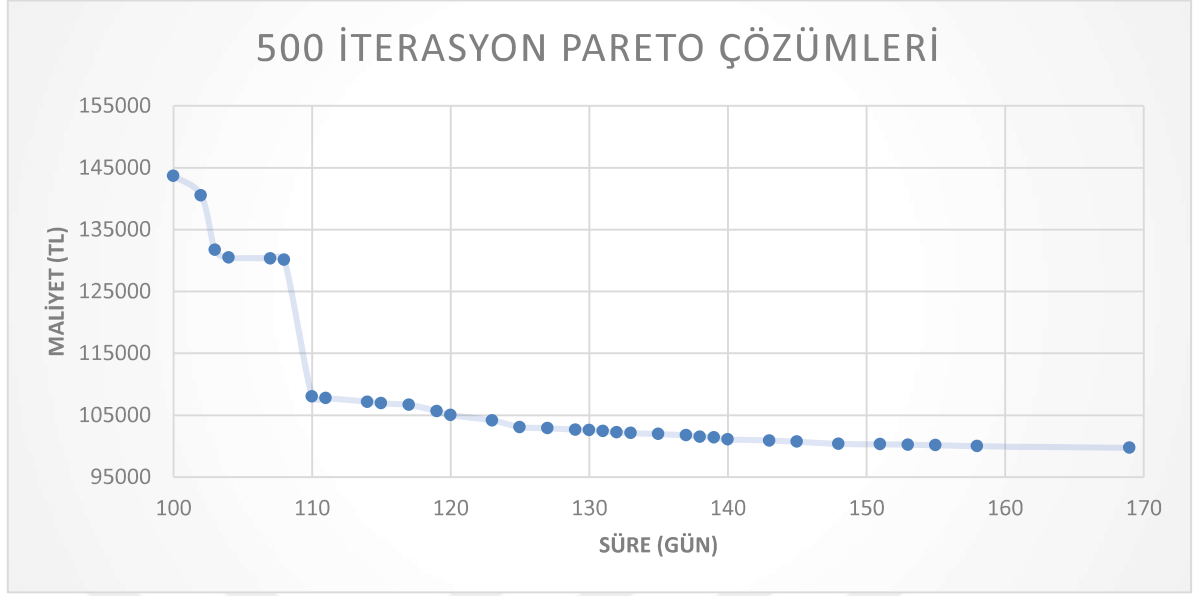
4.2.3. 500 İterasyon ile Çözüm

Başlangıç popülasyonu 100 alınarak 500 iterasyon ilerletilen analizin pareto çözümleri Tablo 4.11'da verilmiştir.

Tablo 4.11: 500 iterasyon sayısı ile bulunan pareto çözümleri.

Psize=100 Pcross=0.70 Pmut=0.05 Run=30 İter=500 Dur= 96,976867sec																			
SÜRE	MALİYET	AKTİVİTELER																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
100	143708	1	4	3	2	2	1	2	4	1	1	1	1	3	2	1	5	1	1
102	140515	3	4	1	2	3	1	3	2	1	1	1	1	3	2	1	4	1	1
103	131720	2	5	3	3	4	1	2	1	1	1	3	1	2	2	1	5	1	1
104	130490	2	5	3	3	4	1	3	3	2	1	2	1	1	2	1	5	1	1
107	130340	1	3	3	3	4	1	2	3	2	1	2	1	3	2	2	5	1	1
108	130120	2	1	3	3	4	1	3	5	1	1	2	2	3	1	2	4	1	1
110	108020	1	5	3	3	4	3	3	1	1	1	2	1	2	2	1	5	1	1
111	107770	2	5	3	3	4	3	3	1	1	1	2	1	2	2	1	5	1	1
114	107170	1	5	2	3	4	3	3	5	1	1	1	1	3	3	1	3	2	1
115	106920	2	5	3	3	4	3	3	1	1	1	2	1	3	1	1	3	2	1
117	106660	1	4	3	3	4	3	3	1	2	1	1	1	3	2	2	4	1	1
119	105660	3	5	2	3	4	3	3	5	2	1	1	1	3	3	2	5	1	1
120	105020	1	5	2	3	4	3	3	5	1	1	1	1	3	3	1	5	3	1
123	104160	1	5	3	3	4	3	3	5	2	1	1	1	3	3	1	5	3	1
125	103070	2	5	3	3	4	3	3	5	1	1	2	1	3	2	2	5	3	1
127	102908	2	5	3	3	4	3	3	4	1	1	2	2	3	2	2	5	3	1
129	102660	3	5	3	3	4	3	3	5	2	1	1	1	3	3	2	5	3	1
130	102620	1	5	3	3	4	3	3	5	1	1	1	1	3	3	2	5	3	2
131	102420	4	5	3	3	4	3	3	5	1	1	2	1	3	2	2	5	3	1
132	102215	3	5	3	3	4	3	3	2	1	1	1	1	3	3	2	5	3	2
133	102120	2	5	3	3	4	3	3	5	1	1	1	2	3	3	2	5	3	2
135	101960	3	5	3	3	4	3	3	5	2	1	2	1	3	3	2	5	3	2
137	101755	3	5	3	3	4	3	3	2	2	1	3	2	3	3	2	5	3	2
138	101520	1	5	3	3	4	3	3	5	1	1	2	4	3	3	2	5	3	2
139	101400	4	5	3	3	4	3	3	3	1	1	1	4	3	3	2	5	3	1
140	101108	3	5	3	3	4	3	3	4	1	1	2	4	3	3	2	5	3	2
143	100910	3	5	3	3	4	3	3	5	2	1	3	4	3	3	2	5	3	2
145	100720	4	5	3	3	4	3	3	5	1	1	1	4	3	3	2	5	3	2
148	100370	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	3
151	100298	5	5	3	3	4	3	3	4	2	1	3	4	3	3	2	5	3	2
153	100238	5	5	3	3	4	3	3	4	3	1	3	4	3	3	2	5	3	2
155	100160	5	5	3	3	4	3	3	5	2	2	3	4	3	3	2	5	3	2
158	99995	5	5	3	3	4	3	3	2	3	2	3	4	3	3	2	5	3	3
169	99740	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3

Yapılan analizler sonucunda 100 başlangıç popülasyonu 500 iterasyon ilerletildiğinde 34 adet pareto değeri bulunmuştur. Analiz süresi 97 saniye sürmüştür. Pareto cephesine ait grafik Şekil 4.11'de gösterilmiştir.



Şekil 4.11: 500 iterasyon ile elde edilen pareto çözümleri grafiği.

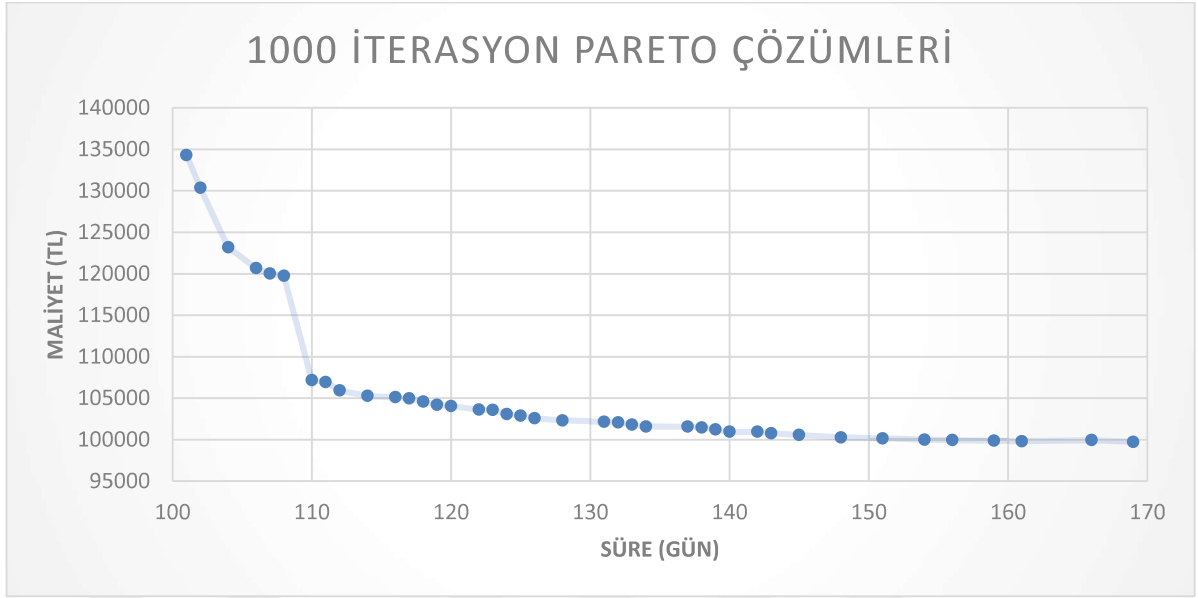
4.2.4. 1000 İterasyon ile Çözüm

Başlangıç popülasyonu 100 alınarak 1000 iterasyon ilerletilen analizin pareto çözümleri Tablo 4.12'da verilmiştir.

