
GAZİ ÜNİVERSİTESİ
Gazi
1926

# YAPAY ZEKA TEKNİKLERİ KULLANARAK YAZILIM PROJE YÖNETİM SÜREÇLERİNİ İYİLEŞTİRME

**Nurhan GÜL**

**DOKTORA TEZİ**
**BİLGİSAYAR MÜHENDİSLİĞİ ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ**
**FEN BİLİMLERİ ENSTİTÜSÜ**

**TEMMUZ 2024**

## ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmasında;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmasında yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Nurhan GÜL
17/07/2024

# YAPAY ZEKA TEKNİKLERİ KULLANARAK YAZILIM PROJE YÖNETİM SÜREÇLERİNİ İYİLEŞTİRME

(Doktora Tezi)

Nurhan GÜL

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Temmuz 2024

## ÖZET

Yazılım projelerinin başarıya ulaşma oranı teknolojik gelişmelere rağmen hala istenen düzeyde değildir. Yazılım projelerinin büyük çoğunluğu ya istenen özelliklerde teslim edilememekte ya da planlanan bütçe ve zaman sınırını aşarak teslim edilebilmektedir. Bu durum, zaman, para, insan kaynağı gibi kayıpların yanı sıra itibar gibi ticari kayıplara da neden olabilmektedir. Bu çalışmada, yazılım projelerinin minimum sürede ve minimum bütçe ile gerçekleştirilebilmesi için proje yöneticilerine yardımcı olacak proje çizelgeleme yöntemleri geliştirilmiştir. Yazılım projelerinde insan kaynaklarının görevlere etkin bir şekilde atanması sağlanarak, projelerin tamamlanma süresi ve bütçesi en aza indirilmiştir. İnsan kaynaklarının görev atamaları öncesinde yaşayabilecekleri boşta bekleme süreleri azaltılarak, insan kaynaklarının etkin kullanımının proje süresi ve bütçesine etkileri analiz edilmiştir. Optimizasyon süreçlerinde yapay zekâ algoritmalarından genetik algoritma, bozkurt optimizasyon algoritması ve yapay arı kolonisi algoritmasının kullanıldığı hibrit bir model tasarlanmıştır. Yöntemlerdeki rastgelelik kaotik fonksiyonlar aracılığıyla gerçekleştirilmiştir. Popülasyon çeşitliliğinde çaprazlama ve mutasyon oranları bulanık mantık süreçleri ile belirlenerek, durum uzayında arama süreçlerinin iyileştirilmesi sağlanmıştır. Geliştirilen yöntemlerin farklı büyüklükte durum uzayına sahip problemlerde etkin sonuçlar ürettiği ve literatürdeki çalışmalarla karşılaştırıldığında %13'e varan iyileştirmeler sağladığı görülmüştür.

| | | |
|---|---|---|
| Bilim Kodu | : | 92432 |
| Anahtar Kelimeler | : | Kaynak kısıtlı yazılım proje çizelgeleme problemi, genetik algoritma, bozkurt optimizasyonu, yapay arı kolonisi algoritması |
| Sayfa Adedi | : | 105 |
| Danışman | : | Prof. Dr. Nursal ARICI |

# IMPROVING SOFTWARE PROJECT MANAGEMENT PROCESSES USING ARTIFICIAL INTELLIGENCE TECHNIQUES

(Ph. D. Thesis)

Nurhan GÜL

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

July 2024

## ABSTRACT

Despite technological advancements, the success rate of software projects is still not at the desired level. Most software projects either fail to deliver the desired features or exceed the planned budget and time limits. This situation causes losses in time, money, human resources, and reputation. In this study, project scheduling methods that will assist project managers in achieving software projects in the minimum time and budget have been developed. By effectively assigning human resources to tasks, the completion time and budget of projects are minimized. Idle times that human resources may experience before task assignments are reduced, and the effects of the efficient use of human resources on project duration and budget are analyzed. A hybrid model using artificial intelligence algorithms, including genetic algorithms, wolf optimization algorithms, and artificial bee colony algorithms, is designed for optimization processes. Randomness in methods is achieved through chaotic functions. In population diversity, crossover and mutation rates are determined by fuzzy logic processes, improving search processes in the state space. The developed methods have been shown to produce effective results in problems with different sizes of solution spaces and have achieved improvements of up to 13% compared to the studies in the literature.

| | | |
|---|---|---|
| Science Code | : | 92432 |
| Key Words | : | Resource-constrained software project scheduling problem, genetic algorithm, grey wolf optimization, artificial bee colony algorithm |
| Page Number | : | 105 |
| Supervisor | : | Prof. Dr. Nursal ARICI |

# TEŞEKKÜR

Bu çalışmanın gerçekleştirilmesinde değerli bilgi ve tecrübelerini cömertçe ve hoşgörüyle aktaran, saygıdeğer danışman hocam Prof. Dr. Nursal ARICI'ya, tez sürecinde yaptıkları akademik yönlendirmeler ve kıymetli katkıları için Prof. Dr. Ahmet COŞAR ve Prof. Dr. Necaattin BARIŞÇI'ya, eğitim öğretim hayatım boyunca bana katkı sağlayan tüm öğretmen ve hocalarıma, bu yolculuğa başlamam için beni teşvik eden eşim Dr. Çağatay GÜL'e, eğitimin her yaşta gerekli olduğunu gerek yetiştirdiği öğrencilerine gerekse biz çocuklarına daima aşılayan babam Alaettin YAVUZ'a, bizim için her zaman elinden geleni yapan annem Fatma YAVUZ'a, bu süreçte manevi destekleriyle yanımda olan ailemin diğer bireylerine ve varlığıyla bana güç veren oğlum Berkay GÜL'e sonsuz teşekkürlerimi sunarım.

# İÇİNDEKİLER

**Sayfa**

# ÇİZELGELERİN LİSTESİ

**Çizelge** **Sayfa**

**Çizelge** **Sayfa**

# ŞEKİLLERİN LİSTESİ

**Şekil** **Sayfa**

# SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

| Kısaltmalar | Açıklamalar |
|---|---|
| **BO** | Bozkurt optimizasyon algoritması |
| **ÇY-KKPÇP** | Çoklu yetenek kaynak kısıtlı proje çizelgeleme |
| **GA** | Genetik algoritma |
| **IEEE** | Elektrik ve Elektronik Mühendisleri Enstitüsü |
| **iMOPSE** | Akıllı çok amaçlı proje planlama ortamı |
| **KKPÇP** | Kaynak kısıtlı proje çizelgeleme problemi |
| **MOGA** | Çok amaçlı genetik algoritma |
| **NP-zor** | Polinomsal zamanda çözülemeyen problem |
| **PGA** | GA temelli yazılım proje çizelgeleme |
| **PGA-B** | PGA'ya bulanık mantık entegre edilmiş çizelgeleme |
| **PHibrit** | Hibrit çizelgeleme |
| **PHibrit-B** | PHibrit'e bulanık mantık entegre edilmiş çizelgeleme |
| **PSO** | Parçacık sürü optimizasyon yöntemi |
| **YAKA** | Yapay arı kolonisi algoritması |

# 1. GİRİŞ

Yazılım projeleri, günümüzde iş dünyasında hızla artan bir öneme sahiptir. Teknolojik gelişmeler, iş süreçlerinin dönüşümü ve dijitalleşme trendleri, şirketlerin rekabet güçlerini artırmak ve pazar taleplerine cevap vermek için yazılım projelerine olan ihtiyacı artırmıştır. Bu yaşanan hızlı gelişmelerle birlikte, yazılım projelerinin belirlenen bütçe ve bitiş tarihinde, müşterinin istediği şekilde başarıyla tamamlanması giderek önemli hale gelmiştir.

Yazılım proje yönetimi, yazılım projelerinin zaman ve bütçe açısından planlanması, organize edilmesi, projede görevlendirilecek insan kaynaklarının yönetimi, risk yönetimi, kalite kontrolü ve proje kapanışı gibi süreçleri kapsayan bir disiplindir ve bu süreçlerin yönetim başarısı, yazılım projelerinin başarılı tamamlanabilmesi açısından önem arz etmektedir [1].

Standish Group tarafından belirli aralıklarla günümüze kadar yayınlanan ve yazılım projelerindeki başarı oranlarını değerlendiren Chaos Raporu'nun ilk versiyonu olan 1994 yılı raporuna göre, 'Başarısız' olan ve tamamlanamayan proje oranı %31, 'Zorluk yaşayan' diğer bir deyişle, belirlenen bütçe ve proje tamamlanma zamanını aşan veya müşterinin istediği kalitede olmadan tamamlanan proje oranı %53, 'Başarılı' proje oranı ise %16 idi. Raporun 2020 yılı verilerine göre ise 'Başarısız' olan ve tamamlanamayan proje oranı %19, 'Zorluk yaşayan' proje oranı %50 ve 'Başarılı' proje oranı ise %31 olarak belirlenmiştir [2-3].

Uzun yıllar geçmiş olmasına rağmen, yazılım projelerinin başarısında iyi bir oran elde edilememesinin nedenleri aşağıda belirtilmiştir:

Karmaşıklık: Yazılım projeleri teknik yapısı ve modüllerin birbiriyle etkileşimlerin ortaya çıkardığı zorluklar nedeniyle karmaşıktır. Bu tür projeler, veri tabanları, kullanıcı ara yüzleri, iş akış yapıları ve dış sistemlerle entegrasyon modülleri gibi farklı bileşenleri içerebilir ve bu bileşenlerin birbirleriyle uyumlu şekilde çalışması ve müşteri gereksinimlerini karşılaması önemlidir. Ayrıca, yazılım projelerinin farklı platformlarda kullanılabilir olma ihtiyacı, uyumluluk ve entegrasyon sorunlarını da beraberinde getirir. Bu durum, diğer proje türleri ile karşılaştırıldığında yazılım projelerinin daha karmaşık olması

sonucunu ortaya çıkarır [1].

Belirsizlik: Yazılım projelerinde belirsizlik yaygın bir sorundur. Müşteri tarafından ortaya konan gereksinimlerin ve hedeflerin net olmaması bu belirsizliğin önemli bir kaynağıdır. Bununla birlikte proje paydaşları arasında yaşanabilecek iletişim eksikliği bu belirsizliği daha da çok arttırır. Eksik iletişim, proje paydaşlarının beklenti ve gereksinimleri doğru şekilde anlamamalarına, belirtilen gereksinimlerle proje çıktılarının tam olarak örtüşmemesine neden olur [4].

Değişkenlik: Yazılım projeleri gereksinimleri gerek müşterinin taleplerini revize etmesi gerek de pazar koşullarının değişmesi sonucu proje süresince değişkenlik gösterebilmektedir. Bu durum, müşterinin başlangıçta net olmayan beklentilerini proje ilerledikçe netleştirmesinden veya proje devam ederken ortaya çıkabilecek teknolojik gelişmeler ve rekabet ortamındaki farklılaşmalardan kaynaklanabilmektedir. Bu nedenle yazılım projeleri, kullanıcı taleplerinde ve teknolojideki değişimlere hızlı bir şekilde uyum sağlamak zorundadır [5].

Soyutluk: Yazılım projeleri, fiziksel ürünlerden ziyade soyut kavramlar üzerine odaklanır ve bu nedenle sonuçları somut olarak tanımlamak zor olabilir [6]. Bu projeler iş süreçlerini otomatize etmek, veri yönetimini sağlamak, bilgi akışını sağlamak gibi işlevleri yerine getirmek için tasarlanan ve genellikle fiziksel olarak tutulamayan veya gözlemlenemeyen ürünler ortaya çıkaran sistemlerdir. Bu durum paydaşlar arasındaki iletişimi zorlaştırabilmektedir.

Yazılım proje yönetiminin başarısız olması durumunda, yazılım projeleri zamanında ve bütçe dahilinde tamamlanamayabilir. Ayrıca, insan kaynaklarının etkin bir şekilde yönetilememesi, risklerin uygun şekilde ele alınamaması ve kalite kontrol süreçlerinin yetersiz olması gibi faktörler, projenin başarısızlıkla sonuçlanmasına neden olabilir. Bu durum, müşteri memnuniyetsizliğine, mali kayıplara, proje paydaşları arasındaki güven kaybına ve projenin tamamen iptal edilmesine kadar çeşitli olumsuz sonuçlara yol açabilir. Dolayısıyla, yazılım projelerinde etkili bir proje yönetimi, projenin başarılı tamamlanabilmesi için kritik öneme sahiptir.

Yazılım proje yönetiminde proje çizelgeleme önemli bir süreçtir. Bu süreçte yazılım

projelerinin her aşaması zaman çizelgesine entegre edilir, görevlerin başlangıç ve bitiş tarihleri belirlenir ve görevlere kaynak tahsis edilir. Proje yöneticileri ve ekipleri için, projenin ilerlemesini etkili bir şekilde yönetme ve takip etme imkânı sunar. Projede olası gecikmelerin önceden belirlenmesi ve risklerin azaltılması adına, projenin her aşamasının zaman çizelgesi üzerinde gösterilmesi, proje yöneticilerine erken müdahale imkânı sağlar. Bu durum projenin zamanında tamamlanmasına ve proje kalitesinin artırılmasına yardımcı olur, dolayısıyla müşteri memnuniyetini de artırır.

İnsan kaynaklarının etkin bir şekilde kullanılması, projenin başarılı bir şekilde yönetilmesi açısından hayati öneme sahiptir. Bu kaynaklar, projenin gereksinimlerini karşılamak ve hedeflenen sonuçlara ulaşmak üzere stratejik olarak kullanılmalıdır. Uygun insan kaynağı planlaması, projenin zaman sınırına uyum sağlayarak insan kaynakların planlanmasını ve görevlere dağıtılmasını sağlar. Ekip üyelerinin yetkinlikleri, deneyimleri ve rolleri, görevlere uygun insan kaynağı atanması sürecinde kilit faktörlerdir. Kaynak atamasının başarılı bir şekilde gerçekleştirilmesi, proje çizelgesinin etkin uygulanmasını ve projenin amaçlanan hedeflere ulaşmasını sağlar. Bu nedenle insan kaynağı ataması proje çizelgeleme sürecinde özel bir öneme sahiptir. Etkili bir çizelgeleme yapılması durumunda kaynak eksikliği veya uyumsuzluğu gibi sorunlar önlenerek proje başarısına katkı sağlanır.

Etkin insan kaynağı yönetimi, yazılım proje çizelgelemesi sürecinin kritik bir bileşenidir. Proje çizelgeleme süreci, hangi insan kaynağının hangi göreve ne zaman atanacağını belirlemek için stratejik kararlar almayı gerektirir. Bu süreç, proje yöneticisi için, projenin tüm olası durumlarını, görev sıralamalarını, kaynak atamalarını ve bu atamaların potansiyel sonuçlarını kapsayan büyük bir durum uzayında seçim yapma zorluğunu ortaya çıkarır. Yazılım projelerinin doğası gereği, bu durumlar arasında etkili seçimler yapmak ve optimal bir proje çizelgesi oluşturmak, manuel yöntemlerle yönetilemeyecek kadar karmaşık hale gelmektedir. Bu nedenle, yapay zekâ ve optimizasyon tekniklerinin kullanımı, yazılım proje çizelgelemesindeki zorlukların üstesinden gelmek için giderek artan bir önem kazanmaktadır.

Yapay zekâ ve optimizasyon teknikleri, projenin durum uzayını etkin bir şekilde araştırarak, görevlerin sıralanması, kaynakların atanması, proje bütçesinin ve proje süresinin minimize edilmesi gibi hedeflere ulaşmayı sağlar. Bu teknikler, çok sayıda değişkeni ve kısıtlamayı dikkate alarak, proje çizelgesinin optimizasyonunu gerçekleştirir.

Bu çalışmada, çoklu yetenek kaynak kısıtlı proje çizelgeleme problemi olarak adlandırılan ve yazılım projelerinde de proje yöneticilerinin etkin bir proje çizelgelemesi yapabilmesini sağlayan problemin çözümü için yapay zekâ tabanlı optimizasyon yöntemleri geliştirilmiştir. Çalışmada gerçek hayattaki projelere uygun veri setleri içeren iMopse veri seti kullanılmıştır [7]. Bozkurt Optimizasyon Algoritması (BO), Genetik Algoritma (GA), Yapay Arı Kolonisi Algoritması (YAKA), kaotik lojistik harita ve bulanık mantık kullanılarak, durum uzayının çok büyük olduğu çoklu yetenek kaynak kısıtlı proje çizelgeleme probleminde (ÇY-KKPÇP) yerel minimuma takılmadan global optimuma ulaşmayı sağlayacak yöntemler geliştirilmiştir. Böylelikle yazılım projelerinde proje tamamlanma süresi, proje bütçesi ve projede görev yapacak insan kaynaklarının görev ataması sürecinde ortaya çıkabilecek boşta bekleme süreleri minimize edilerek, projelerin uygun bütçe ile en kısa sürede tamamlanmasını ve insan kaynaklarının verimli kullanımını sağlayacak çizelgeler oluşturulmuştur. Geliştirilen yöntemler farklı boyutta durum uzayına sahip proje verileri ile test edilerek analiz edilmiş ve literatürdeki yöntemlerle karşılaştırılmıştır. Yapılan deneysel çalışmalarda, önerilen yöntemlerin literatürdeki çalışmalarla karşılaştırıldığında %13'e varan iyileştirmeler sağladığı görülmüştür.

Bu tez çalışmasının başlıca katkıları aşağıda yer almaktadır:

- Gerçek hayattaki yazılım projelerinin minimum sürede ve minimum bütçeyle tamamlanmasını, insan kaynaklarının verimli kullanılmasını sağlayacak hibrit bir model geliştirilmiş ve bu modelle literatüre katkı sağlanmıştır.
- Bu modelin gerçek hayatta uygulanabilmesi için proje yöneticilerinin kullanabileceği bir yazılım ortaya konulmuştur.
- Proje yöneticilerinin farklı yetkinlikteki insan kaynaklarını uygun görevlere ataması ve insan kaynaklarının boşta bekleme sürelerinin azaltılması sağlanmıştır.
- Geliştirilen model ve yazılım üzerinde proje yöneticisinin şirketin beklentileri doğrultusunda farklı optimizasyon yöntemi tercihlerini değişik senaryolar üzerinde uygulayabilme ortamı sağlanmıştır.