Tablo 4.12: 1000 iterasyon sayısı ile bulunan pareto çözümleri.

Psize=100 Pcross=0.70 Pmut=0.05 Run=30 iter=1000 Dur= 126,630960sec																			
SÜRE	MALİYET	AKTİVİTELER																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
101	134315	1	3	2	3	4	1	2	2	1	1	1	1	3	2	1	1	1	1
102	130350	2	5	3	3	4	1	2	3	1	1	2	1	3	2	1	5	1	1
104	123170	1	5	3	3	4	2	2	1	1	1	3	1	3	1	1	5	1	1
106	120665	3	5	3	3	4	2	3	2	1	1	2	1	3	3	1	3	1	1
107	120020	2	5	3	3	4	2	3	1	1	1	1	2	3	3	1	5	1	1
108	119750	3	5	3	3	4	2	3	3	1	1	1	2	3	3	1	5	1	1
110	107170	1	5	3	3	4	3	3	1	1	1	3	1	3	1	1	5	1	1
111	106958	2	4	3	3	4	3	3	4	1	1	1	1	3	2	1	5	1	1
112	105920	3	5	3	3	4	3	3	1	1	1	2	1	3	3	1	5	1	1
114	105270	1	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	1	1
116	105120	3	5	3	3	4	3	3	1	1	1	2	1	3	3	1	5	2	1
117	104970	2	5	3	3	4	3	3	5	1	1	3	2	3	3	1	5	2	1
118	104600	3	5	3	3	4	3	3	3	1	1	3	2	3	3	2	5	1	1
119	104220	2	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	2	1
120	104058	3	5	3	3	4	3	3	4	1	1	3	1	3	3	2	5	2	1
122	103620	3	5	3	3	4	3	3	5	1	1	2	1	3	3	1	5	3	1
123	103570	2	5	3	3	4	3	3	5	1	1	3	2	3	3	1	5	3	1
124	103070	1	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
125	102900	2	5	3	3	4	3	3	3	1	1	3	1	3	3	2	5	3	1
126	102570	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
128	102320	3	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	3	1
131	102170	4	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
132	102070	3	5	3	3	4	3	3	1	1	1	3	1	3	3	2	5	3	2
133	101820	2	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
134	101570	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
137	101560	3	5	3	3	4	3	3	5	2	1	2	4	3	3	2	5	3	1
138	101470	3	5	3	3	4	3	3	5	1	1	3	3	3	3	2	5	3	2
139	101220	2	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
140	100970	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
142	100950	5	5	3	3	4	3	3	3	1	1	3	4	3	3	2	5	3	1
143	100770	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	3
145	100570	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
148	100270	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
151	100150	5	5	3	3	4	3	3	3	1	1	3	4	3	3	2	5	3	3
154	100010	5	5	3	3	4	3	3	5	2	1	3	4	3	3	2	5	3	3
156	99950	5	5	3	3	4	3	3	5	3	1	3	4	3	3	2	5	3	3
159	99870	5	5	3	3	4	3	3	5	4	2	3	4	3	3	2	5	3	3
161	99820	5	5	3	3	4	3	3	5	5	2	3	4	3	3	2	5	3	3
166	99940	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	2
169	99740	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3

Yapılan analizler sonucunda 100 başlangıç popülasyonu 1000 iterasyon ilerletildiğinde 40 adet pareto değeri bulunmuştur. Analiz süresi 127 saniye sürmüştür. Pareto cephesine ait grafik Şekil 4.12’de gösterilmiştir.



Şekil 4.12: 1000 iterasyon ile elde edilen pareto çözümleri grafiği.

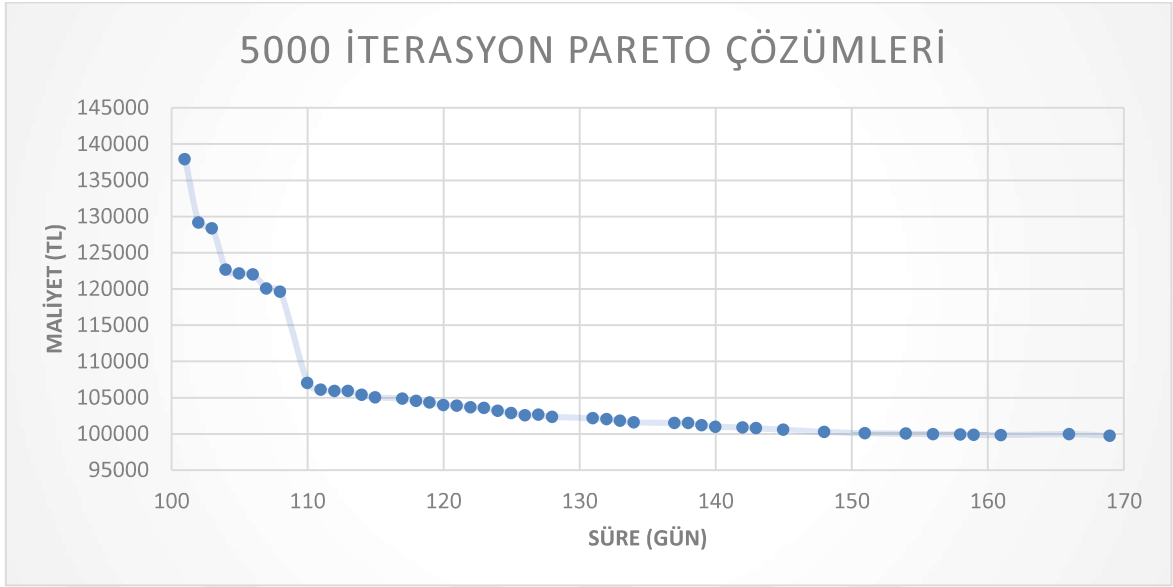
4.2.5. 5000 İterasyon ile Çözüm

Başlangıç popülasyonu 100 alınarak 5000 iterasyon ile gerçekleştirilen analizin pareto çözümleri Tablo 4.13'te verilmiştir.

Tablo 4.13: 5000 iterasyon sayısı ile bulunan pareto çözümleri.