Tezin organizasyonu

Bu tez çalışması yedi bölümden oluşmaktadır. Birinci bölümde yazılım proje çizelgelemenin proje yönetimine sağladığı katkılardan bahsedilmiştir. Tez çalışmasının önemi ve kapsamı açıklanmıştır. İkinci bölümde tez konusu ile ilişkili çalışmalar incelenmiştir. Üçüncü bölümde yazılım proje çizelgeleme süreçleri açıklanmış, dördüncü bölümde de çizelgeleme sürecinde kullanılan veri setinin ve yöntemlerin teorilerine yer verilmiştir. Beşinci bölümde geliştirilen yöntemlerin teknik detayları açıklanmış, altıncı bölümde de yöntemlerin test sonuçlarının detaylı analiz ve değerlendirmesi yapılmıştır. Son bölümde tez çalışmasının genel bir değerlendirmesi yapılıp, tezin katkısı belirtilmiştir. Tez çalışmasının organizasyon şemasına Şekil 1.1'de yer verilmiştir.


Şekil 1.1. Çalışmanın organizasyon şeması

# 2. LİTERATÜR ÇALIŞMALARI

Tez çalışması kapsamında proje çizelgeleme süreçleri ile ilişkili literatür çalışmaları incelenmiş olup, bu kapsamda yapılan çalışmalar kronolojik olarak aşağıda özetlenmiştir.

Chang ve arkadaşları çalışmalarında yazılım proje çizelgesini oluşturmada genetik algoritma kullanmışlardır. Görevlerin birbiriyle bağımlılıkları ve insan kaynaklarının uygun görevlere atanması problemini, minimum maliyet, minimum zaman ve insan kaynaklarının belirli bir limit altında çalıştırılması koşullarını gözeterek rastgele üretilen 30 adet proje ve genetik algoritmanın farklı parametreleriyle incelemişlerdir. Genetik algoritmanın popülasyon boyutu, çaprazlama ve mutasyon oranlarının belirli aralıklarda olması durumunda en iyi sonucu elde ettiğini belirtmişlerdir [8].

Chang ve arkadaşları 2001 yılında yaptıkları çalışmalarını [8] gerçek hayattaki problemlere daha elverişli hale getirmek için biraz daha geliştirmişlerdir. Görevlerin gerektiğinde başlatılmadan geciktirilmesi, insan kaynağının tahsis edildiği görevden alınması, belirli sayıda insan kaynağının göreve tahsis edilmesi ile ortaya çıkacak koordinasyon sorunları gibi gerçek hayatta karşılaşılabilen durumları yeni modelde ele almışlardır. Önceki modelde problem kapsamı insan kaynağı ve görev ataması çerçevesinde belirlenmişken, kurdukları yeni modelde insan kaynağının görevi yaparken birim zamanda sarf ettiği çaba da probleme eklenmiştir [9].

Akbari ve arkadaşları çalışmalarında, kaynak kısıtlı proje çizelgeleme problemini, yapay arı kolonisi kullanarak modellemiş ve yapay arı kolonisinin bu problemdeki performansı üzerine çalışmalar gerçekleştirmişlerdir. Çalışmalarında, aktivite sürelerinin ve aktivitelerin gerçekleşmesi için gerekli insan kaynaklarının sabit olduğu tek modlu kaynak kısıtlı proje çizelgeleme problemi kullanmışlar. Yaptıkları test çalışmalarında, yöntemin büyük boyutlu problemlerde diğer algoritmalara kıyasla daha başarılı olduğu sonucunu elde etmişlerdir [10].

Stylianou ve arkadaşları çalışmalarında yazılım projelerinde görevlerin uygun sırada iş akışları için tahsis edilmesi problemini görevlerin birbirine bağlılığı ve ekip içerisinde iletişim yükünü de ele alarak çözmeye çalışmışlardır. Proje çizelgesinin düşük bütçe ile

minimum sürede gerçekleştirilmesi problemini çok faktörlü genetik algoritma yöntemini (MOGA) kullanarak analiz etmişlerdir. MOGA'nın farklı türlerinin (MOCell, NSGA-II, PAES, SPEA 2) problemin uygulanmasında gösterdiği davranışlara ilişkin sonuçları paylaşmışlardır [11].

Nayak ve arkadaşları çalışmalarında yapay arı kolonisi yöntemini, Rastrigin, Rosenbrock, Sphere ve Schwefel foksiyonlarında uygulayarak elde ettikleri sonuçları parçacık sürü optimizasyon (PSO) yöntemi ile karşılaştırmışlardır. Çalışmalarında, yapay arı kolonisi algoritmasının, parçacık sürü optimizasyonu algoritmasına göre daha iyi performans gösterdiğini ortaya koymuşlardır [12].

Xia ve arkadaşları çalışmalarında, yazılım proje çizelgesinin oluşturulma zorluğuna, karınca kolonisi algoritması ile sezgisel bir yaklaşımı birleştirerek oluşturdukları yöntem ile bir çözüm sunmuşlardır. Çalışmalarında, kaynak atamalarını bir grafta modelleyip problemi karınca kolonisi optimizasyon algoritması ile çözülebilecek şekilde graf tabanlı arama algoritması yaklaşımına dönüştürmüşlerdir. Sonrasında geliştirdikleri sezgisel algoritmalarla, görevlerin kaynaklara atanması problemine bir çözüm üretmişlerdir [13].

Chen ve arkadaşları çalışmalarında, olay tabanlı planlama ile karınca kolonisi optimizasyon algoritmasını birleştirerek yazılım proje çizelgesi oluşturma da kullanılmak üzere bir yöntem geliştirmişlerdir. Projenin başlangıcı, görevlerin sonunda kaynakların boşa çıktığı zaman, kaynakların projeye dahil olduğu ya da projeden ayrıldığı zaman, olay olarak değerlendirilip, bu olaylar oluştuğunda kaynakların tahsis edilmesini sağlayacak algoritmalarıyla kaynak çakışması, görevlerin önceliklendirilmesi gibi problemlere çözüm üretmişlerdir [14].

Suri ve arkadaşları çalışmalarında karınca kolonisi optimizasyon algoritmasını kullanarak yazılım proje yöneticilerinin proje maliyetini ve toplam proje süresini en aza indirgeyebilmeleri amacıyla, elitist strateji adını verdikleri bir yöntem geliştirmişlerdir. Yöntemde yeterli sayıda insan kaynağı kullanıldığında yazılım projelerinin maliyet ve tamamlanma sürelerinin azaltılmasının mümkün olduğunu ortaya koymuşlardır [15].

Bansal ve arkadaşları yaptıkları çalışmada, geliştirildiği 2005 yılından itibaren yapay arı kolonisi üzerinde yapılmış çalışmaları derlemişlerdir. Yapay arı kolonisi algoritmasındaki kontrol parametreleri incelenmiş ve benzer optimizasyon algoritmaları ile performans

karşılaştırması yapmışlardır. Algoritmada kullanılan parametrelerin başlangıç koşullarına duyarlı olmaması ve problemin boyutunun artmasının herhangi bir olumsuz etki oluşturmaması nedeniyle diğer optimizasyon algoritmalarına göre artı yönleri bulunan yöntemin uygulama alanlarına dair örnekler sunmuşlardır [16].

Gharehchopogh ve Dizaji yazılım maliyet tahmini gerçekleştirmek için kullandıkları yöntemde, arı kolonisi ve kaotik optimizasyon algoritmalarından oluşan hibrit bir model kullanmışlardır. Arı kolonisi optimizasyonu başlangıçtaki çözümleri bulmak için kullanılırken, kaotik algoritmalar bulunan bu çözümleri iyileştirmek için kullanılmıştır [17].

Luna ve arkadaşları çalışmalarında, yazılım proje çizelgesi oluşturma probleminde sekiz adet çok amaçlı algoritmayı (NSGA-II, PAES, SPEA2, DEPT, MOCell, MOABC, MO-FA, GDE3) analiz etmişler ve PAES algoritmasının farklı ölçekler (insan kaynağı sayısı, aktivite sayısı) göz önüne alındığında en iyi sonucu ürettiği sonucunu ortaya koymuşlardır [18].

Mirjalili ve arkadaşları, bozkurtların yönetim yapısını ve doğadaki yiyecek arayışlarını modelleyerek yeni bir sezgisel optimizasyon yöntemi geliştirmiş ve bilinen sezgisel algoritmalar ile karşılaştırarak test etmişlerdir. Bu çalışma ile literatüre bozkurt optimizasyon algoritmasını kazandırmışlardır [19].

Pauzi yapay arı kolonisi yaklaşımını kullanarak iş akışı çizelgelendirme problemine çözüm aramış ve yapay arı kolonisi için gözcü arı yaklaşımını değiştirerek bir öneri sunmuştur. Tez çalışmasında örnek verinin rastgele üretilmesi sonucu altı işin, üç makinede minimum sürede çözülebilmesi için çalışmalar gerçekleştirilmiştir [20].

Seo ve arkadaşları genetik algoritma kullanarak yazılım proje çizelgesi oluşturma probleminde maliyeti bir limit altında minimize edip, kaliteyi yüksek tutacak bir yöntem üzerine çalışmışlardır. Kalite değerlendirmesini görev, faz ve proje kapsamında ayrı ayrı gerçekleştirmişlerdir. Bir görevde aynı anda çok sayıda insanın görevlendirilmesi, bir kişinin benzer işlerde görevlendirilmesi, ekibe profesyonel birinin dahil edilmesi gibi durumları görev kalitesi; görevlerin hataya duyarlılığı faz kalitesi fazlar arasında hataya duyarlılık değerlendirmesi de proje kalitesi olarak ele alınmıştır. Kalite ve maliyet parametrelerini içeren uygunluk fonksiyonları ile uygun çözümü elde etmeye çalışmışlardır. Geliştirilen

yöntemin, maliyet tabanlı yöntem dikkate alındığında proje hatalarını düşük seviyede tutabilecek bir yöntem olduğunu ortaya koymuşlardır [21].

Myszkowski ve arkadaşlarının yaptığı çalışmada, proje çizelgesi oluşturma problemine, gerçek hayattaki problemlerle örtüşmesi adına yetenek (skill) yaklaşımı eklenmiştir. Çoklu yetenekli proje çizelgesi oluşturma adı verilen yöntem ile problemi çözme karmaşıklığı; görevlerin birbiri ile bağlantısı, kaynakların iş yükü, görevlerin gerçekleşme süresi, maliyet, insan kaynaklarının sayısı, insan kaynakların yetkinlik seviyesi ve insan kaynaklarının görevlere atanabilme yeterliği kriterlerine göre belirlenmiştir [22].

Myszkowski ve arkadaşları, diferansiyel gelişim algoritmasına her döngü sonunda görevler için insan kaynağı atamasını iyileştirecek bir fonksiyon ekleyerek; yakınsama hızını arttıran, yerel minimuma takılmayı önleyen bir yöntem geliştirmiştir [23].

Mirjalili ve arkadaşlarının yaptığı çalışmada, bozkurt algoritmasına arşiv özelliği eklenerek, optimizasyon sürecinde baskın olmayan verilerin belirlenmesi ve alfa, beta ve delta kurtlarının arşivdeki kayıtlardan seçilmesini sağlayacak bir yaklaşım sunulmuştur [24].

Pauzi ve arkadaşı çalışmalarında, yapay arı kolonisi algoritmasında gözcü arı yaklaşımını değiştirerek, üç farklı gözlemci arı yaklaşımı eklemişlerdir. En iyi çalışan arıya üç gözlemci arının atanması (3+0+0 yöntemi), en iyi çalışan arıya iki gözlemci ve ikinci en iyi çalışan arıya bir gözlemci arının atanması (2+1+0 yöntemi), (3) her bir çalışan arıya bir gözlemci arının atanması (1+1+1 yöntemi). Yapılan çalışmada bu yöntemler analiz edilmiş ve performansları değerlendirilmiştir [25].

Mathew ve arkadaşları çalışmalarında, tekrarlayan projelerde (projenin benzer ya da aynı alt birimlerden oluşması-örn: site inşaatı) proje ekiplerinin projedeki alt birimlerde görevlendirilmesi problemini genetik algoritma kullanarak ele almışlardır. Alt birimler için proje ekibi atamasında aktivite bağımlılıkları ve eş zamanlı gerçekleştirilen işlerin gecikme oluşturmaması durumlarını da dikkate alarak, projedeki süre ve maliyet değerlerini optimize etmeye çalışmışlardır [26].

Long ve arkadaşı bozkurt optimizasyon algoritmasındaki başlangıç popülasyonunu, durum uzayında iyi dağıtılmış noktaları seçerek optimizasyon algoritmasının daha etkin çalışmasını sağlamayı amaçlayan good-point-set yöntemini kullanarak oluşturup, çözüm kalitesi ve yakınsama performansını arttırmaya yönelik çalışmalar gerçekleştirmişlerdir [27].

Hameed ve arkadaşlarının yaptığı çalışmada, sürü yaklaşımı kullanılarak proje aktivitelerinin daha etkin çizelgelendirilmesi ve görevlerin proje üyelerine tecrübe ve yeteneklerine uyumu olacak şekilde atanması amacıyla karınca kolonisi algoritması, evrimsel algoritma ve genetik algoritma yöntemleri uygulanmıştır [28].

Gaidhane ve arkadaşlarının yaptığı çalışmada, bozkurtların arama stratejisinde yapay arıların bilgi paylaşım yöntemi kullanılarak algoritmanın ilerleyen süreçlerinde popülasyonun daha iyi sonuç üreten üyelerden oluşturulması sağlanmıştır [29].

Gai ve arkadaşlarının yaptığı çalışmada bozkurt optimizasyon algoritmasında kontrol parametresinin kosinüs fonksiyonundan türetilmesi ve genetik algoritmadaki mutasyon ve çaprazlama süreçlerinin bozkurt optimizasyon algoritmasına entegre edilmesi sağlanmıştır. Böylelikle algoritmanın ilerleyen döngülerdeki geç yakınsama, yerel minimuma yaklaşma ve yakınsamadaki gecikme sorunlarına çözüm üretilmiştir [30].

Proje çizelgesi oluşturma probleminin çözümü için Myszkowski ve arkadaşlarının geliştirdiği yöntemde, diferansiyel gelişim algoritması görevlere insan kaynaklarının atanmasında, aç gözlü algoritma ise görevlerin öncellerine göre işleme konma sıralamasını belirlemede kullanılacak şekilde hibrit bir yöntem sunulmuştur [31].

Bibiks ve arkadaşlarının yaptığı çalışma ile guguk kuşu arama algoritması, kaynak kısıtlamalı proje çizelgesi oluşturma problemi gibi ayrık zamanlı problemlerin çözümüne elverişli hale getirilmiştir. Yaklaşımda, aynı zamanda başlayabilecek görevler, olay(event) listesinde olay kaydı olarak tutulur. Böylelikle çizelge, olayların başlama zamanına göre tasarlanır. Olay listeleri içerisinde yer alan görevler, birbirinden bağımsız konumlara hareket ettirilerek olaylar arasında geçişler sağlanır [32].

Afshar'ın yaptığı çalışmada, proje maliyetini azaltmak için projedeki görevlerin farklı yürütme modlarında başlatılmasının sağlandığı ve görevlerin daha sonra kesintiye uğrayıp yeniden başlatıldığı senaryolarda projedeki maliyet artışının en aza indirilmesini hedefleyen benzetimli tavlama algoritması önerilmiştir [33].

Nemati-Lafmejani ve arkadaşlarının yaptığı çalışmada, maliyet ve proje süresini en aza indirmek için çok modlu kaynak kısıtlı proje çizelgesi oluşturma problemine yüklenici seçimi sorunu dahil edilmiş ve projedeki yüklenici sayısının artırılmasının proje süresine olumlu katkı sağladığı tespit edilmiştir [34].

Guo ve arkadaşları bozkurt optimizasyon algoritmasının yerel minimuma takılma problemini çözmek için kedi sürü optimizasyonunda bulunan arama ve takip etme yaklaşımı probleme eklemiştir [35].

Lu ve arkadaşlarının çalışmalarında bozkurt optimizasyon algoritmasındaki arama ve sömürme işlemlerini gerçekleştiren parametreler kaotik haritalarla belirlenerek sonuçlar genetik algoritma, yapay arı kolonisi algoritması, biyocoğrafya tabanlı optimizasyon vb. yöntemlerle karşılaştırılmıştır [36].

Proje çizelgesi oluşturma probleminin çözümü için Tian ve arkadaşlarının uyguladıkları yöntemde, kaynakların görevlere dengeli bir şekilde atanması için kaynak seviyelendirme, görevlerin başlaması için ortaya çıkan boşta bekleme süresinin en aza indirilmesi için de çizelge sıkıştırma işlemleri gerçekleştirilmiştir. Sonuçlar genetik algoritma ve meyve sineği algoritması ile karşılaştırılmıştır [37].

Rauf ve arkadaşları, birden çok projenin ilişkili proje planı probleminin çözümü için rakun optimizasyon yöntemini kullanmışlardır. Farklı teslim tarihine sahip projelerin çizelgeleme problemi, aç gözlü arama ve genetik algoritmadaki mutasyon ve çaprazlama işlemleri ile desteklenen rakun optimizasyon problemi ile çözülmüştür [38].

Quoc ve arkadaşları, görevlerin gerçekleşme süresinin yetkinlik seviyesi ile ters orantılı olduğu yaklaşımı öne sürerek kaynak kısıtlamalı proje çizelgesi oluşturma problemini guguk kuşu arama metodu ile çözmüştür [39].

Xie ve arkadaşları, görev maliyetinin belirgin olmadığı durumlarda proje çizelgesi oluşturma problemini çözmek amacıyla belirlenen proje bütçesinin aşılmasını önlemek için lineer olmayan tam sayı yöntemini, popülasyon kalitesini arttırmak için de genetik algoritma ve sezgisel yöntemleri kullanmıştır [40].

Snauwaert ve arkadaşı, kaynakların sahip olduğu yetkinlik sayısını genişlik, bu yetkinliklerdeki seviyeleri de derinlik olarak ele alarak kaynak kısıtlı proje çizelgeleme problemine bir çözüm önermişlerdir. Yaptıkları çalışmada, görevlere yetkin kaynakların atanmasının proje süresini azaltmaya olan etkisi analiz edilmiştir [41].

Ghamginzadeh ve arkadaşları çalışmalarında, görev sürelerinin belirsiz olduğu durumlarda proje yöneticilerinin proje planlaması ve bütçe tahsisi yapabilmesi için bir model önermişlerdir. Modelde, görevlerin belirsiz olan tamamlanma süreleri bulanık mantık ile modellenmekte, optimizasyon süreçlerinde ise Taguchi yöntemi kullanılmaktadır [42].