Psize=100 Pcross=0.70 Pmut=0.05 Run=30 İter=5000 Dur= 523,615602 sec																			
SÜRE	MALİYET	AKTİVİTELER																	
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
101	137920	1	2	3	3	4	1	1	1	1	1	1	3	3	1	5	1	1	
102	129170	2	5	2	3	4	1	3	1	1	1	2	1	3	2	1	5	1	1
103	128365	1	5	3	3	4	1	3	2	1	1	2	2	3	2	1	5	1	1
104	122670	1	5	1	3	4	2	3	5	1	1	1	1	3	2	1	3	1	1
105	122115	1	5	2	3	4	2	3	2	1	1	3	1	3	2	1	3	1	1
106	121970	3	2	3	3	4	2	3	5	1	1	2	1	3	3	1	3	1	1
107	120058	2	5	3	3	4	2	3	4	1	1	3	2	3	2	1	5	1	1
108	119608	1	5	3	3	4	2	3	4	1	1	2	1	3	3	1	5	2	1
110	107020	1	4	3	3	4	3	3	1	1	1	1	1	3	3	1	5	1	1
111	106070	2	5	3	3	4	3	3	5	1	1	2	1	3	3	1	5	1	1
112	105908	3	5	3	3	4	3	3	4	1	1	2	1	3	3	1	5	1	1
113	105900	2	5	3	3	4	3	3	3	1	1	2	2	3	3	1	5	1	1
114	105365	1	5	3	3	4	3	3	2	1	1	3	1	3	3	2	5	1	1
115	105020	2	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	1	1
117	104865	2	5	3	3	4	3	3	2	1	1	3	2	3	3	2	5	1	1
118	104520	3	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	1	1
119	104308	2	5	3	3	4	3	3	4	1	1	3	1	3	3	2	5	2	1
120	103970	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	2	1
121	103870	2	5	3	3	4	3	3	5	1	1	2	1	3	3	1	5	3	1
122	103650	3	5	3	3	4	3	3	3	1	1	3	1	3	3	1	5	3	1
123	103570	2	5	3	3	4	3	3	5	1	1	3	2	3	3	1	5	3	1
124	103150	1	5	3	3	4	3	3	3	1	1	3	1	3	3	2	5	3	1
125	102870	2	5	3	3	4	3	3	5	1	1	2	1	3	3	2	5	3	1
126	102570	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
127	102620	2	5	3	3	4	3	3	5	1	1	2	2	3	3	2	5	3	1
128	102320	3	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	3	1
131	102170	4	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
132	102020	3	5	3	3	4	3	3	5	1	1	2	1	3	3	2	5	3	2
133	101820	2	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
134	101570	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
137	101510	3	5	3	3	4	3	3	5	2	1	3	4	3	3	2	5	3	1
138	101470	1	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
139	101170	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
140	100970	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
142	100870	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
143	100770	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	3
145	100570	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
148	100270	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
151	100070	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	3
154	100010	5	5	3	3	4	3	3	5	2	1	3	4	3	3	2	5	3	3
156	99950	5	5	3	3	4	3	3	5	3	1	3	4	3	3	2	5	3	3
158	99900	5	5	3	3	4	3	3	5	3	2	3	4	3	3	2	5	3	3
159	99870	5	5	3	3	4	3	3	5	4	2	3	4	3	3	2	5	3	3
161	99820	5	5	3	3	4	3	3	5	5	2	3	4	3	3	2	5	3	3
166	99940	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	2
169	99740	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3

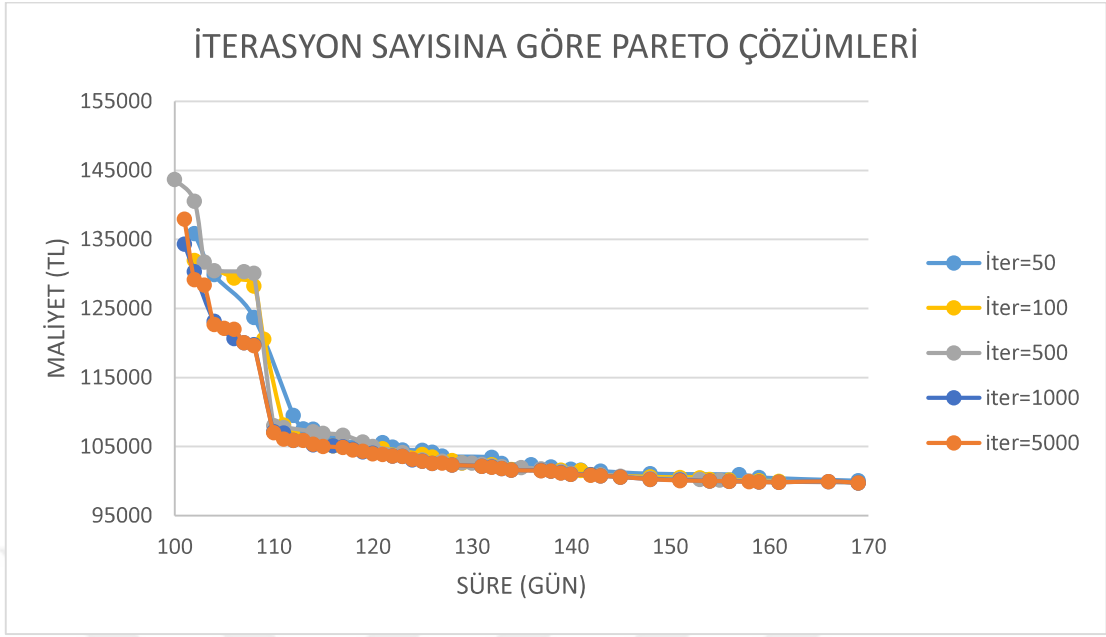
Yapılan analizler sonucunda 100 başlangıç popülasyonu 1000 iterasyon ilerletildiğinde 46 adet pareto değeri bulunmuştur. Analiz süresi 524 saniye sürmüştür. Pareto cephesine ait grafik Şekil 4.13'te gösterilmiştir.



Şekil 4.13: 5000 iterasyon ile elde edilen pareto çözümleri grafiği.

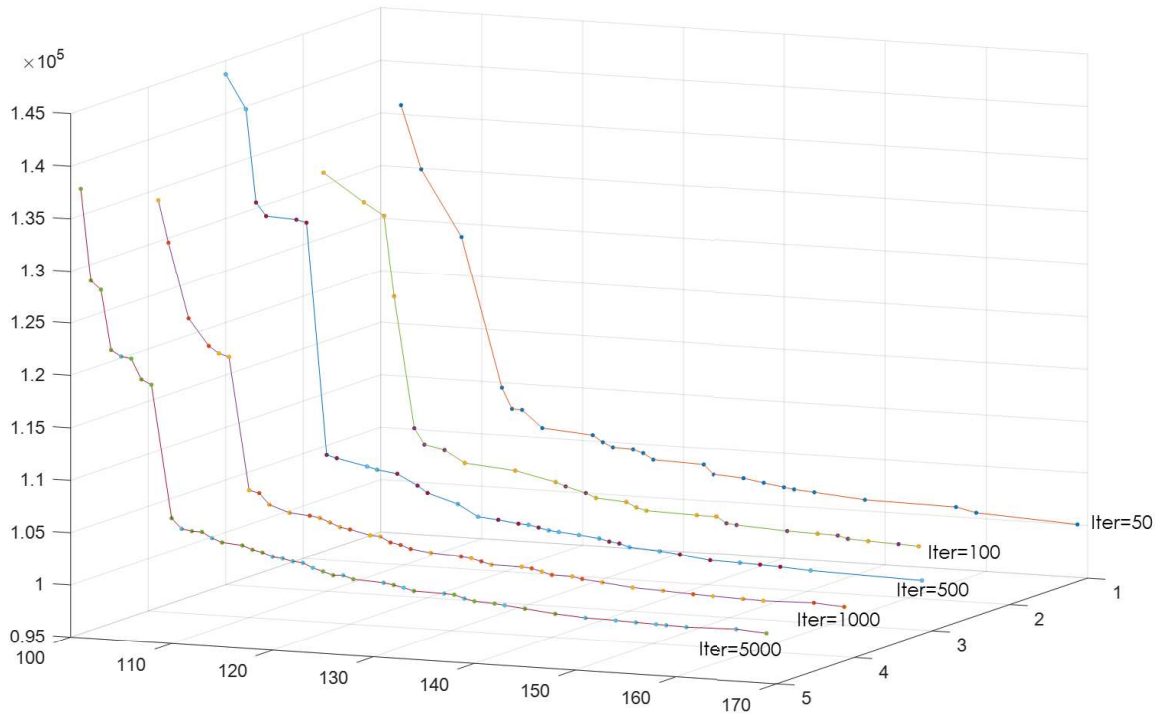
4.2.6. İterasyon Sayısına Göre Çözümlerin Karşılaştırılması

Yapılan analizler sonucunda popülasyon büyüklüğü sabit kalmak üzere iterasyon sayısının artması durumunda, bulunan pareto çözümleri sayısı artış göstermektedir. Farklı iterasyon sayısı değerlerine göre elde edilen pareto çözümleri Şekil 4.14'te karşılaştırmalı olarak verilmiştir.



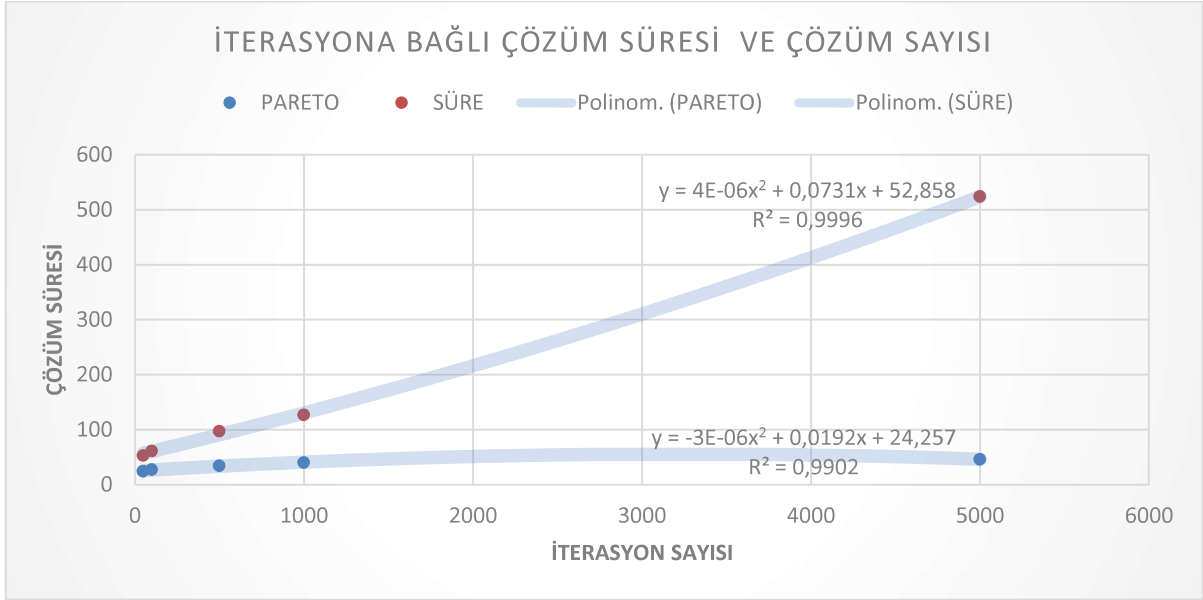
Şekil 4.14: İterasyon sayısına göre değişen pareto çözümleri grafiği.

İterasyon sayısının artışıyla bulunan pareto çözüm değerleri bir önceki sonuçlara göre hem sayı bazında artış göstermiş hem de değer olarak daha iyi çözümler sunmuştur. Çözümlerin daha rahat anlaşılabilmesi için Şekil 4.15'te üç boyutlu grafikler verilmiştir.



Şekil 4.15: İterasyon sayısına göre değişen üç boyutlu pareto çözümleri grafiği.

Yapılan analizlerin çözüm süreleri Şekil 4.16’da grafik halinde gösterilmiştir. Çözüm süresinin yine polinomial zamana uyduğu gözlemlenmiştir. Gerek çözüm süresi, gerekse pareto çözümü sayısının polinomial fonksiyonlara %99 oranında güvenilirlikle uyması ayrıntılı incelenmesi gereken bir durumdur.



Şekil 4.16: İterasyon sayısına bağlı analiz sürelerinin karşılaştırılması.

4.3. MGA VE GA YAKLAŞIMLARININ KARŞILAŞTIRILMASI

Popülasyon büyüklüğü ve iterasyon sayısına bağlı pareto çözümlerindeki değişimler gözlemlendikten sonra, makul sürelerde iyi çözümler verebilecek parametreler belirlenmiştir. Belirlenen parametrelerle hem GA hem de önerilen MGA yaklaşımı ile elde edilen sonuçlar karşılaştırılmıştır. Tablo 4.14’te analiz parametreleri verilmiştir.

Tablo 4.14: GA – MGA analizleri için parametreler.

GA - MGA PARAMETRELERİ	
Çaprazlama oranı (Pc)	0,75
Mutasyon oranı (Pm)	0,05
Başlangıç Popülasyonu (Psize)	500
Run sayısı	30
İterasyon sayısı (It)	5000

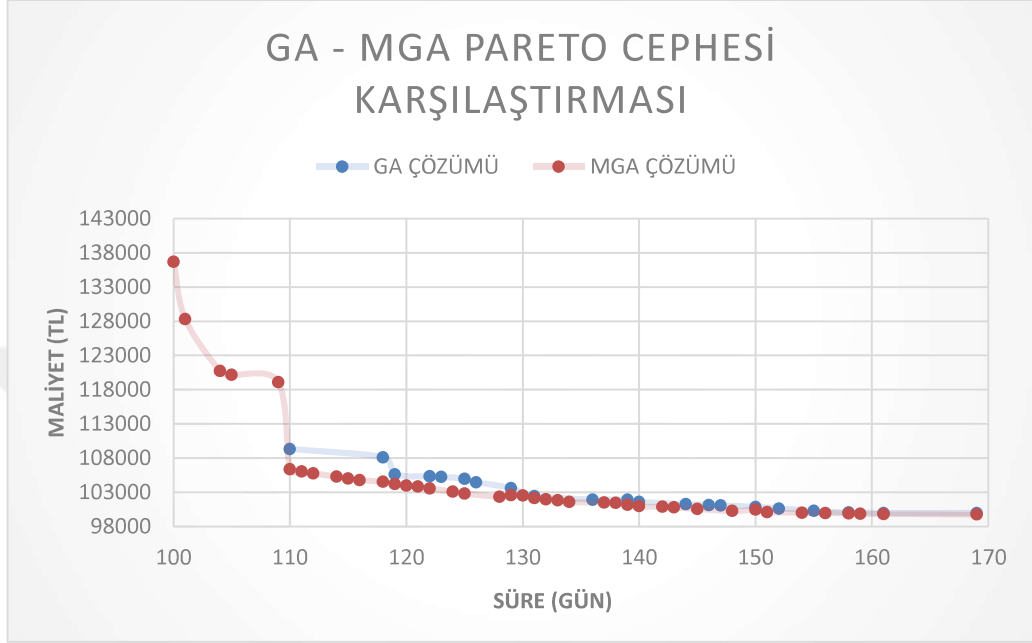
Yapılan analizler sonucunda GA ve MGA yaklaşımlarının verdiği pareto değerleri Tablo 4.15’te karşılaştırmalı olarak verilmiştir.

Tablo 4.15: GA ve MGA yaklaşımı ile analiz pareto sonuçları.

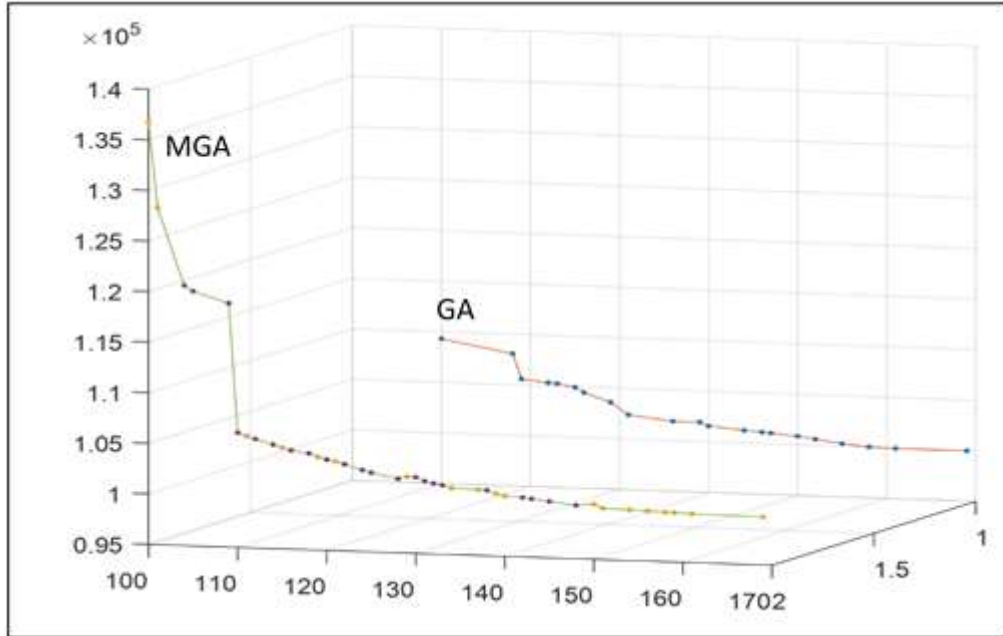
GA ÇÖZÜM		MGA ÇÖZÜM	
DUR	COST	DUR	COST
-	-	100	136708
-	-	101	128320
-	-	104	120715
-	-	105	120170
-	-	109	119108
110	109315	110	106350
-	-	111	106020
-	-	112	105770
-	-	114	105270
-	-	115	105020
-	-	116	104770
118	108070	118	104520
119	105590	119	104220
-	-	120	103970
-	-	121	103820
122	105320	122	103570
123	105258	-	-
-	-	124	103070
125	104948	125	102820
126	104465	-	-
-	-	128	102320
129	103590	129	102560
-	-	130	102510
131	102408	131	102170
-	-	132	101970
-	-	133	101820
-	-	134	101570
136	101920	-	-
-	-	137	101510
-	-	138	101470
139	101915	139	101170
140	101570	140	100970
-	-	142	100870
-	-	143	100770
144	101250	-	-
-	-	145	100570
146	101138	-	-
147	101070	-	-
-	-	148	100270
150	100850	150	100450
-	-	151	100070
152	100620	-	-
-	-	154	100010
155	100270	-	-
-	-	156	99950
158	100030	158	99900
-	-	159	99870
161	99965	161	99820
169	99970	169	99740