Cai ve arkadaşları, çoklu yetenek kaynak kısıtlı proje çizelgeleme problemini çözmek için sezgisel bir algoritma önermişlerdir. Algoritmaya göre, görevlere insan kaynaklarının atanması sonrasında, eğer boşta kalan insan kaynağı varsa, görev tamamlanma süresini azaltmak için bu insan kaynakları için görev ataması gerçekleştirilmektedir [43].

Wang ve arkadaşları çoklu yetenek kaynak kısıtlı proje çizelgeleme problemindeki çok yetenekli kaynakların yeteneklerinin proje süresince değişme olasılığını dikkate alan bir yöntem geliştirmişlerdir. Geliştirilen yöntem, projenin tamamlama zamanını optimize ederken, kaynakların yeteneklerinde değişiklik olup olmadığını kontrol ederek olası kaynak değişikliklerinde proje yöneticilerine alternatif sağlamaktadır [44].

Mahmud ve arkadaşları, evrimsel algoritmaların kaynak kısıtlı proje çizelgeleme problemini çözerken ortaya çıkan parametre ayarlama zorluğuna alternatif bir çözüm getirerek, popülasyonu çeşitlendirmede mutasyon ve çaprazlama işlemlerini birlikte kullanmışlardır. Çalışmada, yerel arama tekniği ile görevlerin en erken başlama ve en geç bitme durumlarına göre sıralanması yaklaşımı ile proje çizelgesi optimize edilmiştir [45].

Akbar ve arkadaşları, geleneksel çoklu yetenek kaynak kısıtlı proje çizelgeleme problemine, insan kaynaklarının kişisel yeteneklerini de dahil ederek, insan kaynaklarının görevleri hangi

verimlilikte gerçekleştirebileceklerini analiz eden bir yöntem geliştirmişlerdir. Eş zamanlı başlayabilecek görevleri bekletmeden ve boşta insan kaynağı varsa uygun göreve ek insan kaynağı ataması yaparak proje süresini optimize etmişlerdir [46].

Ou ve arkadaşları, yazılım proje çizelgeleme problemini kuantumdan esinlenerek ikinci dereceden kısıtlanmamış ikili optimizasyon (Quadratic Unconstrained Binary Optimization) modeline dönüştürmüşlerdir. Modelde, görevler ve insan kaynakları ikili değişkenlerle (0,1) tanımlanmış, klasik tavlama yönteminin dijital versiyonu oluşturularak optimizasyon gerçekleştirilmiştir [47].

Nigar ve arkadaşları, yazılım projelerinde yeni insan kaynaklarının işe alınması durumunda projelerde yaşanabilecek aksaklıkların önlenmesi amacıyla, genetik algoritma ve meta-sezgisel yöntemler kullanarak proje süresi ve maliyetini optimize ettikleri bir yöntem önermişlerdir [48].

Nigar ve arkadaşları geliştirdikleri bir diğer yöntemde bu kez de projede çalışan insan kaynaklarının görevden ayrılması veya değişmesi durumunda yaşanabilecek aksaklıkları minimize etmeye çalışmışlardır. Uygun yetkinlik seviyesine göre yeni insan kaynağı atamalarını yaparak proje üzerindeki olumsuz etkileri azaltacak bir optimizasyon yöntemi önermişlerdir [49].

Akbar ve arkadaşları, ağırlıklı çoklu yetenek kaynak kısıtlı proje çizelgeleme probleminde değişen görev süreleri nedeniyle insan kaynaklarının yetersiz kullanımının ortaya çıkardığı kaynak israfını önlemeyi amaçlamışlardır. Geliştirdikleri yöntemde, görevleri öncelik sırasına göre düzenleyip, kaynakları en kısa sürede tamamlanacak görevlere atayarak verimli kaynak kullanımı sağlamışlardır [50].

Literatürde proje çizelgeleme üzerine geniş çaplı çalışmalar gerçekleştirilmiş olmakla beraber, aşağıda belirtilen konularda yapılacak çalışmaların arttırılması proje başarı oranlarına olumlu katkılar sağlayabilecektir:

- Proje başladıktan sonra ortaya çıkabilecek değişikliklerde, projenin başlangıcında ortaya konan hedeflerden sapma oranını düşük seviyede tutulmalıdır. Bu alanda yapılacak optimizasyon çalışmaları ile proje çizelgeleme süreçleri dinamik olarak beslenmelidir.

- Projede görevlendirilecek insan kaynaklarının görev bağımlılıkları dolayısıyla yaşayacakları olası bekleme sürelerinin projeye maliyetinin analizi yapılmalıdır.
- İnsan kaynaklarının yetkinlik seviyelerinin görev sürelerine etkilerini analiz eden optimizasyon çalışmaları gerçekleştirilmelidir.
- Gerçek hayattaki yazılım projelerinin varsayım ve kısıtları farklı olabilmektedir. Bu kısıtlara göre optimizasyon yöntemlerinin davranışları da farklılaşabilmektedir. Bu nedenle yazılım mühendisliği alanında saha çalışmalarından elde edilen veri setlerine ihtiyaç duyulmaktadır.

# 3. YAZILIM PROJELERİNDE ÇİZELGELEME SÜRECİ

Yazılım projelerinin yönetiminde proje çizelgeleme, projenin başarıyla tamamlanmasının temel taşlarından biridir. Bu süreç projede zaman, kaynak dağılımı ve görevlerin sıralamasını kapsamlı bir şekilde planlamayı içerir. Yazılım proje çizelgesinin doğru bir şekilde oluşturulması, zaman ve maliyetin etkin yönetilmesine imkân sağlar ve proje hedeflerine ulaşılmasında kritik bir rol oynar [51].

Yazılım proje çizelgeleme sürecinin ilk adımı, proje kapsamının net bir şekilde tanımlanmasıdır. Proje kapsamı, projenin hedeflerini, beklenen çıktılarını, kısıtlamalarını ve varsayımlarını içerir. Kapsamın doğru bir şekilde belirlenmesi, projenin gerçekçi ve ulaşılabilir hedeflere sahip olmasını sağlar ve çizelgeyi oluştururken yol gösterici bir rol oynar.

Kapsam belirlendikten sonra, kapsam dokümanında yer alan bilgiler esas alınarak projenin başarıyla tamamlanması için gerekli olan tüm görevlerin detaylı bir şekilde tanımlanması gerekmektedir. Bu tanımlama proje detaylarının tüm paydaşlar tarafından anlaşılabilmesine yardımcı olur. Bu süreçte görevlerin tamamlanması için gerekli süre tahminleri ve görevlerin birbiriyle bağlantıları dikkate alınarak kaynak planlaması gerçekleştirilir. Kaynak planlaması, proje ekibinin, projenin gerekli aşamalarında ihtiyaç duyulacak bütçenin, projede kullanılacak malzemelerin ve hatta projenin hazırlanacağı mekânın belirlenmesi gibi süreçleri kapsar.

Tüm bu bilgiler doğrultusunda, yazılım proje çizelgesi oluşturulur. Bu çizelge, projenin tüm görevlerini, zaman çizelgesini ve kaynak dağılımını içerir. Proje çizelgesi genellikle Gantt şeması gibi araçlar kullanılarak görselleştirilir. Proje ilerledikçe, çizelgenin düzenli olarak izlenmesi ve gerekli durumlarda güncellenmesi önem taşır. Bu sayede projenin beklenmedik değişikliklere uyumlu hale gelmesi sağlanır [52].

## 3.1. Kaynak Kısıtlı Proje Çizelgeleme Problemi (KKPÇP)

Kaynak kısıtlı proje çizelgeleme problemi (KKPÇP), projelerin yönetimi ve verimli bir şekilde planlanması açısından kritik öneme sahip olan bir problemdir. Kısıtlamaların ve sınırlı sayıda insan kaynağının bulunduğu bir projede, başlangıç koşulları öncel durumlarına

bağlı olan bir dizi görevin planlanması problemidir. Bu problemde genellikle iki temel hedef bulunmaktadır:

- Projenin süresini makul olan en düşük seviyeye çekmek,
- Bütçeyi en aza indirmek.

Projenin toplam süresini en aza indirme: Projelerin yönetiminde temel hedeflerden biri, projenin başlangıç tarihinden itibaren olabilecek en kısa sürede tamamlanmasını sağlamaktır. Bu amaç doğrultusunda, proje yöneticisi, projeyi planlarken görevlerin sürelerini ve öncel durumlarını dikkate alarak projenin kritik yolunu belirler. Kritik yol, projenin tamamlanması için en uzun süreyi gerektiren etkinlikler dizisidir ve projenin tamamlanma süresini belirler. Bu etkinliklerden herhangi birinde yaşanacak gecikme, tüm projenin gecikmesine neden olabilir. Bu nedenle, bu yol üzerindeki etkinliklerin yönetimi, projenin zamanında tamamlanması açısından önem taşır [53].

Projenin toplam bütçesini en aza indirme: Proje yöneticisinin bir diğer amacı, projenin maliyetlerini en aza indirgemektir. Projenin her etkinliğine atanan kaynaklar ve bu kaynakların maliyetleri göz önünde bulundurularak, projenin toplam maliyeti en aza indirgenecek şekilde kaynaklar tahsis edilmeye çalışılır.

Proje yöneticisi, projenin süresini en aza indirgemek için ek kaynaklar kullanabilir, ancak bu değişiklik projenin toplam maliyetini artırabilir. Benzer şekilde, projenin maliyetini en aza indirgemek için kaynak kullanımını kısıtlamak, projenin süresini uzatabilir. Projenin başlangıcından bitişine kadar gerçekleştirilmesi beklenen tüm faaliyet ve görevler için öngörülen toplam finansal kaynak olan proje bütçesi ile projenin gerçekleştirilmesi sırasında gerçekten harcanan miktar olan proje maliyeti arasında denge kurmak gerekmektedir [51,54]. Bu nedenle, kaynak kısıtlı proje planlaması probleminde, proje yöneticileri bu iki hedef arasındaki en uygun dengeyi bulmak için uygun yöntemler ve stratejiler kullanmaya çalışır.

Her görevin belirli bir süresi vardır ve görevleri belirli yetkinlikteki insan kaynakları gerçekleştirebilmektedir [55]. Görevlerin, öncelik durumlarının ve insan kaynaklarının sayısındaki artış, problemin çözümünü giderek zorlaştırır. Bu özellikleri nedeniyle kaynak kısıtlamalı yazılım proje çizelgeleme problemi NP-zor problem sınıfına girer ve bu özelliği

onu standart matematiksel yöntemlerle çözüme ulaşılamaz yapar [56-57]. Problemin çözümünde sezgisel veya meta-sezgisel algoritmalar kullanılarak, çok büyük olan durum uzayında en iyi çözüme ulaşılabilmesi amaçlanır [58-59]. Geliştirilen yöntemlerin hiçbiri en iyi sonucun elde edileceği garantisini vermez.

### 3.2. Çoklu Yetenek Kaynak Kısıtlı Proje Çizelgeleme Problemi (ÇY-KKPÇP)

KKPÇP, projelerdeki görevlerin zamanında ve verimli bir şekilde tamamlanabilmesi için insan kaynaklarının optimal kullanımını hedefler. Bu, projenin tamamlanma süresini kısaltmayı, maliyetleri düşürmeyi ve insan kaynağı kullanımını en üst düzeye çıkarmayı amaçlar. Ancak bu yaklaşımda proje ekibinin bireysel yetenekleri ve görevlerin başarı ile tamamlanması için gereken yetkinlik düzeyleri göz ardı edilmektedir.

Yazılım projelerinin karmaşıklığının artması ve projelerdeki farklı yetkinliklere sahip insan kaynaklarının verimli kullanılmasının önem kazanması ile daha gerçekçi proje çizelgeleme süreçlerine ihtiyaç duyulmuştur. ÇY-KKPÇP, KKPÇP'nin gelişmiş bir versiyonu olup, proje çizelgeleme sürecini bir adım ileri taşır. Kaynak planlamasına yetkinlik seviyesi boyutunu ekleyerek, projelerin yönetimini ve planlamasını daha gerçekçi şekilde ele alır. Bu problem, projelerdeki kaynak atama ve yetkinlik seviyeleri ile ilgili zorlukları daha ayrıntılı bir şekilde modeller. KKPÇP'deki kaynakların her faaliyeti aynı yetkinlik düzeyinde gerçekleştireceği genellemesinin aksine, ÇY-KKPÇP'de bazı kaynakların belirli faaliyetlerde diğerlerinden daha yetenekli olduğu varsayımı bulunmaktadır. ÇY-KKPÇP, kaynakların yetkinlik seviyelerini dikkate alarak, her bir faaliyete en uygun kaynağın atanmasını sağlar. Bu, projelerin daha etkin yönetilmesine ve planlanmasına imkân tanırken, kaynak kullanımının optimizasyonunu da sağlar [60].

ÇY-KKPÇP proje çizelgelemesinde karşılaşılan zorlukları ele alırken aynı zamanda hesaplama karmaşıklığını da artırır. KKPÇP'nin NP-zor problem yapısında olması, ÇY-KKPÇP'yi de NP-zor sınıfına sokar. O nedenle ÇY-KKPÇP'yi çözmek için de geleneksel hesaplama yöntemleri yeterli gelmez. NP-zor problemlerde geleneksel yöntemlerin kullanımı çok fazla matematiksel işlem gerektireceğinden, ÇY-KKPÇP'nin çözümü için sezgisel ve meta-sezgisel yöntemlere ihtiyaç duyulur. Bu tür yöntemler, durum uzayında etkin arama yapılabilmesine ve olası en iyi sonuca ulaşılabilmesine yardımcı olur [61].

### 3.2.1. Problem tanımı ve formülasyonu

ÇY-KKPÇP, insan kaynaklarının projedeki görevlere, bu kaynakların farklı yetkinlik seviyelerini dikkate alarak en uygun şekilde atanmasını optimize eder. Bu yaklaşım, her görevin en uygun yeteneklerle yerine getirilmesini sağlayarak, projenin genel performansını ve verimliliğini artırır.

Görev öncel ilişki tablosu, bir projede belirli bir görev başlamadan önce tamamlanması gereken diğer görevlerin bilgisini içerir. Bu bilgiler, projenin zamanında ve etkin bir şekilde tamamlanması için proje yöneticisinin yapacağı planlamada, görevlerin hangi sırayla gerçekleştirilmesi gerektiği konusunda yol göstericidir.

Çizelge 3.1'de örnek bir proje için görev öncel ilişkisi yer almaktadır. Bu örnek projede görülebileceği üzere, projelerde bazı görevler hiçbir ön koşul olmadan başlayabileceği gibi ($G_1$ ve $G_2$), bazı görevler bir veya daha fazla görevin tamamlanması koşulu ile başlayabilmektedir ($G_3$, $G_6$ gibi).

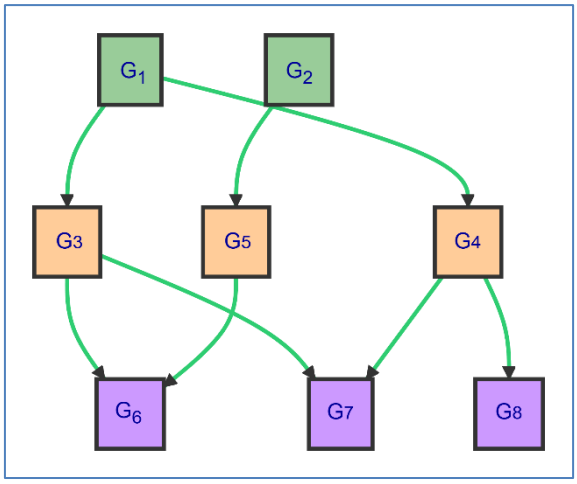
Çizelge 3.1. Görev öncel ilişkisi

| Görev | Öncel |
|---|---|
| $G_1$ | - |
| $G_2$ | - |
| $G_3$ | $G_1$ |
| $G_4$ | $G_1$ |
| $G_5$ | $G_2$ |
| $G_6$ | $G_3$, $G_5$ |
| $G_7$ | $G_3$, $G_4$ |
| $G_8$ | $G_4$ |

Görevlerin, öncel ilişkileri ve bağımlılıklarının gösterimi çizelge ile yapılabileceği gibi, graf ile de gerçekleştirilebilir. Graf modeli görevlerin öncel ilişkileri ve bağımlılıkları ile ilgili

görsel bir model sunarak, proje yöneticilerinin planlama ve yönetim süreçlerini gerçekleştirmede daha etkin kararlar almasına yardımcı olur.

Şekil 3.1'de, tablo gösterimi verilen örnek bir proje için görevlerin öncel ilişkilerinin ve bağımlılıklarının graf şeklindeki gösterimi yer almaktadır.




Şekil 3.1. Görev öncel ilişkisi graf gösterimi

ÇY-KKPÇP'de projenin tamamlanma süresinin belirlenebilmesi için projede yer alan görevlerin her birinin tamamlanma sürelerinin de bilinmesi gerekmektedir. Çizelge 3.2'de örnek bir proje için görev tamamlanma süreleri belirtilmektedir.

Çizelge 3.2. Görev tamamlanma süreleri

| Görev | Süre (Saat) |
|---|---|
| $G_1$ | 3 |
| $G_2$ | 4 |
| $G_3$ | 2 |
| $G_4$ | 5 |
| $G_5$ | 6 |
| $G_6$ | 2 |
| $G_7$ | 3 |
| $G_8$ | 4 |

Örnek projede görevlendirilen beş adet insan kaynağının yetkinlik seviyeleri Çizelge 3.3'te, görevlerin gerçekleştirilmesi için gerekli olan asgari insan kaynağı yetkinlik seviyeleri de Çizelge 3.4'te belirtilmektedir.