Tablo 4.15'te görüldüğü üzere GA yaklaşımı 21 pareto sonucu verirken MGA yaklaşımı ile 41 pareto sonucu elde edilmiştir. GA yaklaşımında bulunan sonuçlardan MGA'da karşılığı olan tüm sonuçlar daha küçük maliyet değerlerine tekabül etmektedir. Dolayısıyla daha iyi pareto

çözümlerinin elde edildiği aşikârdır. Farkın daha rahat anlaşılabilmesi için GA ile MGA yaklaşımı sonucunda bulunan pareto değerleri Şekil 4.17 ve Şekil 4.18’de ayrıntılı olarak verilmiştir.



Şekil 4.17: GA – MGA pareto cephesi karşılaştırması.



Şekil 4.18: GA – MGA pareto cephesi üç boyutlu grafik karşılaştırması.

Tablo 4.16'da MGA metoduyla bulunan tüm pareto değerleri ve bu değerlere karşılık gelen mod bileşenleri gösterilmiştir. İlk üç mod bileşeni için kritik faaliyetler matris metotla Tablo 4.17 - 4.21 arasında gösterilmiştir.

Tablo 4.16: MGA pareto çözümleri ve opsiyon bileşenleri.

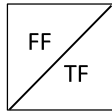
DUR	COST	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
100	136708	1	2	3	3	3	1	3	4	1	1	1	1	3	1	1	2	1	1
101	128320	1	5	3	3	4	1	3	5	1	1	2	1	3	3	1	5	1	1
104	120715	1	5	3	3	4	2	3	2	1	1	1	1	3	2	1	5	1	1
105	120170	2	5	3	3	4	2	3	5	1	1	1	1	3	3	1	5	1	1
109	119108	2	5	3	3	4	2	3	4	1	1	3	1	3	3	2	5	1	1
110	106350	1	5	3	3	4	3	3	3	1	1	3	1	3	3	1	5	1	1
111	106020	2	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	1	1
112	105770	3	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	1	1
114	105270	1	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	1	1
115	105020	2	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	1	1
116	104770	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	1	1
118	104520	1	5	3	3	4	3	3	5	1	1	2	1	3	3	2	5	2	1
119	104220	2	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	2	1
120	103970	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	2	1
121	103820	2	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	3	1
122	103570	3	5	3	3	4	3	3	5	1	1	3	1	3	3	1	5	3	1
124	103070	1	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
125	102820	2	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
128	102320	3	5	3	3	4	3	3	5	1	1	3	2	3	3	2	5	3	1
129	102560	3	5	3	3	4	3	3	5	2	1	2	1	3	3	2	5	3	1
130	102510	2	5	3	3	4	3	3	5	2	1	3	2	3	3	2	5	3	1
131	102170	4	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	1
132	101970	3	5	3	3	4	3	3	5	1	1	3	1	3	3	2	5	3	2
133	101820	2	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
134	101570	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
137	101510	3	5	3	3	4	3	3	5	2	1	3	4	3	3	2	5	3	1
138	101470	3	5	3	3	4	3	3	5	1	1	3	3	3	3	2	5	3	2
139	101170	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
140	100970	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
142	100870	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	1
143	100770	3	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	3
145	100570	4	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
148	100270	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	2
150	100450	4	5	3	3	4	3	3	5	3	1	3	4	3	3	2	5	3	2
151	100070	5	5	3	3	4	3	3	5	1	1	3	4	3	3	2	5	3	3
154	100010	5	5	3	3	4	3	3	5	2	1	3	4	3	3	2	5	3	3
156	99950	5	5	3	3	4	3	3	5	3	1	3	4	3	3	2	5	3	3
158	99900	5	5	3	3	4	3	3	5	3	2	3	4	3	3	2	5	3	3
159	99870	5	5	3	3	4	3	3	5	4	2	3	4	3	3	2	5	3	3
161	99820	5	5	3	3	4	3	3	5	5	2	3	4	3	3	2	5	3	3
169	99740	5	5	3	3	4	3	3	5	5	3	3	4	3	3	2	5	3	3

4.4. MGA SONUÇLARININ CPM HESAPLAMALARI

Tablo 4.17’de Tablo 4.16’da verilen ilk opsiyon bileşenine karşılık gelen CPM hesaplamaları yapılmıştır. Proje süresi 100 gün ve maliyeti 136708 TL olarak bulunmuştur. Kırmızı ile işaretlenmiş alanlar kritik faaliyetleri göstermektedir. Buna göre 1-3-5-6-12-15-17-18 faaliyetleri kritik faaliyetlerdir. Projenin ilerleyişi esnasında bu faaliyetlerde bir aksama olursa projenin bitim süresinde de aksaklık olacaktır.

Tablo 4.17: 100-136708 pareto çözümünün CPM hesaplamaları.

OPT	0	1	2	3	3	3	1	3	4	1	1	1	1	3	1	1	2	1	1	PROJE SÜRESİ	
ACT	St	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	100	
DUR	0	14	18	33	20	28	14	18	21	15	15	12	22	24	9	12	22	14	9	PROJE MALİYETİ	
COST	0	2400	2400	3200	30000	15000	40000	22000	208	300	450	450	2000	1800	3000	4500	2000	4000	3000	136708	
ES	0	0	0	0	0	14	14	42	28	28	28	60	60	33	43	65	57	77	91	EF	ACT
																				0	St
																				14	1
																				18	2
																				33	3
																				20	4
																				42	5
																				28	6
																				60	7
																				49	8
																				43	9
																				43	10
																				72	11
																				65	12
																				57	13
																				52	14
																				77	15
																				79	16
																				91	17
																				100	18



Tablo 4.18’de Tablo 4.16’da verilen ikinci opsiyon bileşenine karşılık gelen CPM hesaplamaları yapılmıştır. Proje süresi 101 gün ve maliyeti 128320 TL olarak bulunmuştur. Kırmızı ile işaretlenmiş alanlar kritik faaliyetleri göstermektedir. Buna göre 1-5-6-7-10-11-12-15-17-18 faaliyetleri kritik faaliyetlerdir. Projenin ilerleyişi esnasında bu faaliyetlerde bir aksama olursa projenin bitim süresinde de aksaklık olacaktır.

Tablo 4.18: 101-128320 pareto çözümünün CPM hesaplamaları.

OPT	0	1	5	3	3	4	1	3	5	1	1	2	1	3	3	1	5	1	1	PROJE SÜRESİ	
ACT	St	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	101	
DUR	0	14	25	33	20	30	14	18	24	15	16	22	24	18	12	30	14	9	PROJE MALİYETİ		
COST	0	2400	1000	3200	30000	10000	40000	22000	120	300	450	350	2000	1800	2200	4500	1000	4000	3000	128320	
ES	0	0	0	0	0	14	14	44	28	28	28	62	62	33	43	66	61	78	92	EF	ACT
	0	0	0	0	0															0	St
	0	0	3	0	23															14	1
	0	0				0	0													25	2
	3	3				0	0				3									33	3
	0	3	3								3									20	4
	23	23												0						44	5
						0								18						28	6
						0								18						62	7
						0	0	0						0						52	8
						0	0	0						0						43	9
						0	0	0						0						43	10
						0	0	0						0						78	11
						0	0	0						0						66	12
						0	0	0						0						57	13
						0	0	0						0						61	14
						0	0	0						0						78	15
						0	0	0						0						91	16
						0	0	0						0						92	17
						0	0	0						0						101	18

Tablo 4.19’de Tablo 4.16’da verilen üçüncü opsiyon bileşenine karşılık gelen CPM hesaplamaları yapılmıştır. Proje süresi 104 gün ve maliyeti 120715 TL olarak bulunmuştur. Kırmızı ile işaretlenmiş alanlar kritik faaliyetleri göstermektedir. Buna göre 1-5-6-10-12-15-17-18 faaliyetleri kritik faaliyetlerdir. Projenin ilerleyişi esnasında bu faaliyetlerde bir aksama olursa projenin bitim süresinde de aksaklık olacaktır.

Tablo 4.19: 104-120715 pareto çözümünün CPM hesaplamaları.

OPT	0	1	5	3	3	4	2	3	2	1	1	1	1	3	2	1	5	1	1	PROJE SÜRESİ	
ACT	St	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	104	
DUR	0	14	25	33	20	30	18	18	15	15	15	12	22	24	15	12	30	14	9	PROJE MALİYETİ	
COST	0	2400	1000	3200	30000	10000	32000	22000	215	300	450	450	2000	1800	2400	4500	1000	4000	3000	120715	
ES	0	0	0	0	0	14	14	44	32	32	32	62	62	33	47	69	62	81	95	EF	ACT
	0	0	0	0	0															0	St
	0	0	7	0	27															14	1
	0	0				0	0													25	2
	7	7									7									33	3
			7								7									20	4
				7										0						44	5
					7									5						32	6
						27								27						62	7
							27							27						47	8
								0						18						47	9
								0						18						74	11
								0						0						69	12
								0						0						57	13
								0						0						62	14
								0						0						81	15
								0						0						92	16
								0						0						95	17
								0						0						104	18

Tablo 4.20’de Tablo 4.16’da verilen dördüncü opsiyon bileşenine karşılık gelen CPM hesaplamaları yapılmıştır. Proje süresi 105 gün ve maliyeti 120170 TL olarak bulunmuştur. Kırmızı ile işaretlenmiş alanlar kritik faaliyetleri göstermektedir. Buna göre 1-5-6-10-12-14-15-16-17-18 faaliyetleri kritik faaliyetlerdir. Projenin ilerleyişi esnasında bu faaliyetlerde bir aksama olursa projenin bitim süresinde de aksaklık olacaktır.

Tablo 4.20: 105-120170 pareto çözümünün CPM hesaplamaları.

OPT	0	2	5	3	3	4	2	3	5	1	1	1	1	3	3	1	5	1	1	PROJE SÜRESİ	
ACT	St	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	105	
DUR	0	15	25	33	20	30	18	18	24	15	15	12	22	24	18	12	30	14	9	PROJE MALİYETİ	
COST	0	2150	1000	3200	30000	10000	32000	22000	120	300	450	450	2000	1800	2200	4500	1000	4000	3000	120170	
ES	0	0	0	0	0	15	15	45	33	33	33	63	63	33	48	70	66	82	96	EF	ACT
	0	0	0	0	0															0	St
	0	0	8	0	28															15	1
		0				0	0													25	2
			8							8										33	3
				0										0						20	4
					28										28					45	5
						0		0						18						33	6
							0		0	0				6	0	0				63	7
								0						7						57	8
									6					13						48	9
										0				15						48	10
											15			15						75	11
														0						70	12
															0					57	13
																9				66	14
																0	16			82	15
																0	0			96	16
																	0	0		96	17
																		0	0	105	18

Tablo 4.21’de Tablo 4.16’da verilen dördüncü opsiyon bileşenine karşılık gelen CPM hesaplamaları yapılmıştır. Proje süresi 109 gün ve maliyeti 119108 TL olarak bulunmuştur. Kırmızı ile işaretlenmiş alanlar kritik faaliyetleri göstermektedir. Buna göre 1-5-6-10-12-15-17-18 faaliyetleri kritik faaliyetlerdir. Projenin ilerleyişi esnasında bu faaliyetlerde bir aksama olursa projenin bitim süresinde de aksaklık olacaktır.

Tablo 4.21: 109-119108 pareto çözümünün CPM hesaplamaları.

OPT	0	2	5	3	3	4	2	3	4	1	1	3	1	3	3	2	5	1	1	PROJE SÜRESİ	
ACT	St	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	109	
DUR	0	15	25	33	20	30	18	18	21	15	15	20	22	24	18	16	30	14	9	PROJE MALİYETİ	
COST	0	2150	1000	3200	30000	10000	32000	22000	208	300	450	300	2000	1800	2200	3500	1000	4000	3000	119108	
ES	0	0	0	0	0	15	15	45	33	33	33	63	63	33	48	70	66	86	100	EF	ACT
	0	0	0	0	0															0	St
	0	0	8	0	28															15	1
	0	0			0	0														25	2
	8	8								8										33	3
			0											0						20	4
			9											9						45	5
			28											18						33	6
			28											18						63	7
					0									0						54	8
					0									3						48	9
					0									3						48	10
					0									0						83	11
					0									0						70	12
					0									0						57	13
					0									9						66	14
					0									13						86	15
					0									0						96	16
					0									4						100	17
					0									4						109	18

5. TARTIŞMA VE SONUÇ

Bu çalışma kapsamında inşaat projelerinde zaman-maliyet ödünleşim problemi incelenmiştir. Problemi çözmek için evrimsel tabanlı meta sezgisellerden en bilineni olan Genetik Algoritma kullanılmış, evrimsel stratejilerin çözüm algoritmasından esinlenerek, MAWA ağırlıklandırma yönteminin pareto çözümlerini kaçırmaması yönündeki handikapını gidermek üzere algorithmada modifikasyona gidilmiştir.

Amaçlanan çözüm, her bir süre değerine karşılık gelen minimum maliyetleri bulmak ve bu maliyetler arasında da pareto cephesini oluşturan, baskılanamayan (nondominated) çözümlerin elde edilmesini sağlamaktır.

MGA uygulanmadan önce popülasyon büyüklüğü ve iterasyon sayısının, analiz süresi ve pareto değerlerinin sayısına etkisi incelenmiştir. Hem popülasyon büyüklüğü, hem de iterasyon sayısının artması pareto çözümlerini nicelik ve niteliksel yönden geliştirmiştir. Popülasyon büyüklüğündeki artışla beraber çözüm süresindeki kayda değer büyüme GA-MGA karşılaştırma analizi için başlangıç popülasyonu adedinin 500 ile sınırlandırılmasına neden olmuştur. İterasyon sayısındaki artışın çözüm süresine etkisi kabul edilebilir boyutlarda olduğundan iyi çözümler aramak için iterasyon sayısı 5000 olarak alınmıştır.

Analizle literatürden alınan çaprazlama oranı, mutasyon oranı değerleri ile yapılan çalışmalardan elde edilen sonuçlara göre belirlenen popülasyon büyüklüğü ve iterasyon sayısı değerleri kullanılarak, istatistiksel açıdan anlam ifade etmesi için 30 koşulda gerçekleştirilmiştir. Bulunan pareto çözümleri değerleri MGA'da 41 sonuç verirken, GA'da 21 sonuçta kalmıştır. Ayrıca MGA çözümlerinin GA çözümlerine maliyet bazında da üstün geldiği görülmüştür.