Çizelge 3.3. İnsan kaynağı yetkinlik seviyeleri

| İnsan Kaynağı | Yetkinlik Seviyesi$_1$ | Yetkinlik Seviyesi$_2$ | Yetkinlik Seviyesi$_3$ |
|---|---|---|---|
| İK$_1$ | 2 | 0 | 1 |
| İK$_2$ | 1 | 1 | 1 |
| İK$_3$ | 0 | 2 | 0 |
| İK$_4$ | 1 | 0 | 2 |
| İK$_5$ | 1 | 1 | 0 |

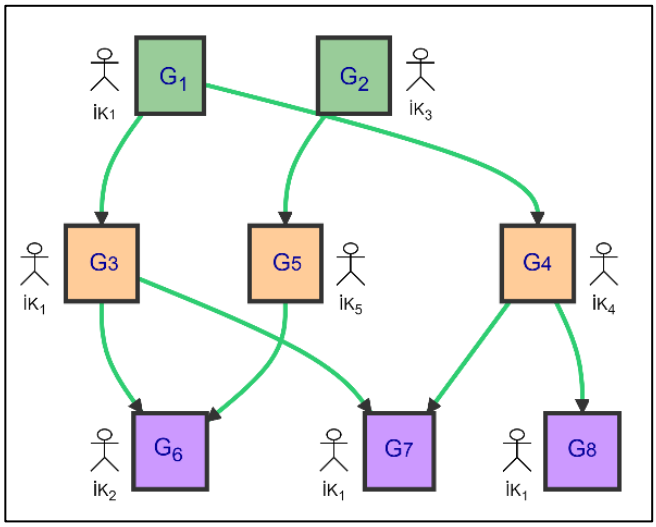
Çizelge 3.4. Görevler için yetkinlik seviyeleri

| Görev | Yetkinlik Seviyesi$_1$ | Yetkinlik Seviyesi$_2$ | Yetkinlik Seviyesi$_3$ |
|---|---|---|---|
| G$_1$ | 1 | 0 | 0 |
| G$_2$ | 0 | 2 | 0 |
| G$_3$ | 2 | 1 | 0 |
| G$_4$ | 0 | 0 | 1 |
| G$_5$ | 1 | 1 | 0 |
| G$_6$ | 0 | 1 | 1 |
| G$_7$ | 1 | 0 | 1 |
| G$_8$ | 2 | 0 | 0 |

Örnek projede, 8 adet görev ve 5 adet insan kaynağı mevcuttur. Görevlerin öncel durumları ve insan kaynaklarının yetkinlikleri dikkate alınarak manuel olarak oluşturulan bir proje çizelgesi için kaynak atamaları Şekil 3.2'de gösterilmektedir.

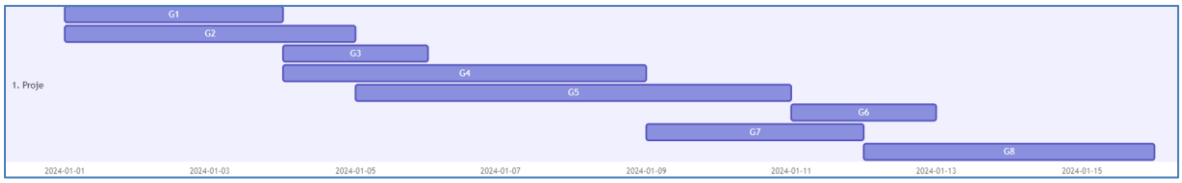
Proje çizelgesinin manuel oluşturulmasına ilişkin adımlar aşağıda belirtilmiştir:

- Başlangıç görevleri olan $G_1$ ve $G_2$, proje başladığında hemen başlamaktadır. $G_1$, $İK_1$ kaynağı tarafından gerçekleştirilirken, $G_2$, $İK_3$ kaynağı tarafından gerçekleştirilmektedir.
- $G_1$'in ardından $G_3$ ve $G_4$ görevleri başlamaktadır. $G_3$, $İK_1$ kaynağı tarafından, $G_4$ ise $İK_4$ kaynağı tarafından gerçekleştirilmektedir.
- $G_2$'nin tamamlanmasıyla $G_5$ görevi başlamakta ve bu görev $İK_5$ kaynağı tarafından gerçekleştirilmektedir.
- $G_3$ ve $G_5$ görevlerinin tamamlanmasının ardından, $G_6$ görevi başlamakta ve görev $İK_2$ kaynağı tarafından gerçekleştirilmektedir.
- $G_3$ ve $G_4$ görevlerinin tamamlanmasıyla $G_7$ görevi başlamakta ve görev $İK_1$ kaynağı tarafından gerçekleştirilmektedir.
- $G_4$ görevinin tamamlanmasının ardından $G_8$ görevi başlamakta ve görev $İK_1$ kaynağı tarafından gerçekleştirilmektedir.



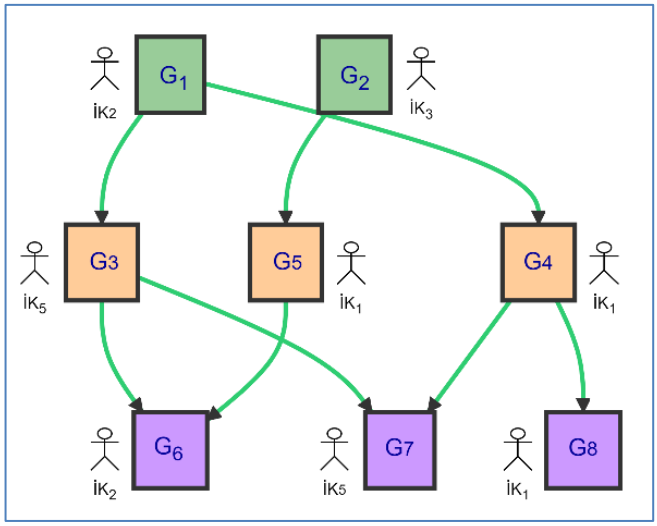

Şekil 3.2. Proje 1 için kaynak ataması

Manuel olarak oluşturulan örnek proje çizelgesi için, proje yönetiminde sıklıkla kullanılan ve projelerin zaman çizelgesini görselleştirmeye yardımcı olan Gantt şeması Şekil 3.3'te gösterilmektedir. Şema, projenin toplam süresinin 15 saat olduğunu göstermektedir. Bu süre içinde, kaynakların etkin kullanılması ve görevler arasındaki öncel ilişkilerinin dikkate alınması sağlanmıştır. Ancak, projenin Gantt şeması üzerinden elde edilen 15 saat, manuel olarak yapılan kaynak atamaları ve görev öncelikleri ile elde edilmiş olup, optimal bir sonuç olmayabilir.

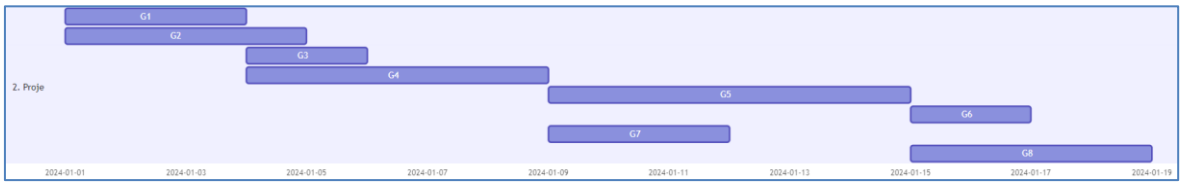



Şekil 3.3. Proje 1 için Gantt şeması

Aynı örnek proje için, farklı bir proje çizelgesine ait çıktılar Şekil 3.4'te ve Şekil 3.5'te yer almaktadır.




Şekil 3.4. Proje 2 için kaynak ataması




Şekil 3.5. Proje 2 için Gantt şeması

Örnek olarak verilen iki adet proje çizelgesinde proje yöneticisi, görevlerin öncel durumlarını ve insan kaynakları yetkinliklerini dikkate alarak, insan kaynağı ataması için iki farklı senaryo ortaya koymuş, bu atamalar sonucunda ortaya çıkan çizelgelerinden biri projenin 15 saatlik sürede, diğeri ise 18 saatlik sürede tamamlanabileceğini göstermiştir. Bu durum, proje çizelgesi oluştururken yapılacak kaynak atamalarının projenin süresi ve verimliliği üzerinde kritik öneme sahip olduğunu ortaya koymakta olup, manuel yöntemlerin proje çizelgesi oluşturmada sınırlı verimlilik sağladığını göstermektedir.

Projelerin daha büyük ve karmaşık olduğu durumlarda, manuel olarak proje çizelgesi oluşturma süreçleri yetersiz kalmakta ve zaman alıcı olmaktadır. Bu nedenle, proje yöneticilerinin kaynak atamalarını ve zaman çizelgelerini uygun şekilde planlaması için bilimsel yöntemlerin kullanılması ihtiyacı ortaya çıkmaktadır. Yapay zekâ ve optimizasyon teknikleri, projelerin çizelgelenmesi ve kaynakların etkin bir şekilde yönetilmesine katkı sağlayarak, süreçlerin daha hızlı ve doğru bir şekilde gerçekleştirilmesini mümkün kılar. Bu tür teknikler, özellikle karmaşık proje yapıları ve kısıtlı kaynaklarla çalışılan projelerde, daha iyi çözümler üretebilmekte ve karar verme süreçlerini iyileştirebilmektedir. Bu sayede, proje yöneticileri, en uygun zaman ve kaynak atamalarını elde edebilmekte ve projelerin başarılı bir şekilde tamamlanmasını sağlayabilmektedirler [62-63].

### 3.2.2. Problemin durum uzayı boyutu

ÇY-KKPÇP, NP-zor optimizasyon problemi olarak tanımlanır. p adet görev ve q adet insan kaynağının olduğu bir projede hesaplanan durum uzayı boyutu (SS) Eş. 3.1'de belirtilmektedir.

$$SS = p! \, x \, q^p \tag{3.1}$$

Bu hesaplama projedeki görev ve insan kaynağının oluşturacağı tüm çizelgeleri kapsar, bir başka değişle hesaplanan durum uzayında gerçekleştirilemeyecek çizelgeler de mevcuttur [56,64-65].

### 3.2.3. Kısıtlar ve varsayımlar

ÇY-KKPÇP'de öne çıkan varsayımlar ve kısıtlar aşağıda belirtilmektedir:

<u>Varsayımlar</u>

- ÇY-KKPÇP'de görev sürelerinin kesin ve değişmez olduğu varsayılır [54]. Bu durum, gerçek hayatta sıklıkla karşılaşılan beklenmedik gecikmeleri göz ardı eder. Projelerdeki olası gecikmelerde de optimizasyon süreçlerinin tekrarlanması ve proje planının yeniden oluşturulması zorunluluğunu ortaya çıkarır.

- Probleme ilişkin diğer bir varsayım görevlerin kesintiye uğramayacağı ve başladıktan sonra sürekli olarak yürütüleceğidir. Başka bir deyişle, bir görev başladığında, başka bir görev ile değiştirilemez ve görev tamamlanana kadar devam eder [66]. Bu varsayım, görevlerin zamanlamasını ve kaynak kullanımını basitleştirir, ancak gerçek hayattaki projelerde bazı durumlarda görevlerde kesintilere izin vermek gerekebilmektedir.
- ÇY-KKPÇP'de kaynakların sınırlı olduğu kabul edilmektedir [67]. Ancak proje başlangıcında belirlenen proje ekibi, projenin ilerleyen dönemlerinde karşılaşılabilecek çeşitli durumlar nedeniyle değişiklik gösterebilmektedir. Özellikle bir ekip üyesinin beklenen performansı gösterememesi, kişisel nedenlerle işten ayrılması veya diğer zorunlu durumlar, projeden çıkarılmasını gerektirebilir. Bu tür değişiklikler, proje yöneticisinin kaynakları yeniden düzenlemesini ve proje hedeflerine ulaşmak için gerekli uyumları yapmasını zorunlu kılar. Bu durumda optimizasyon süreçlerinin tekrarlanması ve proje planının yeniden oluşturulması zorunluluğunu ortaya çıkar [68].
- ÇY-KKPÇP'de insan kaynaklarının belirli yetkinliklere ve bu yetkinliklerin farklı seviyelere sahip olduğu kabul edilir. Görevlerin başarıyla tamamlanması için gerekli yetkinlik seviyesinin kaynak atamasında önemli bir faktör olduğu varsayılır [69]. Yetkinlik seviyeleri dikkate alınarak yapılan kaynak ataması, projenin başarısı için kritik olsa da bu durum modelin karmaşıklığına sebep olmaktadır. Ayrıca, yetkinlik seviyesinin belirlenme süreci de her zaman kolay olmamaktadır.

<u>Kısıtlar</u>

- ÇY-KKPÇP'de, görevlerin başlama sırasındaki en önemli faktör öncel kısıtlamalarıdır. Öncel kısıtlamaları, projedeki görevlerin hangi sırada ve ne zaman başlayacağını belirler. Projede görevler, öncel (predecessor) ve ardıl (successor) ilişkilerle birbirine bağlanır. Bu ilişkiler, bir görevin başlamadan önce belirli başka görevlerin tamamlanması gerektiğini belirtir ve proje planlamasında önemli bir rol oynar [70]. Ancak bu kısıtlama, gerçek hayattaki projelerin ilerleyişi sırasında ortaya çıkabilecek yeni bilgilere ve değişen koşullara uyum sağlamayı zorlaştırır. Bu durum, proje çizelgesinin güncellenmesi ve öncellerin yeniden değerlendirilmesi ihtiyacına yol açabilir.
- Kaynak kısıtlamaları ise, projelerde kullanılabilir kaynakların sınırlı olduğu durumlarda görevlerin planlanmasını etkileyen bir başka kısıtlama türüdür. Her bir görev, belirli bir süre boyunca belirli miktarda kaynak gerektirir ve eşzamanlı olarak yürütülen görevlerin

toplam kaynak kullanımı, herhangi bir zamanda kullanılabilir kaynakları aşamaz. Başka bir deyişle, projede aynı anda gerçekleştirilen çeşitli görevler için gerekli olan toplam kaynak ihtiyacı, o zaman diliminde projeye ayrılan kaynak limitini aşamaz. KKPÇP'de, bu tür kısıtlamalar göz önünde bulundurularak görevlerin uygun bir şekilde planlanması gerekir [71].

### 3.3. Amaç Fonksiyonları

ÇY-KKPÇP, projelerin tamamlanmasında karşılaşılan iki temel zorluğu ele alır:

- Kullanılabilir kaynakların sınırlı olması,
- Bu kaynakların çeşitli yeteneklere sahip olması.

ÇY-KKPÇP, projenin zaman ve kaynak kullanımını optimizasyon yoluyla iyileştirmeyi hedefleyerek, sürecin daha verimli ve etkili bir şekilde yönetilmesini sağlar. Optimizasyon süreci, projenin süresinin kısaltılmasına, maliyetlerin azaltılmasına, kaynak kullanımının daha etkin hale getirilmesine ve olası gecikmelerin minimuma indirilmesine odaklanır [72].

ÇY-KKPÇP çözümlerinin temelini oluşturan amaç fonksiyonları, optimizasyon sürecinin odak noktasını belirler. Bu fonksiyonlar arasında en yaygın olanları aşağıda belirtilmiştir [73], [65]:

- Proje süresinin minimize edilmesi: Bu amaç fonksiyonu, projenin tamamlanması için gereken süreyi en aza indirmeye odaklanır. Bu, projenin daha hızlı tamamlanmasını ve piyasaya sürülme süresinin kısalmasını sağlar.
- Proje maliyetinin minimize edilmesi: Bu amaç fonksiyonu, projenin tamamlanması için gereken maliyeti en aza indirmeye odaklanır. Bu, projenin daha az kaynak kullanarak tamamlanmasını ve karlılığının artmasını sağlar.

Yukarıda bahsedilen amaç fonksiyonlarının yanı sıra, birden fazla kritere odaklanan çok amaçlı optimizasyon yaklaşımları da kullanılabilmektedir. Bu yaklaşımlar projenin tamamlanma süresini ve maliyetini aynı anda minimize ederek, projenin farklı boyutlarını optimize etmeyi hedefler [74]:

- Ağırlıklı toplam maliyet: Bu yaklaşım, projenin süresi ve maliyeti gibi birden fazla faktörü, belirlenen ağırlıklar aracılığıyla dengeler ve her iki kriteri de aynı anda minimize etmeyi hedefler. Bu, proje bütçesinin azaltılmasın yanı sıra ve zamanında tamamlanmasının da desteklenmesini sağlar.
- Pareto optimal çözümler: Pareto etkinliği, bir amacın geliştirilmesinin sadece diğer bir amacın zarar görmesiyle mümkün olabileceği durumları tanımlar. Bu metodoloji, projenin süresi ile maliyeti arasında en uygun dengenin sağlanmasını hedefler, böylelikle karar vericilerin farklı seçenekler arasından tercih yapabilmelerine olanak tanır.

# 4. KULLANILAN YÖNTEMLER

Etkin bir yazılım proje çizelgesi oluşturmak için projenin görev bağımlılıklarının belirlenmesi, görevlerin insan kaynaklarının yetkinlik ve deneyimlerine göre atanması ve görevlerin yaklaşık tamamlanma sürelerinin tahmin edilmesi gerekmektedir. Bu yapılırken, proje bütçesi ve hedeflenen proje tamamlanma süresi aşılmamalıdır. Tüm bu faktörler arama uzayını çok büyük (NP-zor) yapar ve çözüm süresini azaltmak için çeşitli yapay zekâ teknikleri uygulanabilir [75-76].

Tez çalışmasında, ÇY-KKPÇP'nin çözümü için hibrit bir yöntem geliştirilmiştir. İnsan kaynakları ve görevler ile bu varlıklar arasındaki ilişkiler kullanılarak, projenin tamamlanma süresinin kısaltılması, bütçenin düşürülmesi ve insan kaynaklarının boşta bekleme sürelerinin en aza indirgenmesi hedeflenmiştir. Bu çalışma ile en uygun yazılım proje çizelgesi elde edilmeye çalışılmıştır. Test sürecinde iMOPSE veri seti kullanılmış olup, proje süresi, proje bütçesi ve insan kaynaklarının boşta bekleme süresi amaç fonksiyonları ayrı ayrı ele alınmıştır. Denklemlerde kullanılan notasyonlar ve açıklamaları Çizelge 4.1'de verilmiştir.

Projedeki görevleri icra edebilmek için, görev ve insan kaynağı atamasının yetkinlik seviyelerine uygun şekilde gerçekleştirilmesi gerekmektedir. Bir görevin tamamlanması için gerekli yetkinliğe sahip en az bir Q çalışan olmak zorundadır. p adet görevin q adet insan kaynağıyla gerçekleştirilmek istendiği bir projede, $\forall i \in \{1,2,...,p\}, \exists j \in \{1,2,...,q\}$ iken $T_i$ görevinin $E_j$ kaynakları ile gerçekleştirilebilmesi için Eş. 4.1'in sağlanmış olması gerekmektedir.

$$MQ_i \leq QD_j \tag{4.1}$$

Eş. 4.2, projedeki tüm görevleri tamamlayabilmek için uygun yetkinlikte insan kaynaklarına sahip olunması zorunluluğunu ifade etmektedir. $skill(E_j,TQ_i)$fonksiyonu, $E_j$ insan kaynağının $TQ_i$ görevi için gerekli olan yetkinliğe sahip olup olmadığını kontrol eder.

$$\forall T_i \in T, \exists E_j \in E : skill(Ej,TQ_i) \geq MQ_{ij} \tag{4.2}$$

Çizelge 4.1. Denklemlerde kullanılan notasyonlar ve açıklamaları

| Notasyon | Açıklama |
|---|---|
| p | Toplam görev sayısı |
| q | Projede görevli toplam insan kaynağı sayısı |
| $T=\{T_1, T_2,\ldots,T_p\}$ | Projede tamamlanması gereken görevler |
| $E=\{E_1, E_2,\ldots,E_q\}$ | Projede görevlendirilen insan kaynakları |
| $PT=\{T_1, T_2,\ldots,T_k\}$ | Görevin öncelleri: $k \leq p$ |
| $Q=\{Q_1, Q_2,\ldots,Q_l\}$ | İnsan kaynağının sahip olduğu yetenekler: $0 < 1 \leq p$ |
| $QD=\{QD_1, QD_2,\ldots,QD_l\}$ | İnsan kaynağının sahip olduğu yeteneklerin seviyesi |
| $TQ=\{Q_n\}$ | Görevi gerçekleştirmek için gerekli yetenek: $0 < n \leq l$ |
| $MQ=\{QD_m\}$ | Görevi gerçekleştirmek için gerekli asgari yetenek seviyesi: $0 < m \leq l$ |
| $CT=\{CT_1, CT_2,\ldots,CT_p\}$ | Görevlerin tamamlanma süresi |
| $ET=\{ET_1, ET_2,\ldots,ET_q\}$ | Bir görev tamamlanana kadar insan kaynağı tarafından harcanan süre |
| Πx | Bir görevi tamamlamak için gereken süre |

$ET_{ij}$ $i$'inci görev için $j$'inci insan kaynağının harcadığı süre olmak üzere, $i$'inci görevin tamamlanma süresi $CT_i$ Eş. 4.3 ile hesaplanabilir.

$$CT_i = ET_{ij}/n, \quad n=\sum_{j=1}^{q} x_{i,j}, \quad x_{i,j} = \begin{cases} 0, & i.kaynağa\ j.görev\ atanmamış \\ 1, & i.kaynağa\ j.görev\ atanmış \end{cases} \tag{4.3}$$

Eğer bir görevde sadece bir kişi çalışıyorsa, görevin tamamlanma süresi Eş. 4.3'te belirtilen $CT_i$ olur. Eğer görev n kişi tarafından yürütülüyorsa ve her kişi de aynı anda ve aynı verimlilikte çalışabiliyorsa, görevin tamamlanma süresi $CT_i/n$ olur. Bununla birlikte, pratikte genellikle bir görev üzerinde çalışan her bir ek kaynağın, görevin süresini eşit oranda azaltmadığı kabul edilir, çünkü çeşitli etkenler (örneğin, iletişim zorlukları, kaynaklar arası koordinasyon ihtiyacı vb.) bu verimliliği etkileyebilmektedir.

Her görev için ardıl-öncel ilişkileri, görevin başlangıç (*start*($T_i$)) ve bitiş zamanı (*finish*($T_i$)) kullanılarak Eş. 4.4 ile ifade edilir:

$$start(T_i) \geq \max(\{ finish(T_k) \mid T_k \in PT(T_i) \}) \\ finish(T_i) = start(T_i) + CT_{i,n} \quad (4.4)$$

$\max(\{ finish(T_k) \mid T_k \in PT(T_i) \})$, $Ti$ görevinin başlayabilmesi için tamamlanması gereken tüm öncül görevlerin yani $PT(T_i)$'nin bitiş zamanlarının en büyüğünü ifade eder.

Proje tamamlanma süresi(makespan) $\Pi_{xx}$, tüm görevlerin bitiş zamanlarının en büyüğü hesaplanarak belirlenir. Eş. 4.5'te belirtildiği üzere, her bir görevin başlama ve bitiş zamanları göz önünde bulundurulup, görevlerin ardıl-öncel ilişkilerine göre en son biten görevin zamanı belirlenerek projenin toplam tamamlanma süresi hesaplanır.

$$\Pi_{xx}=max(\{finish(T_1),finish(T_2),...,finish(T_p)\}) \quad (4.5)$$

Proje yönetiminde, kaynakların etkin kullanımı ve görev atamalarının optimize edilmesi kritik öneme sahiptir. Bu kapsamda, görevler arası bağımlılıklar ve insan kaynaklarının atamaları değerlendirilirken, görevlerin başlaması için gerekli bekleme süreleri dikkate alınmalıdır. Tez çalışmasında proje planlamasının bir parçası olarak, her görevin öncel şartlarının ve kaynakların mevcut iş yükünün ortaya çıkaracağı boşta bekleme süreleri de analiz edilmektedir.

Bir görevin başlayabilmesi için; görevin öncel görevlerinin tamamlanmış olması ve göreve atanmış insan kaynağının önceki görevini tamamlamış olması gerekmektedir. Bu durumlar, projedeki genel bekleme sürelerini etkilemektedir. Öncel görevlerin tamamlanma zamanlarına bağlı bekleme süresi şöyle tanımlanabilir:

Eğer bir $T_i$ görevi için öncel görevler $PT(T_i)$ belirlenmişse, $T_i$ görevinin başlama zamanı bu öncel görevlerin bitiş zamanlarına bağlıdır. İnsan kaynağı $E_j$, tüm öncel görevler tamamlanana kadar $T_i$ görevine başlayamaz. Bu bekleme süresi Eş. 4.6'da belirtildiği şekilde maksimum öncel görev bitiş zamanının şu anki zamanla farkı olarak ifade edilir:

$$TGBekle = \max(\{ finish(T_k) \mid T_k \in PT(T_i) \}) - current\_time_{ij} \quad (4.6)$$

Diğer yandan, bir insan kaynağı şu anda bir $T_m$ görevi üzerinde çalışıyorsa ve $T_m$ henüz tamamlanmamışsa, kaynağın $T_i$ görevine başlamadan önce $T_m$ görevini bitirmesi gerekmektedir. Bu senaryoda bekleme süresi, Eş. 4.7'de belirtildiği üzere kaynağın şu anda üzerinde çalıştığı görevin bitiş zamanının şu anki zamanla farkı olarak ifade edilir:

$$TGBekle_{jm}=finish(T_m)-current\_time \tag{4.7}$$

Eş. 4.8'de her iki durumun ortaya çıkardığı maksimum boşta bekleme süresi tanımlanmıştır.

$$TGBekle_j=\max(TGBekle_{ij},TGBekle_{jm}) \tag{4.8}$$

Bu hesaplamalar, insan kaynaklarının görevlere atanmasında stratejilerin belirlenmesi ve projenin zaman çizelgesinin optimize edilmesi açısından önem taşımaktadır.

## 4.1. Kaotik Rastgele Sayı Üretimi

Yapay zekâ optimizasyon algoritmalarında, durum uzayı etkin bir şekilde taranarak en iyi çözüm elde edilmeye çalışılır. Bunun için, arama yapılan bölgelerde en iyi sonucu ifade eden yerel optimumlara takılmadan, tüm durum uzayında bulunabilecek en iyi çözüm olan global optimuma yakınsanmaya çalışılır. Bu durum, algoritmanın başarısını değerlendirmede önemli bir faktördür [77]. Arama sürecinde durum uzayında gezinmenin kalitesi, yapay zekâ optimizasyon algoritmasının keşfetme ve sömürme (exploration and exploitation) davranışları üzerinde önemli bir etkiye sahiptir.

Kaotik sistemler, doğrusal olmayan dinamik sistemlerin bir parçasıdır ve kaotik bölgede öngörülemeyen davranışlar gösterirler [78]. Kaotik sistemlerin bu özelliği, optimizasyon algoritmalarında çeşitli avantajlar sağlar [79–81]:

- Çeşitlilik ve keşif yeteneği: Kaotik sistemler başlangıç koşullarına duyarlıdır ve benzersiz sayı dizileri üretirler. Bu sayede büyük ve karmaşık çözüm alanlarının etkili bir şekilde taranmasında kritik bir rol oynarlar.
- Yerel minimumlardan kaçış: Kaotik sistemlerin tekrarlanmayan ve tahmin edilemeyen sayı dizileri üretmesi, algoritmanın yerel çözümlerden çıkıp daha iyi çözüm bölgelerine geçiş yapmasına olanak tanır, böylece global optimuma yakınsama şansı arttırılır.

- Yakınsama hızı: Durum uzayındaki arama yetenekleri sayesinde kaotik sistemler, optimizasyon algoritmalarının global optimuma yönelik daha doğrudan ve etkin bir arama yapmasını sağlar. Bu da yakınsama hızını önemli ölçüde artırır.

Olumlu katkıların yanı sıra kaotik sistemlerin öngörülemez bir yapıya sahip olması, belirli parametreler veya koşullar altında optimizasyon algoritmasının sürekli olarak farklı çözümler arasında geçiş yapmasına ve algoritmanın kararlılığının azalmasına neden olabilmektedir. Kaotik davranışın kontrol edilememesi, algoritmanın sürekli olarak iyi çözümlerden uzaklaşmasına ve genel optimizasyon kalitesinin düşmesine sebep olabilmektedir. Ayrıca kaotik sistemlerin kullanımının algoritmanın hesaplama maliyetine olumsuz etkileri de bulunmaktadır [82].

Tez çalışmasında kaotik sistemlerin temel özelliklerinden yararlanılıp, bunu yaparken de dezavantajlarının oluşturabileceği etkiler minimumda tutulmaya çalışılarak rastgele sayı üreteci tasarlanmıştır [83-84]. Kaotik fonksiyon olarak Eş. 4.9'da belirtilen kaotik lojistik harita kullanılmıştır [85].

$$x_{n+1} = r x_n (1 - x_n),\ 0 < r \leq 4 \tag{4.9}$$

Kaotik lojistik harita, popülasyon yönetimi, optimizasyon ve kriptografi alanlarında kullanılan basit bir yinelemeli denklemdir. Basit bir matematiksel ifadeye sahip olmasına rağmen, kaotik sistemlerin doğal davranışlarından kaynaklanan öngörülemez ve karmaşık davranışlar sergilemektedir. Eş. 4.9'daki $x_n$, n. yinelemedeki popülasyon yoğunluğunu, r ise kontrol parametresini ifade eder. Kontrol parametresi kaotik lojistik haritanın davranışlarını belirleyen ana faktördür ve tanımlı olduğu aralıkta aşağıda belirtilen davranışları sergiler [86]:

- $r < 1$ durumunda, sistemde kaotik davranış gözlenmez ve sistem basit bir davranış sergiler. Popülasyon sürekli olarak azalır ve sıfıra doğru yakınsar. Bu durum, yeterince kaynak olmadığını, popülasyonun devam ettirilemeyeceğini ve nihayetinde de tükeneceğini ifade eder.

- $1 < r < 3$ durumunda, kaotik sistem 0'dan farklı sabit bir noktaya yakınsar ve sistem stabil kalır. Başlangıç popülasyon büyüklüğü ne olursa olsun zamanla popülasyon büyüklüğü Eş. 4.10'da ifade edilen değere yakınsar ve dengeli davranışlar sergiler.
  $N = K * ((r - 1)/r)$ (4.10)
- $3 < r < 3.57$ durumunda, sistem düzenden kaosa geçiş yapar ve periyodik davranışlar göstermeye başlar. Başlangıçta period sayısı 2'dir. r değeri arttıkça period sayısı katlanarak artar ( 4, 8, 16, 32…). Her bir katlanma popülasyonun gidip geldiği döngüsel durumların artması anlamına gelir. r değeri 3.57'ye yaklaştıkça periyodik katlanmalar daha sık hale gelir.
- $r > 3.57$ durumunda lojistik harita kaotik davranışlar sergiler. Başlangıç koşullarına (popülasyon değerlerine) aşırı duyarlılık söz konusu olur ve başlangıç koşullarındaki çok küçük değişiklikler, zamanla büyük farklılıklara yol açar, bu da sistemin uzun süreli davranışının öngörülemez olması sonucunu ortaya koyar.
- r=4 durumunda lojistik harita tamamen kaotik bir duruma girer ve bu noktada, sistem en yüksek derecede kaotik davranışı sergiler. Bu, sistemin çıktılarının tamamen rastgele gibi göründüğü ve herhangi bir düzenli veya tekrarlanabilir desenin olmadığı anlamına gelir.

Lojistik haritanın kaotik özellik göstermesi başlangıç değerine ($x_n$) ve kontrol parametresine ($r$) bağlıdır. Bu fonksiyon her başlangıç değeri ve her kontrol parametresi için kaotik özellik göstermeyebilir. Eş. 4.11'de belirtilen Lyapunov üsteli kullanılarak bir sistemin kaotik özellik gösterip göstermediği yorumlanabilmektedir [78].

$$L(c) = \lim_{n\to\infty} \left(\frac{1}{n}\right) \log 2 \left|f'^{n}_{r}(x)\right| \quad (4.11)$$

Tez çalışmasında kullanılan kaotik rastgele sayı üretecine ait sözde kod aşağıda belirtilmektedir [87]:

Kaotik rastgele sayı üreteci algoritması

Girdi: Kaotik sistemin parametreleri
Adım 1: Lojistik harita N döngü boyunca çalıştırılır.
lojistik_harita(N)

Adım 2: Varılan nokta için Lyapunov üsteli hesaplanır.

lyapunov_usteli = hesapla_lyapunov_usteli(nokta)

Adım 3: Lyapunov üsteli pozitif ise sistem kaotik davranış göstermeye adaydır.

```
if lyapunov_usteli > 0:
        kaotik_davranis = true
else:
        kaotik_davranis = false
```

Adım 4: a, b, c Çizelge 4.2'de gösterilen 64 bit olarak tanımlanır.

$a = rx_n(1 - x_n), x_n = a$

$b = rx_n(1 - x_n), x_n = b$

$c = rx_n(1 - x_n), x_n = c$

Adım 5: Sayıların Mantis 1 kısmı alınır ve kaotik sayı hesaplanır.

X=Mantis1(a) $\oplus$ Mantis1(b) $\oplus$ Mantis1(c)

Çıktı: X % Popülasyon Sayısı

Çizelge 4.2. IEEE 754-2008 64 bit çift duyarlı gösterim formatı

| 1 bit | 11 bit | 20 bit | 32 bit |
|---|---|---|---|
| İşaret biti | Üs | Mantis 0 | Mantis 1 |

## 4.2. Amaç Fonksiyonları

Yapay zekâ optimizasyon algoritmalarında amaç fonksiyonu, algoritmanın ne elde etmeyi amaçladığını tanımlayan bir fonksiyondur. Algoritma her bir çözümün amaç fonksiyonu değerini hesaplar ve en uygun değere sahip çözümü bulmaya çalışır [88].

Yazılım proje yönetimi problemlerini etkin bir şekilde çözebilmek için tez çalışması kapsamında geliştirilen yöntemde üç farklı amaç fonksiyonu kullanılmaktadır. Bu amaç fonksiyonları, proje süresi, proje bütçesi ve kaynakların boşta bekleme zamanını minimize etmeyi hedeflemektedir. Tez çalışmasında kullanılan amaç fonksiyonlarının detayları aşağıda belirtilmektedir:

- Proje tamamlanma süresi: Projenin toplam tamamlanma süresini azaltmayı amaçlayan bir fonksiyondur. Bu fonksiyon, zaman kısıtının önemli olduğu durumlarda, projenin

verimli şekilde yönetilmesini sağlamak için projede başlangıçtaki görevden son görev tamamlanıncaya kadar geçen zamanın en aza indirgenmesini hedefler.

- Proje bütçesi: Projenin toplam maliyetini düşürmeyi hedefler. Bu fonksiyon, projedeki tüm görevler için gerekli insan kaynağı maliyetlerini minimize etmeye çalışır. Maliyet minimizasyonu, özellikle bütçe kısıtlamaları olan projelerde önemlidir ve maliyet verimliliği sağlayarak projenin finansal açıdan sürdürülebilirliğini artırır.
- Kaynakların boşta bekleme zamanı: Projede kullanılan insan kaynaklarının mümkün olduğunca verimli kullanılmasını amaçlayan bir fonksiyondur. Bu fonksiyon, kaynakların âtıl kalmadan, projenin çeşitli aşamalarında etkin bir şekilde kullanılmasını hedefler. Kaynak kullanımını optimize etmek hem zaman hem maliyet hem de insan kaynağı motivasyonunu arttırması açısından projenin genel verimliliğini artırır.

### 4.3. Uygunluk Değeri Hesaplama

Yapay zekâ algoritmalarında uygunluk değeri, bir çözümün ne kadar etkili ve başarılı olduğunu ölçen kritik bir değerdir. Bu değer, bir algoritmanın farklı çözümleri arasında seçim yapmasına ve en iyi sonuca ulaşabilmesine yardımcı olur. Algoritma, her bir çözümün uygunluk değerini hesaplar ve daha yüksek değere sahip çözümleri tercih eder. Bu sayede, algoritma zamanla daha iyi çözümlere doğru ilerler.