Deterministik yöntemlerin üstel çözüm sürelerine ulaşması sebebiyle çözemediği NP-Zor tipi gerçek hayat problemlerine yönelik polinomial çözüm zamanına sahip metasezgisel metotların kullanımı, makul sürelerde iyi çözümler sunduklarından proje yöneticilerinin karar vermelerinde çok önemli bir yere sahiptir.

Bu analiz sonucunda proje yöneticilerinin doğru kararlar verebilmesi için pareto çözümlerinin bulunmasına yönelik bir matlab kodu yazılmış, literatürde sıklıkla kullanılan test problemi ile

test edilerek doğruluđu teyit edilmiş ve geliştirilen modifiye algoritma ile makul sürelerde daha iyi sonuçlar elde edilmiştir.

İlerleyen aşamalarda bu çalışmadan elde edilen sonuçlar farklı modifikasyonlarla geliştirilebilir, diğer sezgisel algoritmaların güçlü yönleri ile birleştirilerek hibrit modeller elde edilebilir, şantiye şeflerinin karar mekanizmalarına yardımcı olması için planlama araçlarına entegre edilebilir. Diğer algoritmalarla karşılaştırılarak verimlilik yönünden değerlendirmeye tabi tutulup çalışmanın ileri seviye analizlere temel teşkil etmesi sağlanabilir.



KAYNAKLAR

- Adrian, J. J. (1979). *Construction Accounting: Financial, Managerial, Auditing and Tax*. Reston, Virginia: Prentice-Hall.
- Afruz, E. N., Roghanian, E., Nejafi, A. A., & Mazinani, M. (2013). A multi-mode resource-constrained discrete time–cost tradeoff problem solving using an adjusted fuzzy dominance genetic algorithm. *Scientia Iranica E*, 20(3), 931-944.
- Afshar, A., Kaveh, A., & Shoghli, O. R. (2007). Multi-objective Optimization of Time-Cost-Quality. *Asian Journal of Civil Engineering (Building and Housing)*, 8(2), 113-124.
- Alataş, B. (2007). Kaotik Haritalı Parçacık Sürü Optimizasyon Algoritmaları Geliştirme. Fırat Üniversitesi Fen Bilimleri Enstitüsü.
- Albayrak, B. (1998). *Proje Yönetimi ve Danışmanlık* (2. b.). İstanbul: Alfa Basım Yayın Dağıtım.
- Albayrak, B. (2009). *Proje Yönetimi ve Analizi*. Ankara: Nobel Yayın Dağıtım.
- Albayrak, G., & Özdemir, İ. (2016). Yapı Projelerinin Süre-Maliyet Optimizasyonunda Metasezgisel Algoritma Kullanımı. *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, 5, 39-49.
- Atlı, Ö. (2012). *Bulanık Çok Modlu Kaynak Kısıtlı Proje Çizelgeleme Problemlerinin Çözümü İçin Matematiksel Bir Model*. Hava Harp Okulu Havacılık ve Uzay Teknolojileri Enstitüsü: Doktora Tezi.
- Azaron, A., Perkgoz, C., & Sakawa, M. (2005). A Genetic Algorithm Approach for the Time-Cost Trade-off in PERT Networks. *Applied Mathematics and Computation*, 168, 1317–1339.
- Azeez, M. N., & Alsaffar, A. (2014). Construction Time-Cost Optimization Modeling Using Ant Colony Optimization. *Journal of Engineering*, 20(1), 114-131.
- Aziz, R. F. (2013). RPERT: Repetitive-Projects Evaluation and Review Technique. *Alexandria Engineering Journal*, 53, 81-93.
- Bettemir, Ö. H. (2009). Optimization of Time-Cost-Resource Trade-Off Problems in Project Scheduling Using Meta-Heuristic Algorithms. *Orta Doğu Teknik Üniversitesi*.
- Bettemir, Ö. H. (2010). Experimental Design for Genetic Algorithm Simulated Annealing for Time Cost Trade-Off Problems. *International Journal of Engineering & Applied Sciences (IJEAS)*, 3(1), 15-26.
- Blazewics, J., & Kobler, D. (2002). Review of properties of different precedence graphs for scheduling problems. *European Journal of Operational Research*, 142, 435-443.

- Burke, R. (2003). *Project Management Planning and Control Techniques* (4. b.). West Sussex: Wiley Chichester.
- Coello, C. C. (2000). An Updated Survey of GA-Based Multiobjective Optimization techniques. *ACM Comput. Surv.*, 32(2), 109-143.
- Coello, C. C., Veldhuizen, D. V., & Lamont, G. B. (2002). *Evolutionary Algorithms for Solving Multi-Objective Problems* (1. b.). New York: Springer Science + Business Media.
- Dasgupta, D., & Michalewicz, Z. (1997). *Evolutionary Algorithms in Engineering Applications* (1 b.). New York: Springer.
- Elbeltagi, E. (2009). Lecture Notes on Construction Project Management. Mansoura, Egypt: Mansoura University Faculty of Engineering. <http://osp.mans.edu.eg/elbeltagi/Management.html> adresinden alındı
- Elbeltagi, E., Hegazy, T., & Grierson, D. (2005). Comparison among Five Evolutionary-Based Optimization Algorithms. *Advanced Engineering Informatics*, 19, 43-53.
- Feng, C. W., Liu, L., & Burns, S. A. (1997). Using Genetic Algorithms To Solve Construction Time-Cost Trade-Off Problems. *Journal of Computing in Civil Engineering*, 11(3), 184-189.
- Feng, C. W., Liu, L., & Burns, S. A. (2000). Stochastic Construction Time-Cost Trade-Off Analysis. *Journal Of Computing In Civil Engineering*, 117-126.
- Feng, C.-W., Liu, L., & Burns, S. A. (1997). Using Genetic Algorithms to Solve Construction Time-Cost Trade-Off Problems. *Journal of Computing in Civil Engineering*, 11(3), 184-189.
- Geem, Z. W. (2010). Multiobjective Optimization of Time-Cost Trade-Off Using Harmony Search. *Journal of Construction Engineering and Management*, 136(6), 711-716.
- Gen, M., & Cheng, R. (2000). *Genetic Algorithms & Engineering Optimization* (1. b.). New York: Wiley-Interscience.
- Ghods, R., Skandari, M. R., Allahverdiloo, M., & Iranmanesh, S. H. (2009). A New Practical Model to Trade-off Time, Cost, and Quality of a Project. *Australian Journal of Basic and Applied Sciences*, 3(4), 3741-3756.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning* (2 b.). Canada: Addison-Wesley Publishing Company, Inc.
- Goldberg, D. E., & Richardson, J. (1987). Genetic algorithms with sharing for multimodel function optimization. *Proc. Second International Conference on Genetic Algorithms (ICGA '87)*, 218-229.
- Gray, C. F., & Larson, E. W. (2003). *Project Management: The Managerial Process*. USA: Irwin Mc Graw-Hill.