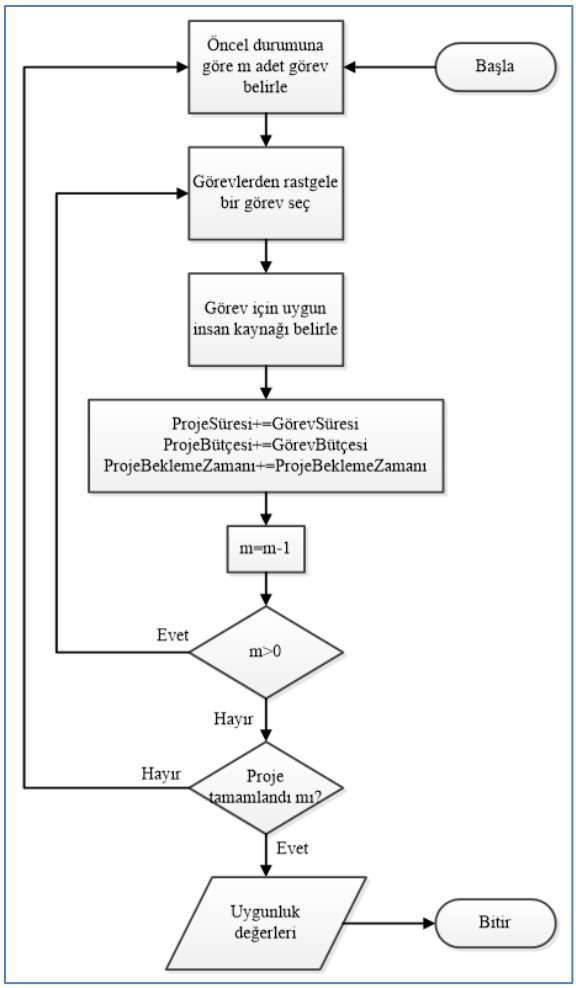
Uygunluk değeri, problemin çözümünde hedeflenen değere ne kadar yakınsandığını gösteren bir parametredir. Tez çalışmasında kullanılan amaç fonksiyonları optimize edilirken, her biri için minimum değere yakınsanmaya çalışılır. Görevler icra edilirken, her bir fonksiyonun o ana kadar elde ettiği değer belirlenir. Son göreve gelindiğinde, artık proje tamamlanmıştır ve projenin bütçesi, süresi ve insan kaynaklarının boşta bekleme süresi için nihai değere ulaşılır. Bu aşamada, hesaplanan uygunluk değeri ile değerlendirme yapılarak, sonucun iyileşip iyileşmediği belirlenir. $TS\ddot{u}re_a$, a. görevin tamamlanma süresi, $TB\ddot{u}t\c{c}e_a$ a. görevin insan kaynağı maliyeti, $TBekle_a$ a. görevin başlaması için ortaya çıkan insan kaynaklarının boşta bekleme zamanı ve $p$ projedeki toplam görev sayısı olmak üzere; oluşturulan çizelgeye ait proje tamamlanma süresine ($PS\ddot{u}re$), proje tamamlanınca ortaya çıkan insan kaynakları maliyetine ($PB\ddot{u}t\c{c}e$) ve insan kaynaklarının boşta bekleme sürelerinin toplamına ($PBekle$) karşılık gelen uygunluk değerleri Eş. 4.12, Eş. 4.13, Eş. 4.14'te belirtildiği şekilde hesaplanmaktadır.

$$PSüre = \sum_{a=1}^{p} TSüre_a \qquad uygunluk_{Süre} = {}^{1}/_{PSüre} \tag{4.12}$$

$$PBütçe = \sum_{a=1}^{p} TBütçe_a \qquad uygunluk_{Bütçe} = {}^{1}/_{PBütçe} \tag{4.13}$$

$$PBekle = \sum_{a=1}^{p} TBekle_a \qquad uygunluk_{Bekle} = {}^{1}/_{PBekle} \tag{4.14}$$

Uygunluk değerinin hesaplanmasında kullanılan algoritmanın akış şeması Şekil 4.1'de yer almaktadır.




Şekil 4.1. Uygunluk değeri hesaplama

## 4.4. Genetik Algoritma

Genetik Algoritma (GA), doğal evrim süreçlerini modelleyen bir meta-sezgisel optimizasyon yöntemidir. Bu algoritma, Charles Darwin'in doğal seçilim teorisinden esinlenmiştir ve popülasyon içindeki bireylerin (çözüm adaylarının) genetik operasyonlarla

sürekli olarak iyileştirilmesi esasına dayanır. GA, özellikle karmaşık arama uzaylarına sahip optimizasyon problemlerinde etkilidir çünkü çeşitlilik içeren bir popülasyon üzerinde çalışır ve geniş bir arama alanını kapsar [89].

GA'nın temel bileşenleri ve işleyişi aşağıda belirtilmiştir:

Popülasyon oluşturma: Başlangıçta rastgele oluşturulan bir popülasyon ile süreç başlar. Her bir birey, problemi çözmek için potansiyel bir sonucu temsil eden bir kromozom ile ifade edilir.

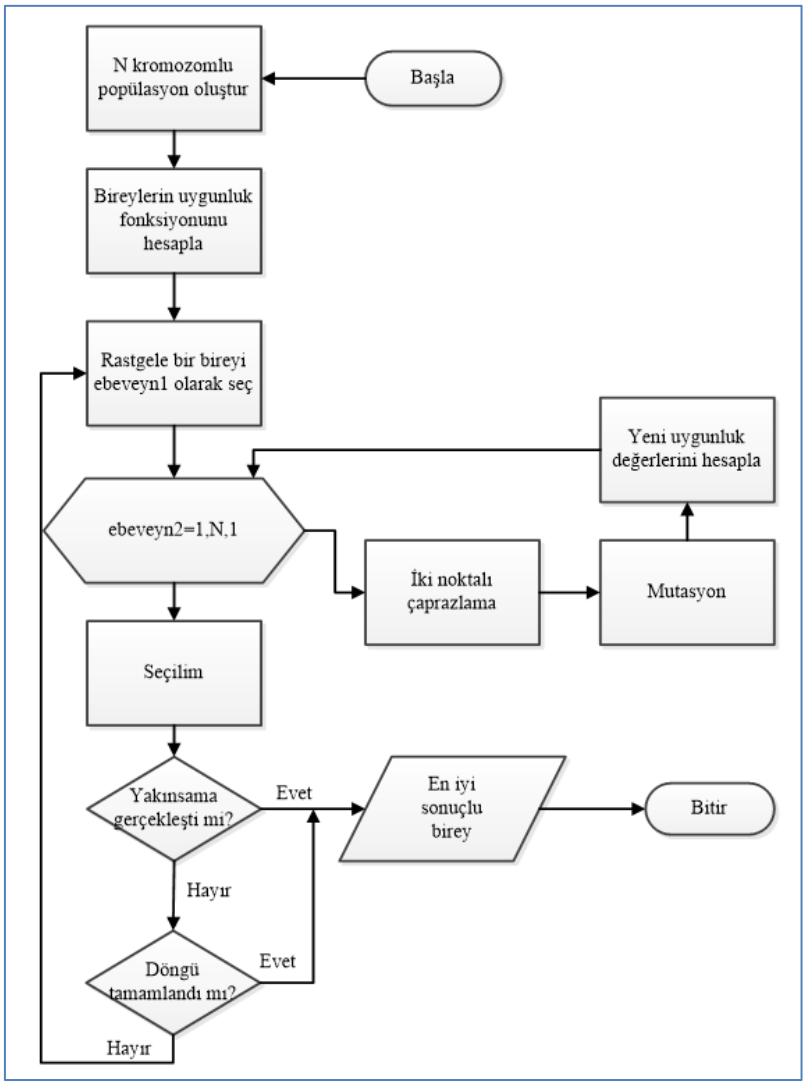
Uygunluk fonksiyonu: Her bireyin ne kadar iyi performans gösterdiğini belirlemek için bir uygunluk fonksiyonu kullanılır. Bu fonksiyon, bireyin problem çözümündeki başarısını ölçer ve doğal seçilim sürecinde hangi bireylerin devam edeceğini belirler.

Genetik operatörler: Üç ana genetik operatör kullanılır [90]:

- Çaprazlama: Genetik çeşitliliği artırmak ve yeni çözüm adayları oluşturmak için iki bireyin genetik bilgisinin birleştirilmesi işlemidir.
- Mutasyon: Popülasyonda rastgele değişiklikler yaparak yeni genetik yapılar oluşturmak ve böylece algoritmanın yerel minimumlarda takılmasını önleme işlemidir.
- Seçilim: Uygunluk fonksiyonuna göre en iyi performans gösteren bireylerin bir sonraki nesle aktarıldığı, daha az uygun olanların ise elemesinin yapıldığı süreçtir.

GA, büyük durum uzaylarında etkin arama yapılabilmesi için geniş çeşitlilik sunar. Farklı türde problemlere uygulanabilme esnekliği vardır. Çözüm süreçlerinin paralelleştirilmesi kolaydır. Bu nedenle de büyük durum uzayına sahip problemlerde sonuca hızla ulaşılabilme imkânı sağlar. Ancak, çaprazlama ve mutasyon oranlarının doğru ayarlanmaması halinde yerel optimumlarda takılma riski bulunur ve algoritmanın performansı parametre ayarlarına (çaprazlama oranı, mutasyon oranı, popülasyon büyüklüğü) oldukça duyarlıdır, bu yüzden de bu parametrelerin dikkatlice seçilmesini gerektirir [91].

Tez çalışmasında GA temelli yöntemlerin uygulanmasında kullanılan algoritmaların genel akış şeması Şekil 4.2'de belirtilmektedir [92].




Şekil 4.2. Genetik algoritma akış şeması

## 4.5. Yapay Arı Kolonisi Algoritması

Yapay Arı Kolonisi Algoritması (YAKA), doğadaki arıların yiyecek kaynaklarını arama ve bulma davranışlarını modelleyen bir meta-sezgisel optimizasyon yöntemidir. Gerçek arılar, çiçeklere giderek nektar toplar ve bu nektarın miktarını değerlendirirler. Nektarın çok olduğu çiçekler daha değerli kaynaklar olarak kabul edilir ve arılar bu bilgiyi kovanlarına dönerken diğer arılara aktarırlar. YAKA, bu nektar değerlendirme süreçlerini esas alır.

YAKA'da, işçi arılar, kâşif arılar ve gözcü arılar olmak üzere üç farklı arı rolü bulunur. İşçi arılar mevcut çözümleri geliştirirken, kâşif arılar potansiyel yeni çözüm alanlarını keşfeder. Gözcü arılar ise çözüm kalitesine göre işçi arıları bilgilendirir ve yönlendirir.

YAKA, basitlik oluşturması açısından zorunlu olmayan bir takım temel kabuller içerir [93]:

- Her bir yiyecek kaynağı sadece bir arı tarafından kullanılır.
- İşçi arı sayısı yiyecek kaynağı sayısına eşittir.
- İşçi arı sayısı gözcü arı sayısına eşittir.
- İşçi arı, yiyecek kaynağını tüketmişse kâşif arı türüne dönüşür.

YAKA'da başlangıçta, kâşif arılar rastgele yiyecek ararlar. Bir yiyecek kaynağı bulan arı, işçi arıya dönüşür ve bulduğu yiyeceği kovana taşımaya başlar. Kovana varan işçi arı, yiyecek taşımaya devam edebilir veya kovanda, bulduğu yiyecek kaynağının yerini belirten bir dans yaparak diğer arılara bilgi verebilir. Bu dansı yorumlayan gözcü arılar, yiyecek kaynağının kalitesi ve konumu hakkında bilgi edinir ve bu bilgilere dayanarak kaynak tercihinde bulunurlar. Arılar, bu süreç boyunca yiyecek kaynaklarından kovana sürekli olarak yiyecek taşır. Yiyecek kaynağı tükendiğinde, yiyecek taşıyan işçi arılar tekrar kâşif arıya dönüşür ve yeni yiyecek kaynakları aramaya başlar [94].

Arılar, sürekli olarak çevrelerindeki nektar kaynaklarını değerlendirir ve bu kaynaklar tükendiğinde veya daha verimli kaynaklar bulunduğunda, işçi arılar tekrar kaşif arıya dönüşebilir. Bu dönüşüm, durum uzayında geniş arama yapabilme kabiliyeti getirerek, yerel optimumlara takılma riskini azaltır ve global optimuma ulaşma şansını arttırır [95].

YAKA, farklı tür problemlere kolayca uyarlanabilir ve çözüm sürecinde beklenmedik durumlara karşı dayanıklılık gösterir. Hızlı yakınsama özelliği vardır ve benzer algoritmalara oranla daha az kontrol parametresi kullanımını gerektirir. Çok modlu (birden fazla yerel minimum veya maksimum noktası olan fonksiyonlar) ve çok değişkenli problemlerin çözümünde etkin sonuçlar üretir [95–97].

Bununla birlikte YAKA popülasyon çeşitliliğini sağlamada yetersizdir, yerel optimuma takılma ve yavaş yakınsama davranışı gösterebilir [98].

## 4.6. Bozkurt Optimizasyon Algoritması

Bozkurt optimizasyon algoritması (BO), bozkurtların sürüdeki hiyerarşi ve avlanma yöntemlerini modelleyen ve özellikle proje çizelgeleme, kaynak tahsisi ve optimizasyon problemleri gibi alanlarda etkili çözümler sunan meta-sezgisel bir algoritmadır. Bu algoritma, bozkurtların avlanma davranışlarını taklit eder ve bu davranışları problem çözme sürecine uygular. Yöntem genel olarak aşağıda belirtilen temel özelliklere sahiptir [19,24]:

Sosyal hiyerarşi: Sürüdeki alfa, beta ve delta kurtları en iyi çözümleri temsil eder. Bu kurtlar sürünün liderleri olarak kabul edilir ve avlanma sırasında diğer kurtlara yol gösterir. Alfa kurt en iyi çözümü, beta ikinci en iyi çözümü ve delta ise üçüncü en iyi çözümü temsil eder.

Yiyecek arama ve takip davranışı: Sürünün diğer üyeleri olan omega kurtlar, bu lider kurtların belirlediği yönlerde hareket eder. Omega kurtlar, alfa, beta ve delta kurtlarının belirlediği potansiyel av yerlerine doğru ilerleyerek yiyecek arama sürecine katılır.

Avın yerini tahmin etme: Alfa, beta ve delta kurtları, avın potansiyel yerlerini tahmin eder. Bu tahminler, çözüm uzayında daha iyi bölgelere doğru iteratif bir şekilde ilerlemeyi sağlar. Omega kurtlar bu tahminlere göre hareket eder ve böylece çözüm uzayının geniş bir bölümü taranmış olur.

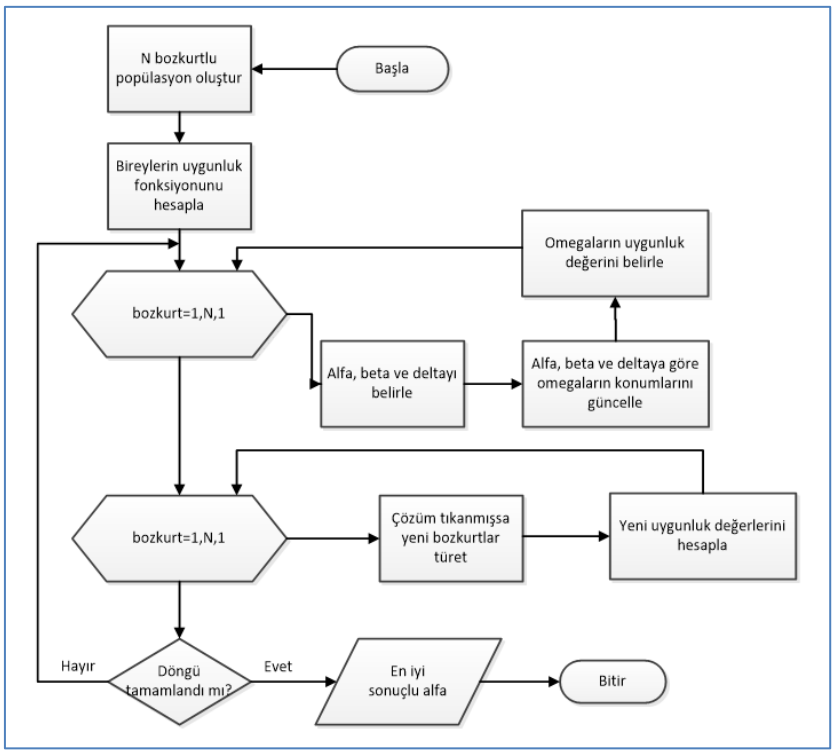
Kontrol parametreleri: Algoritmanın etkinliğini artırmak ve algoritmik durgunluk durumlarından kaçınmak için çeşitli kontrol parametreleri kullanılır. Bu parametreler, algoritmanın yakınsama hızını ve çözüm kalitesini optimize etmeye yardımcı olur.

BO, yiyecek arama ve takip etme özellikleri sayesinde, durum uzayının geniş bir bölümünün taranmasını ve global optimuma ulaşma olasılığının artırılmasını sağlar. Parametre sayısının nispeten az olması, kullanım kolaylığı ve işlem süresinin daha az olması avantajlarını getirir. Yöntemdeki sosyal hiyerarşi çözüm aşamasında hızlı sonuç alınabilmesine olanak sağlar [19].

BO'da sürü üyeleri, lider kurtların konumlarına göre hareket ettiğinden, eğer bu lider kurtlar yeterince iyi çözümler bulamaz ve durum uzayının geniş bir bölümünü tarayamazsa, çözümde yerel minimuma takılma problemi ortaya çıkar. Bununla birlikte, sürüdeki her

bireyin durumunu güncellemek ve yönetmek algoritmanın hesaplama maliyetini arttırır. Algoritma, sürüdeki lider sayısı, sürü büyüklüğü gibi parametrelere aşırı duyarlıdır ve uygun seçilmediği takdirde algoritmanın verimliliğini azaltır [24].

Tez çalışmasında kullanılan BO algoritmasının genel yapısı Şekil 4.3'te gösterilen akış şemasında belirtilmiştir.




Şekil 4.3. BO algoritması akış şeması

## 4.7. Veri Seti: iMOPSE

iMOPSE veri seti, ÇY-KKPÇP için detaylı ve gerçek dünya projelerine uygun verileri içeren bir test kütüphanesidir. Bu veri seti, projelerdeki görevleri ve kaynakları etkin şekilde kullanmak amacıyla çeşitli algoritmaların test edilmesi için tasarlanmıştır. Veri seti ile farklı çözüm senaryoları test edilerek algoritmalarının performansı değerlendirebilmekte ve optimizasyon gerçekleştirilebilmektedir.

iMOPSE veri seti, projelerin içerdiği görevleri ve görevleri gerçekleştirebilecek insan kaynaklarının ayrıntılı bilgilerini içerir. Bu bilgiler arasında, her görev için belirlenen süre, görevi gerçekleştirebilmek için gerekli yetkinlikler ve önceki görevlere olan bağımlılıklar

yer alır. Ayrıca, kaynaklar belirli yetkinliklere göre sınıflandırılmıştır, bu da kaynak yönetiminin karmaşıklığını artırır [7,22].

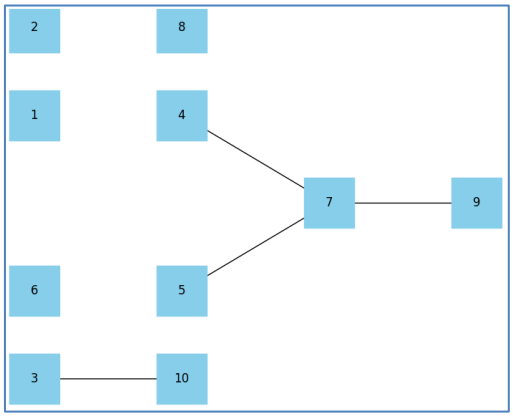
Tez çalışmasında ÇY-KKPÇP'nin çözümü için, her biri farklı büyüklük ve yapıya sahip 36 adet proje verisi ve 6 adet test proje verisinden oluşan iMOPSE veri seti kullanılmıştır. Veri seti isimlendirme yapısı Çizelge 4.3'te belirtilmektedir.

Çizelge 4.3. Veri seti isimlendirme (g_ik_ö_y)

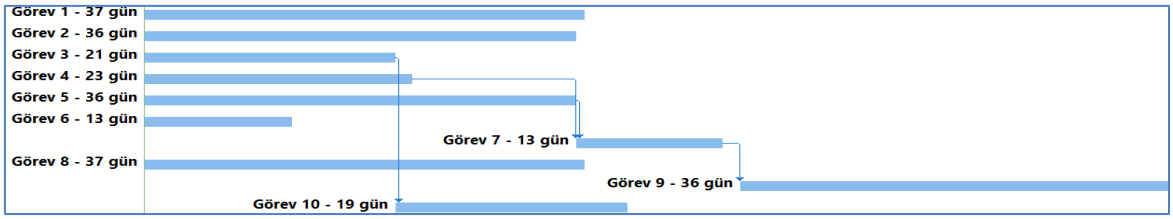
| Kısaltma | Açıklama |
|---|---|
| g | Görev |
| ik | İnsan kaynağı |
| ö | Öncel |
| y | Yetkinlik seviyesi |

Veri setinde her bir görev için görevin tamamlanma süresi tanımlıdır. Görevlerin birbirine bağımlılıkları, görevlerin başlama koşulunu belirler.

Şekil 4.4'te, iMOPSE 10_3_5_3 eğitim veri seti içerisinde yer alan görevlere ait bağımlılıklar, Şekil 4.5'te görevlerin işlem sırası gösterilmektedir. 7, 9 ve 10 numaralı görevler, öncelleri olan görevlerdir ve bu görevlerin başlayabilmesi için öncellerinin tamamlanmış olması zorunluluğu bulunmaktadır.




Şekil 4.4. 10_3_5_3 eğitim veri setindeki görevler ve bağımlılıkları




Şekil 4.5. 10_3_5_3 eğitim veri setindeki görevlerin işlem sırası

Veri setinde her bir insan kaynağı için yetkinlik seviyesi tanımlanmıştır ve bu yetkinlik seviyeleri insan kaynaklarının hangi görevleri gerçekleştirebileceğini belirler. İnsan kaynaklarının saatlik ücretleri de tanımlanır. Çizelge 4.4 ve Çizelge 4.5'te 10_3_5_3 veri setinde tanımlanmış insan kaynağı bilgileri ve görevler için gerekli insan kaynağı yetkinlik seviyesi ilişkileri belirtilmektedir.

Çizelge 4.4. 10_3_5_3 veri setindeki ücret ve yetkinlik seviyesi ilişkisi

| | Ücret (Saat) | Yetkinlik Seviyesi | | |
|---|---|---|---|---|
| | | $Y_0$ | $Y_1$ | $Y_2$ |
|  | 56.0 | - | 0 | 1 |
|  | 53.6 | 1 | - | 2 |
|  | 28.9 | 1 | 0 | - |

Çizelge 4.5. 10_3_5_3 veri setindeki görev, insan kaynağı ve yetkinlik seviyesi ilişkisi

| Görev | Yetkinlik Seviyesi |  |  |  |
|---|---|---|---|---|
| 0 | $Y_2$: 1 | 1 | 2 | - |
| 1 | $Y_1$: 2 | - | 2 | - |
| 2 | $Y_0$: 1 | - | 1 | 1 |
| 3 | $Y_1$: 0 | 0 | - | 0 |
| 4 | $Y_0$: 1 | - | 1 | 1 |
| 5 | $Y_2$: 1 | 1 | 2 | - |
| 6 | $Y_1$: 0 | 0 | - | 0 |
| 7 | $Y_0$: 1 | - | 1 | 1 |
| 8 | $Q_2$: 1 | 1 | 2 | - |
| 9 | $Q_1$: 0 | 0 | - | 0 |

36 adet veri seti içerisinden literatürde en çok kullanılan ve Çizelge 4.6'da belirtilen 10 adet proje verisi tez çalışmasında kullanılmak üzere seçilmiştir.

Çizelge 4.6. Kullanılan veri setleri

| Veri Seti Adı | Görev Sayısı | İnsan Kaynağı | Öncel Sayısı | Yetkinlik Türü |
|---|---|---|---|---|
| 100_5_22_15 | 100 | 5 | 22 | 15 |
| 100_10_26_15 | 100 | 10 | 26 | 15 |
| 100_10_47_9 | 100 | 10 | 47 | 9 |
| 100_20_46_15 | 100 | 20 | 46 | 15 |
| 100_20_65_9 | 100 | 20 | 65 | 9 |
| 200_10_50_15 | 200 | 10 | 50 | 15 |
| 200_20_54_15 | 200 | 20 | 54 | 15 |
| 200_20_55_9 | 200 | 20 | 55 | 9 |
| 200_40_91_15 | 200 | 40 | 91 | 15 |
| 200_40_133_15 | 200 | 40 | 133 | 15 |

iMOPSE veri setinin kullanımı, ÇY-KKPÇP gibi karmaşık çizelgeleme problemlerini çözmek için geliştirilen algoritmaların doğruluk ve etkinliğini test etmek amacıyla kritik öneme sahiptir. Bu veri seti, tez çalışmasında ÇY-KKPÇP'nin çözümü için geliştirilen optimizasyon modelinin performansını değerlendirmede kullanılmıştır.

# 5. HİBRİT BİR YAKLAŞIM İLE YAZILIM PROJE ÇİZELGESİ OLUŞTURMA

Tez çalışmasında BO algoritması temelli bir ana model kullanılmış olup, bu modele GA, YAKA ve kaotik lojistik fonksiyon entegre edilerek hibrit bir optimizasyon yöntemi oluşturulmuştur.

Bu gelişmiş hibrit model, ÇY-KKPÇP'nin çözümünde uygulanmıştır. Model, yazılım projelerinde insan kaynağı atama süreçlerinin optimizasyonunu gerçekleştirirken, projenin zaman, bütçe ve insan kaynaklarının boşta bekleme sürelerini minimize etmeyi hedeflemektedir. Böylece, maliyetlerin düşürülmesi ve proje teslim sürelerinin kısaltılması ile yazılım projelerinin daha verimli ve etkin bir şekilde yönetilmesine katkı sağlanması planlanmaktadır.

Geliştirilen modelde yazılım proje çizelgeleme sürecinde en iyi çözümleri bulma amacıyla BO'daki sürü davranışları simüle edilerek, lider kurtların (alfa, beta ve delta) yer aldığı bir hiyerarşik yapı kurulmuştur. Modelde alfa kurt, mevcut en iyi proje çizelgesi çözümünü temsil ederken, beta ve delta sırasıyla ikinci ve üçüncü en iyi çözümleri temsil eder. Bu liderler, çözüm uzayındaki stratejik noktaları işaret eder ve diğer kurtlar (omega), bu liderler tarafından belirlenen çözüm noktalarına doğru yönlendirilir.

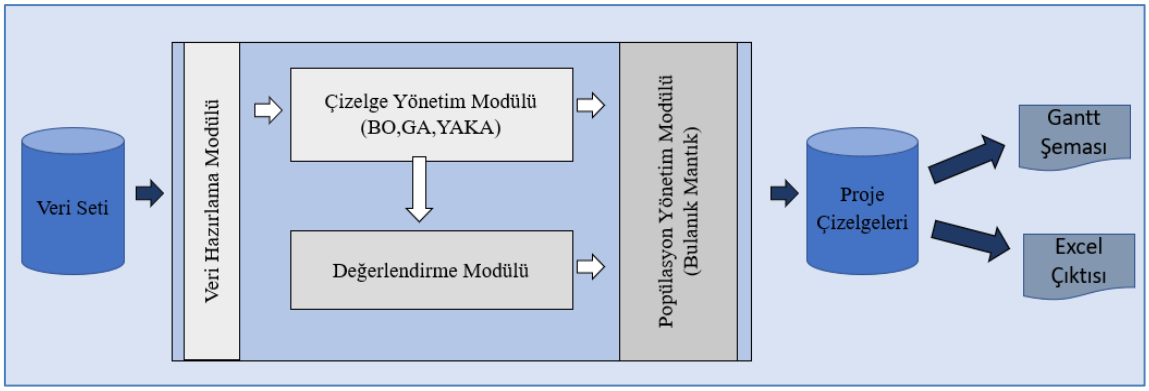
GA'nın entegrasyonu, bu modelin çeşitliliğini artırarak, belirli bir çözüme odaklanma eğilimi olan yerel minimumlara takılma olasılığını düşürür. GA, çaprazlama ve mutasyon işlemleriyle popülasyonun genişletilmesini sağlayarak, çözüm uzayında daha geniş bir arama yapılmasını sağlar.

YAKA, algoritmanın durgunluğunu azaltmak amacıyla entegre edilmiş, özellikle durgun fazlarda popülasyonun yenilenmesi ve çeşitlilik artırılması gibi görevleri üstlenmiştir. YAKA'nın benzersiz arama stratejileri, çözüm uzayının daha geniş kısımlarının taranmasını sağlayarak, genel optimizasyon sürecinin verimliliğini artırmıştır.

Algoritmanın rastgelelik seviyesini iyileştirmek için kaotik lojistik fonksiyon kullanılmıştır. Bu fonksiyon, algoritmanın rastgele sayı üretiminde kullanılarak, standart rastgele sayı üreteçlerine kıyasla daha öngörülemeyen ve karmaşık sayı dizilerinin elde edilmesini

sağlamıştır. Bu sayede algoritma sıradan yaklaşımların ötesinde, durum uzayında daha etkili bir keşif ve daha iyi global optimumlara yakınsama potansiyeli kazanmıştır.

Problemin çözümü için geliştirilen yöntem Şekil 5.1'de belirtildiği üzere dört ana modülden oluşmaktadır.




Şekil 5.1. Yazılım Mimarisi

Veri hazırlama modülü, proje çizelgesi bilgilerini içeren veri setini diğer modüllerin kullanabileceği uygun formata dönüştürür.

Çizelge yönetim modülü, BO hiyerarşisindeki popülasyonu oluşturur ve proje çizelgeleri ile ilişkilendirir. Popülasyondaki bireyler başlangıçta rastgele üretilirken, ilerleyen döngülerde popülasyon yönetim modülünden sağlanan bilgilere göre üretilir. Popülasyon yönetim modülünden gelen bilgiler doğrultusunda GA'daki çaprazlama ve mutasyon işlemleri gerçekleştirilerek yeni bireyler türetilir. Durgunluğun önlenmesi için YAKA'daki popülasyon yenileme süreçleri kullanılarak gerektiğinde bireyler yeniden oluşturulur. Bu modülde bireylerle ilişkilendirilen tüm çizelgeler veri setindeki kriterlere göre üretildiğinden, durum uzayındaki gerçek çözümler kontrol edilip, hatalı çözümlerin elenmesi sağlanır. Bu işlem her ne kadar hesaplama maliyetini arttırsa da kullanışlı olmayan bir proje çizelgesine yakınsama olasılığını sıfıra indirir.

Değerlendirme modülü, üretilen proje çizelgelerinin amaç fonksiyonuna uygun olacak şekilde kalite değerlendirmesini gerçekleştirir.

Popülasyon yönetim modülü, değerlendirme modülünden aldığı bilgiler doğrultusunda, çaprazlama ve mutasyon oranlarını hesaplayıp, seçilimin hangi bireylerde gerçekleştirileceğini belirler.

Her döngü sonunda en iyi kritere sahip olan çizelge global optimuma yakınsama sürecinde kullanılmak üzere saklanır.

Elde edilen optimum proje çizelgeleri, proje yöneticilerinin talep etmesi halinde geliştirilen yazılım üzerinden Gantt şeması veya Microsoft Excel dosyası formatında raporlanabilmektedir. Bu raporlara ilişkin örnekler EK-4 ve EK-6'da verilmektedir.

### 5.1. Yöntemin Bileşenleri ve Temel İşleyişi

Sürü oluşturma ve proje çizelgeleme: Başlangıçta, sürü rastgele olarak oluşturulur ve sürüdeki her bir birey, bir yazılım proje çizelgesini temsil eder. Çizelgeler, veri setindeki koşullara uygun şekilde oluşturulur. Böylelikle, durum uzayında global optimuma yakınsama işlemlerinde, hatalı sonuçlara yakınsama durumu önlenmiş olur.

Değerlendirme ve lider seçimi: Her bir proje çizelgesi, belirlenen amaç fonksiyonlarına göre değerlendirilir. En iyi performans gösteren üç çizelge, sırasıyla alfa, beta ve delta olarak adlandırılan kurtlar tarafından temsil edilir. Bu çizelgeler, sürünün liderleri olarak diğer çizelgeler üzerinde yönlendirici bir etkiye sahiptir. İlk üçe giremeyen çizelgeler omegalar tarafından temsil edilir.

Çaprazlama ve mutasyon: Her döngüde omega kurtları lider kurtlar olan alfa, beta ve delta kurtları ile çaprazlanır. Yöntem olarak iki noktalı çaprazlama işlemi gerçekleştirilir. Bu lider kurtlar çözüm uzayında en iyi konumları işaret ettikleri için, onlarla gerçekleştirilen çaprazlama omega kurtlarının temsil ettiği proje çizelgelerinin kalitesini artırır. Çaprazlama işlemi sonrasında elde edilen yeni omega kurtları daha sonra mutasyona uğratılarak, algoritmanın daha geniş bir çözüm uzayını keşfetmesini sağlar ve yerel minimumlarda takılma riskini azaltır. Omega kurtlarının bu sürekli yeniden üretilme süreci, algoritmanın dinamikliğini korur ve sürekli olarak çözüm kalitesini iyileştirmeyi amaçlar.

Popülasyon yenileme: Algoritmanın durgunluk yaşadığı durumlarda YAKA devreye girer. Belirli bir süre boyunca arama sürecine katkı sağlayamayan sürü üyeleri yeniden üretilir. Bu sayede, algoritma yenilenmiş sürü ile daha dinamik bir arama yapısına kavuşur ve genel optimizasyon sürecinin verimliliği arttırılır.

Kaotik lojistik harita ile rastgelelik: Modelde rastgelelik gerektiren tüm süreçlerde, daha iyi rastgelelik sağlamak amacıyla kaotik lojistik harita kullanılır. Kaotik lojistik harita, proje çizelgelerinin yeniden üretilmesinde kullanılan rastgele sayıların öngörülemeyen ve karmaşık olmasını sağlar, böylece algoritma geniş çözüm uzayında daha etkin aramalar yapabilir duruma gelir.

## 5.2. Hesaplama Karmaşıklığı

Geliştirilen modelin her bir bileşeninin işlem adımlarının ve bu adımların toplam karmaşıklığa etkileri incelenmiştir. $N$ popülasyon boyutu, $p$ projedeki tamamlanması gereken görev sayısı, $q$ insan kaynağı sayısı olmak üzere,

- BO'da sürüde $N$ adet kurt olduğunda hesaplama karmaşıklığı $O = (N)$ olmaktadır [99]. Sürü oluşturma ve başlangıç popülasyonu sürecinde proje çizelgesi oluşturma süreci de dahil edildiğinde, her bir kurt için $pxq$ işlem yapılması gerekir ve bu da sürecin işlem karmaşıklığını $O(N\ x\ p\ x\ q\ )$ yapar.
- Değerlendirme ve lider seçiminde, popülasyondaki en iyi ilk üç kurt belirlenir ve hesaplama karmaşıklığı $O(N)$ olur.
- $i$ GA bileşenindeki iterasyon sayısı, $k = 3(alfa, beta, delta)$, popülasyondaki en iyi değere sahip maksimum kromozom sayısı $N \gg (i, k)$ olmak üzere, algoritmanın hesaplama karmaşıklığı yaklaşık olarak $O = (N)$ olur [100]. Algoritmaya proje çizelgesi oluşturma süreci de dahil edildiği durumda işlem karmaşıklığı $O(N\ x\ p\ x\ q\ )$ olur.
- YAKA ve kaotik lojistik harita süreçleri sürüdeki her bir kurt için işlem yaptığından $O(N)$ karmaşıklığını ortaya çıkarır.

Algoritmanın $m$ iterasyon ile çalıştırılması durumunda, uygulamadaki tüm bileşenlerin işlem karmaşıklığı göz önüne alınırsa, modelin işlem karmaşıklığı (K) Eş. 5.1'de belirtildiği şekilde olur.

$$K = O(N\ x\ p\ x\ q\ ) + \mathrm{O}(N) + O(N\ x\ p\ x\ q\ ) + \mathrm{O}(N\ ) \approx\ O(\mathrm{m\ x}\ N\ x\ p\ x\ q\ ) \qquad (5.1)$$

Bu hesaplama karmaşıklığı, $p$ adet görev ve $q$ adet insan kaynaklı bir projede geliştirilen yöntemin oluşturulan popülasyon ile ideal sonuca ulaşabilmesi için harcayacağı çabaya karşılık gelmektedir.

### 5.3. Veri Setinin Uygulanması

ÇY-KKPÇP'nin çözümü için geliştirilen modelin işleyişinin anlaşılabilmesi için iMOPSE veri setinde yer alan ve 10 adet görev, 3 adet insan kaynağı, 5 adet öncel görev ve 3 adet yetkinlik seviyesinin tanımlı olduğu projeyi temsil eden 10_3_5_3 isimli test verisi kullanılmıştır. Çizelge 5.1 ve Çizelge 5.2, söz konusu proje verisinin geliştirilen model ile işlenmesi sonucunda elde edilen bir adet alfa ve 1 adet omega kurdunun temsil ettiği proje çizelgelerini göstermektedir.

Çizelge 5.1. Alfa için proje çizelgesi (Proje süresi:121 saat)

| Görev | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| İnsan Kaynağı | 0 | 1 | 2 | 2 | 1 | 0 | 2 | 1 | 0 | 0 |

Çizelge 5.2. Omega için proje çizelgesi (Proje süresi:125 saat)

| Görev | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| İnsan Kaynağı | 0 | 1 | 2 | 0 | 2 | 1 | 2 | 1 | 0 | 2 |

Kullanılan yöntemde, alfa, beta ve deltanın avantajlarından yararlanılarak omegaların temsil ettiği proje çizelgelerinin daha iyi seviyelere çekilmesi amaçlanmaktadır. Bu nedenle, omegalar alfa, beta ve delta ile çaprazlanarak yeni omegalar türetilir. Çizelge 5.3'te alfa ve omeganın çaprazlanması sonucu daha iyi bir proje çizelgesinin ortaya çıktığına dair bir örnek yer almaktadır. Alfa ile ilişkili proje çizelgesinin süresi 121, omeganınki ise 125 saattir.