- Hadju, M., (1997). *Network Scheduling Techniques for Construction Project Management*. Budapeşte: Kluwer Academic Publishers.
- Haque, K. M., & Hasin, M. A., (2012). Genetic Algorithm for Project Time-Cost Optimization in Fuzzy Environment. *Journal of Industrial Engineering and Management*, 5(2), 364-381.
- Haupt, R. L., & Haupt, S. E., (2004). *Practical Genetic Algorithms* (2. b.). Hoboken, New Jersey: John Wiley & Sons, Inc.
- Hegazy, T., (1999). Optimization of construction time-cost trade-off analysis using genetic algorithms. *Canadian Journal of Civil Engineering*, 26(6), 685-697.
- Hegazy, T., (2002). *Computer-based Construction Project Management*. New Jersey: Prentice Hall.
- Karaboğa, D., & Görkemli, D. (2011). A Combinatorial Artificial Bee Colony Algorithm for Traveling Salesman Problem. *International Symposium on Innovations in Intelligent Systems and Applications*, 50-53.
- Kaya, S., & Fiğlalı, N., (2016). Çok Amaçlı Optimizasyon Problemlerinde Pareto Optimal Kullanımı. *Social Sciences Research Journal*, 5(2), 2147-5237.
- Ke, H., Ma, W., & Ni, Y., (2009). Optimization models and a GA-based algorithm for stochastic time-cost trade-off problem. *Applied Mathematics and Computation*, 215, 308–313.
- Kerzner, H., (2009). *Project Management: A System Approach to planning, scheduling and controlling* (10. b.). New Jersey: John Wiley & Sons, Inc.
- Klein, R., (2000). *Scheduling of Resource-Constrained Projects*. Boston: Kluwer Academic Publishers.
- Konak, A., Coit, D. W., & Smith, A. E., (2006). Multi-Objective Optimization Using Genetic Algorithms: A Tutorial. *Reliability Engineering & System Safety*, 91(9), 992-1007.
- Koza, J. (1992). *Genetic programming: on the programming of computers by means of natural selection*. Cambridge: MIT Press.
- Lacouture, D., ASCE, A. M., Süer, G. A., Joaqui, J. G., & Yates, J. K., (2009). Construction Project Scheduling with Time, Cost, and Material Restrictions Using Fuzzy Mathematical Models and Critical Path Method. *Journal of Construction Engineering and Management*, 135(10(1906)), 733-9364.
- Levy, S. M., (2012). *Project Management in Construction* (6. b.). New York: Mc Graw Hill.
- Li, H., & Love, P., (1997). Using Improved Genetic Algorithms to Facilitate Time-Cost Optimization. *Journal of Construction Engineering and Management*, 123(3), 233-237.
- Manisalı, E., Okuyan, E., & Giran, Ö., (2000). CPM İnşaat Planlamasında Etkin Bir Çözüm Yolu: Matris Yöntem. 2. *Yapı İşletmesi Kongresi*.

- Mendes, J. M., (2015). Multiobjective Genetic Algorithm-Based for Time-Cost Optimization. *New Developments in Pure and Applied Mathematics*, 88-95.
- Meredith, J. R., & Mantel, S. J., (2009). *Project Management: A Managerial Approach*. USA: John Wiley & Sons Inc.
- Miettinen, K., Neittaanmaki, P., Makela, M. M., & Periaux, J., (1999). *Evolutionary Algorithms in Engineering and Computer Science*. Chichester: John Wiley & Sons Ltd.
- Moder, J. J., Philips, C. R., & Davis, E. W., (1983). *Project Management with CPM, PERT and Precedence Diagraming* (3. b.). New York: Van Nostrand Reinhold.
- Mubarak, S., (2010). *Construction Project Scheduling and Control* (2. b.). New Jersey: John Wiley & Sons.
- Namlı, E., (2012). *Proje Yönetimi Kapsamında Risk Tabanlı ve Yapay Zeka Destekli bir Maliyet Tahmin Modeli*. İstanbul Üniversitesi Fen Bilimleri Enstitüsü: Doktora Tezi.
- Paksoy, S., (2007). *Genetik Algoritma ile Proje Çizelgeleme*. Çukurova Üniversitesi Sosyal Bilimler Enstitüsü: Doktora Tezi.
- Park, W. R., & Chapin, W. B., (1992). *Construction Bidding: Strategic Pricing for Profit*. New York: John Wiley & Sons.
- Parveen, S., & Saha, S. K., (2012). GA Based Multi-Objective Time-Cost Optimization in a Project with Resources Consideration. *International Journal of Modern Engineering Research (IJMER)*, 2(6), 4352-4359.
- PMI., (2008). *A Guide To The Project Management Body Of Knowledge* (4 b., Cilt 5). Pennsylvania: Project Management Institute.
- Robert, K. W., & Gary, R. M., (2003). *Effective Project Management* (3. b.). Indiana: Wiley Publishing.
- Roofpanah, H., & Hassanlou, K., (2013). A Probabilistic Approach for Project Cost Estimation Using Bayesian Networks. *Life Science Journal*, 10(6), 342-349.
- Sanchez, E., Squilero, G., & Tonda, A., (2012). *Industrial Applications of Evolutionary Algorithms*. New York: Springer.
- Sakawa, M., (2002). *Genetic Algorithms and Fuzzy Multiobjective Optimization*, Kluwer Academic Publishers.
- Sarı, T., (2008). *Metasezgisel Yöntemlerle Proje Çizelgeleme Optimizasyonu*. İstanbul: Marmara Üniversitesi, Sosyal Bilimler Enstitüsü.
- Shrivastava, R., Singh, S., & Dubey, G. C., (2012). Multi Objective Optimization of Time Cost Quality Quantity Using Multi Colony Ant Algorithm. *Int. J. Contemp. Math. Sciences*, 7(16), 773 - 784.

- Sorrentino, M., (2013). Genetic Algorithms for Construction Time-Cost-Quality Trade-Off: A Road Project Case Study. *Construction Management*, 2, 163-176.
- Stevens, D. J., (1990). *Techniques for Construction Network Scheduling* (international b.). Kentucky: McGraw-Hill.
- Taha, H. A., (2007). *Operations Research: an Introduction* (8. b.). New Jersey: Pearson Prentice Hall.
- Tutunji, T. A., (2010). Week 7 Project Management Lecture Note. Jordan: Philadelphia University.
- Uğur, L. O., Erdal, M., & Baykan, U. N., (2006). Yapım İşleri Genel Şartnamesi'nde İşveren ve Yüklenicinin Sorumluluk Paylaşımının Proje Maliyetine Etkisi. *Selçuk Üniversitesi Teknik Online Dergi*, 5(3), 133-149.
- Ulusoy, G., (2002). Proje Planlamada Kaynak Kısıtlı Çizelgeleme. *Savancı Üniversitesi Mühendislik ve Doğa Bilimleri Fakültesi Dergisi*, 8, 23-29.
- Winston, W. L., (2004). *Operations Research Applications and Algorithms* (4. b.). Canada, USA: Brooks/Cole—Thomson Learning.
- Wysocki, R. K., & McGary, R., (2003). *Effective Project Management* (3. b.). Indianapolis: Wiley Publishing Inc.
- Xiong, Y., & Kuang, Y., (2008). Applying an ant colony optimization algorithm-based multiobjective approach for time-cost trade-off. *J. Constr. Eng. Manage.*, 134(2), 153–156.
- Yang, I. T., (2007). Using elitist particle swarm optimization to facilitate bicriterion time-cost trade-off analysis. *J. Constr. Eng. Manage.*, 133(7), 498–505.
- Yang, X.-S., (2010). *Engineering Optimization: An Introduction with Metaheuristic Applications*. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Young, T. L., (1998). *How to be a Better Project Manager*. London: Kogan Page.
- Yu, X., & Gen, M., (2010). *Introduction to Evolutionary Algorithms*. New York: Springer.
- Zadeh, L. A., (1965). Fuzzy Sets. *Information and Control*, 8(3), 338-353.
- Zheng, D. X., Ng, S. T., & Kumaraswamy, M. M., (2004). Applying a genetic algorithm-based multiobjective approach for time-cost optimization. *J. Constr. Eng. Manage.*, 130(2), 168–176.
- Zheng, D. X., Ng, T. S., & Kumaraswamy, M. M., (2002). Applying Genetic Algorithm Techniques For Time - Cost Optimization. *18th Annual ARCOM Conference*, 2(801-10).
- Zitzler, E., Deb, K., & Thiele, L., (2000). Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation Journal*, 8(2), 125-148.

ÖZGEÇMİŞ

Kişisel Bilgiler	
Adı Soyadı	Çağrı UYSAL
Doğum Yeri	KEŞAN
Doğum Tarihi	06.10.1990
Uyruğu	<input checked="" type="checkbox"/> T.C. <input type="checkbox"/> Diğer:
Telefon	0(545) 545 53 85
E-Posta Adresi	cagriuyosal90@gmail.com
Web Adresi	-



Eğitim Bilgileri	
Lisans	
Üniversite	İstanbul Üniversitesi
Fakülte	Mühendislik Fakültesi
Bölümü	İnşaat Mühendisliği
Mezuniyet Yılı	08.06.2012

Yüksek Lisans	
Üniversite	İstanbul Üniversitesi
Enstitü Adı	Fen Bilimleri Enstitüsü
Anabilim Dalı	İnşaat Mühendisliği
Programı	İnşaat Mühendisliği