Çaprazlama sonucunda omegadaki 4. ve 5. görevleri gerçekleştiren insan kaynakları alfadaki ilgili görevlerin insan kaynakları ile değiştirilir. Bunun sonucunda da omega ile ilişkili yeni

proje çizelgesinin süresi 121 saate düşer. Daha kısa süreli bir proje çizelgesi elde edildiğinden, ilk omega popülasyondan çıkarılır ve yeni omega popülasyona dahil edilir.

Çizelge 5.3. Omega için proje çizelgesi (Proje süresi:121 saat)

| Görev | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| İnsan Kaynağı | 0 | 1 | 2 | 0 | 1 | 0 | 2 | 1 | 0 | 2 |

Popülasyondaki çeşitliliği arttırmak adına, omegalarda mutasyon işlemi gerçekleştirilir. Mutasyon işlemi, proje çizelgesindeki görevi gerçekleştiren insan kaynağının uygun yetkinlik seviyesindeki farklı bir insan kaynağı ile değiştirilmesi şeklinde gerçekleştirilir.

Çizelge 5.4'te, 10_3_5_3 proje verisi kullanılarak oluşturulan yeni bir proje çizelgesi yer almaktadır. Bu çizelgedeki 167 saatlik proje süresi, ikinci görevde gerçekleştirilen bir mutasyon sonrası, Çizelge 5.5'te belirtildiği şekilde 146 saate düşmüştür. 1 ve 2 nolu görevler artık farklı insan kaynakları tarafından gerçekleştirilmektedir ve bu görevler eş zamanlı olarak yürütülebilmektedir. Bu durum da proje süresinin kısalmasına neden olmaktadır.

Çizelge 5.4. Mutasyon öncesi omega (Proje süresi: 167 saat)

| Görev | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| İnsan Kaynağı | 1 | 1 | 1 | 0 | 2 | 0 | 2 | 1 | 1 | 2 |

Çizelge 5.5. Mutasyon sonrası omega (Proje süresi: 146 saat)

| Görev | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| İnsan Kaynağı | 1 | 1 | 2 | 0 | 2 | 0 | 2 | 1 | 1 | 2 |

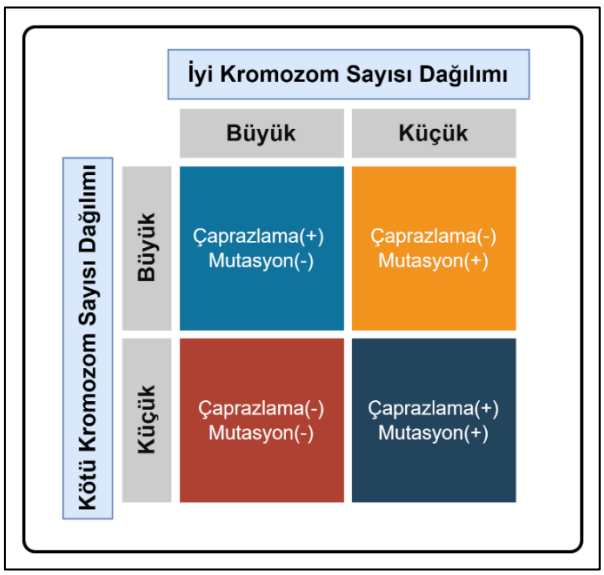
## 5.4. Bulanık Mantık Entegrasyonu

Tez çalışmasında geliştirilen yöntemde kullanılan GA'nın temel fonksiyonlarından olan çaprazlama ve mutasyon işlemlerinin hangi oranlarda gerçekleştirileceği bulanık mantık kullanılarak belirlenmiş ve sonuçlara etkisi analiz edilmiştir. Kullanılan yöntemde,

popülasyondaki iyi ve kötü kromozom sayısına bağlı olarak çaprazlama ve mutasyon oranları ayarlanmaktadır [101].

Şekil 5.2'de de belirtildiği üzere;

- İyi kromozomların sayısı yüksek ve kötü kromozomların sayısı düşükse, mutasyon ve çaprazlama oranları azaltılır.
- Hem iyi hem de kötü kromozomların sayısı yüksekse, mutasyon oranı azaltılır, çaprazlama oranı artırılır.
- İyi kromozomların sayısı düşük ve kötü kromozomların sayısı yüksekse, mutasyon oranı artırılır, çaprazlama oranı azaltılır.
- Hem iyi hem de kötü kromozomların sayısı düşükse, mutasyon ve çaprazlama oranları artırılır.




Şekil 5.2. Mutasyon ve çaprazlama oranlarını belirleme

Mutasyon ve çaprazlama oranlarını belirlemeye ilişkin sözde kod aşağıda yer almaktadır:

Mutasyon ve çaprazlama oranlarını belirleme

Girdi: Popülasyondaki bireyler
Adım 1: Popülasyondaki kaliteli bireyler seçilir.
Adım 2: Seçilen bireylerin uygunluk değerlerinin standart sapması hesaplanır.

Adım 3: Ortalama ve maksimum uygunluk değerleri hesaplanır.

Adım 4: GB: min1 + std1'den daha az uygunluk değerine sahip kromozomların sayısı bulunur.

Adım 5: GW: min1 - std1'den daha fazla uygunluk değerine sahip kromozomların sayısı bulunur.

Adım 6: GB ve GW, toplam kromozom sayısına bölünerek normalleştirilir ve bulanık sisteme giriş olarak verilir.

Çıktı: Çaprazlama ve mutasyon oranları

Elde edilen mutasyon ve çaprazlama oranları ile;

- Türetilen bireyleri popülasyona dahil etme,
- Ebeveynleri popülasyondan çıkarma,
- Ebeveyn mutasyonu ile devam etme

şeklindeki kararlar değerlendirilerek popülasyon çeşitliliği sağlanmaktadır.

### 5.5. Üç Boyutlu Baskın Olmayan Çözüm Kümelerinin Belirlenmesi

Çok amaçlı optimizasyon problemlerinde amaç fonksiyonu sayısı birden fazla olduğundan en iyi sonucu belirleyebilmek için, amaç fonksiyonlarının birlikte değerlendirilmesi gerekmektedir. Bu tür optimizasyon problemlerinde birden fazla parametre en iyi değerine çekilmeye çalışılır. Bu durumda da bir kriter en iyi seviyeye çekilirken, diğer kriter ya da kriterlerin en iyi durumu yakalanamayabilir. Bu nedenle çok amaçlı optimizasyon problemlerinde çözüm uzayı bir vektördür [101]. Bu tür problemlerin çözümünde hedeflenen, karar vericiye önceliklere ve değişen koşullara uygun çözüm kümeleri sunmaktır [102].

Çok amaçlı optimizasyon problemlerinde amaç, baskın olmayan çözümleri bulmaktır. Bu çözüm kümesine Pareto optimal çözüm kümesi denir. Baskın olmayan çözümlerin oluşturduğu eğriye de Pareto önü eğrisi denir [103]. Eş. 5.2'de, amaç fonksiyonunun her $x_1$ değeri, $x_2$ değerinden küçükse ve diğer amaç fonksiyonları için $x_1$ değerinin $x_2$ değerinden küçük olduğu en az bir çözüm varsa, $x_1$ değeri $x_2$ değerine baskın olur.

$$\forall i \in \{1,2,3..,N\}: f_i(x_1) \leq f_i(x_2)$$
$$\exists j \in \{1,2,3..,N\}: f_j(x_1) < f_j(x_2) \quad (5.2)$$

Tez çalışmasında, proje süresi, proje bütçesi ve insan kaynaklarının boşta bekleme sürelerinin optimizasyonunda Ozlen vd. tarafından çok amaçlı tamsayı programlama çalışmaları kapsamında geliştirilen algoritma temel alınmıştır [104-105]. Söz konusu yöntemde, amaç fonksiyonlarının alt ve üst sınırlarının uygun olarak belirlenmesi ve bu sınırlar çerçevesinde üçüncü amaç fonksiyonuna göre diğer iki amaç fonksiyonunun baskın olmayan çözümlerinin elde edilmesi sağlanmaktadır. Bu yaklaşımdan yararlanılarak yazılım proje takvimi oluşturulması probleminde kullanılan üç amaç fonksiyonu için baskın olmayan çözüm kümesi belirlenmiştir.

# 6. DENEYSEL BULGULAR VE DEĞERLENDİRMELER

Tez çalışmasında, gerçek hayattaki projelerle ve tez kapsamında çalışılan problemle uyumlu olması sebebiyle 36 adet farklı boyutta proje verisi içeren iMOPSE veri setinden alınan 10 adet proje verisi kullanılarak sonuçlar analiz edilmiştir. Problemin çözümü için kullanılan yöntemler Çizelge 6.1'de belirtilmiştir.

Çizelge 6.1. Önerilen yöntemler

| Yöntem | Açıklama |
|---|---|
| PGA | GA yöntemi ile proje çizelgesi oluşturma |
| PHibrit | Hibrit yöntem ile proje çizelgesi oluşturma |

PGA yöntemi, temelinde GA olan ve proje çizelgelesi oluşturma optimizasyonunda popülasyon yönetiminin ve çeşitliliğinin GA ile yapıldığı, durgunluğun YAKA ile önlenmeye çalışıldığı ve rastgeleliğin de kaotik lojistik harita ile güçlendirildiği bir yöntemdir.

PHibrit yöntemi ise, popülasyon yönetiminin BO ile yapıldığı, popülasyon çeşitliliğinin GA ile sağlandığı, durgunluğun YAKA ile önlenmeye çalışıldığı ve rastgeleliğin de kaotik lojistik harita ile güçlendirildiği bir yöntemdir.

Tez çalışması kapsamında PHibrit yöntemi ve PHibrit yönteminin önemli bir alt modülü olan PGA yöntemi test edilip, sonuçları analiz edilmiştir. Her iki yöntemde kullanılan çaprazlama ve mutasyon oranlarının bulanık mantık ile belirlenmesinin sonuçlara katkısı da ayrıca incelenmiştir.

## 6.1. Yöntemlerin Değerlendirilmesi

Tez çalışması kapsamında, Visual Studio geliştirme ortamında, .NET platformu üzerinde C# dili kullanılarak ve MS SQL Server veri tabanı entegreli bir yazılım geliştirilmiştir. Geliştirilen yazılım, tüm yöntemlerin parametrelerinin kullanıcı tarafından ayarlanabildiği

parametrik bir yapıya sahip olmakla birlikte, deneysel çalışmalar için kullanılan parametreler Çizelge 6.2'de belirtilmektedir.

Çizelge 6.2. Deneysel çalışmalarda kullanılan parametreler

| Parametre | Değer |
|---|---|
| Popülasyon Boyutu | Veri setinde yer alan insan kaynağı sayısına eşittir. |
| BO hiyerarşisi | Yönetim kadrosunu oluşturan alfa, beta ve delta kurtları her biri bir adettir. |
| GA oranları | Mutasyon ve çaprazlama oranı, görev sayısının %20'sini geçmeyecek şekilde rastgele belirlenmektedir. |
| YAKA durgunluk parametresi | 50 olarak alınmıştır. |
| Lojistik harita kontrol parametresi | 3,7801 olarak alınmıştır. |
| Bulanık mantık | Katsayı belirlenmesinde popülasyonun en iyi %10 bireyi kullanılır.<br>Eşik değeri 0,25 alınmıştır. |

PGA ve PHibrit yöntemleri ile 100_20_46_15 proje verisi kullanılarak elde edilen sonuçlar Çizelge 6.3 ve Çizelge 6.4'te gösterilmektedir. Proje süresi ve insan kaynaklarının boşta bekleme sürelerinin optimizasyonunda PHibrit yönteminin, bütçe optimizasyonunda ise PGA yönteminin daha iyi sonuçlar ürettiği görülmüştür.

Çizelge 6.3. PGA* kullanarak proje süresi, proje bütçesi ve boşta bekleme süresinin (GBekle) karşılaştırılması

| Süre Optimizasyonu | | | Bütçe Optimizasyonu | | | GBekle Optimizasyonu | | |
|---|---|---|---|---|---|---|---|---|
| Süre | Bütçe | GBekle | Süre | Bütçe | GBekle | Süre | Bütçe | GBekle |
| 164 | 146865 | 5621 | 598 | 71491,2 | 21383 | 164 | 146605,8 | 5269 |
| 165 | 148824,9 | 5691 | 627 | 71576,7 | 20516 | 172 | 148319,9 | 5278 |
| 166 | 144743,7 | 6006 | 667 | 71576,7 | 20601 | 181 | 145317,1 | 5304 |
| 168 | 147439,7 | 5694 | 610 | 71714,7 | 20245 | 171 | 149642,8 | 5312 |
| 171 | 150411,2 | 5702 | 596 | 71727 | 20422 | 178 | 144552,4 | 5320 |
| 171 | 144919 | 5666 | 596 | 72006,1 | 20723 | 194 | 146636,9 | 5333 |

Çizelge 6.3. (devam) PGA* kullanarak proje süresi, proje bütçesi ve boşta bekleme süresinin (GBekle) karşılaştırılması

| Süre Optimizasyonu | | | Bütçe Optimizasyonu | | | GBekle Optimizasyonu | | |
|---|---|---|---|---|---|---|---|---|
| Süre | Bütçe | GBekle | Süre | Bütçe | GBekle | Süre | Bütçe | GBekle |
| 172 | 143792,2 | 5649 | 599 | 72006,1 | 20279 | 200 | 142517,5 | 5341 |
| 172 | 149548,5 | 6021 | 693 | 72006,1 | 21327 | 217 | 146609,4 | 5345 |
| 173 | 152379,1 | 6006 | 631 | 72144,1 | 20143 | 179 | 146660,5 | 5359 |
| 173 | 148320,6 | 5960 | 663 | 72240,1 | 20972 | 190 | 145427,9 | 5364 |

* 10 ayrı çalıştırma

Çizelge 6.4. PHibrit* kullanarak proje süresi, proje bütçesi ve görev bekleme süresinin (GBekle) karşılaştırılması

| Süre Optimizasyonu | | | Bütçe Optimizasyonu | | | GBekle Optimizasyonu | | |
|---|---|---|---|---|---|---|---|---|
| Süre | Bütçe | GBekle | Süre | Bütçe | GBekle | Süre | Bütçe | GBekle |
| 164 | 142172,7 | 5572 | 595 | 79174,8 | 18140 | 204 | 143402,1 | 5228 |
| 164 | 143281,4 | 5773 | 526 | 79574,8 | 15876 | 204 | 139202,9 | 5237 |
| 165 | 145825,3 | 5758 | 522 | 79652,8 | 16879 | 192 | 142046,3 | 5268 |
| 168 | 141859,7 | 5916 | 594 | 79871,5 | 15954 | 181 | 144252,4 | 5275 |
| 168 | 140119 | 5615 | 486 | 80149,9 | 16889 | 184 | 141086 | 5282 |
| 168 | 141201 | 5432 | 521 | 80646,4 | 17497 | 192 | 144039 | 5286 |
| 168 | 145508,5 | 5704 | 445 | 81063,9 | 14115 | 187 | 143031,3 | 5292 |
| 170 | 147548,8 | 5636 | 454 | 81129,5 | 14816 | 194 | 140458,4 | 5310 |
| 171 | 142572,7 | 5896 | 521 | 81378,6 | 16807 | 193 | 143628,6 | 5343 |
| 171 | 144088,7 | 5734 | 635 | 81386,4 | 18536 | 186 | 143046,5 | 5359 |

* 10 ayrı çalıştırma

Çizelge 6.3 ve Çizelge 6.4 incelendiğinde, proje süresi optimizasyonunda PGA ve PHibrit yöntemlerinin aynı sonuca yakınsadığı, bütçe optimizasyonunda PHibrit yönteminin yerel minimuma takıldığı, insan kaynaklarının boşta bekleme sürelerinin optimizasyonunda ise, PHibrit yönteminin daha düşük sonuçlar ürettiği görülmektedir.

## 6.2. Bulanık Mantık Entegrasyonu Sonucu Yöntemlerin Değerlendirilmesi

Çizelge 6.1'de belirtilen yöntemlerde kullanılan çaprazlama ve mutasyon oranlarının bulanık mantık ile belirlenmesi durumunda, bu değişikliğin sonuçlara nasıl etki edeceği aynı veri seti kullanılarak (100_20_46_15 proje verisi) analiz edilmiştir. Bulanık mantık entegre edilen yöntemlerin isimlendirmesi Çizelge 6.5'te yer almaktadır.

Çizelge 6.5. Bulanık mantık kullanılan yöntemler

| Yöntem | Açıklama |
|---|---|
| PGA-B | PGA yöntemine bulanık mantık süreçleri eklenmiştir. |
| PHibrit-B | PHibrit yöntemine bulanık mantık süreçleri eklenmiştir. |

Çizelge 6.6 ve Çizelge 6.7'de yer alan sonuçlar incelendiğinde, PHibrit-B yönteminin bütçe ve görevlerin boşta bekleme sürelerinin optimizasyonunda daha etkin sonuçlar ürettiği, PGA-B yönteminin proje tamamlanma süresi optimizasyonunda PHibrit-B'ye oranla daha başarılı olduğu, PHibrit-B yönteminin proje süresi optimizasyonunun ise yerel minimuma takıldığı görülmektedir.

Mutasyon ve çaprazlama oranlarının optimizasyon süreçlerine etkisini değerlendirmek için Çizelge 6.6 ve Çizelge 6.3'te yer alan veriler karşılaştırıldığında, proje süresi, bütçe ve boşta bekleme sürelerinin optimizasyonu sonucunda PGA-B yönteminin PGA yöntemine göre daha etkin sonuçlar ürettiği görülmektedir. Çizelge 6.7 ve Çizelge 6.4'te yer alan veriler karşılaştırıldığında ise proje süresi, bütçe ve boşta bekleme süresi optimizasyonunda PHibrit-B yönteminin PHibrit yöntemine göre daha etkin sonuçlar ürettiği görülmektedir.

Çizelge 6.6. PGA-B* kullanarak proje süresi, proje bütçesi ve görev bekleme süresinin (GBekle) karşılaştırılması

| Süre Optimizasyonu | | | Bütçe Optimizasyonu | | | GBekle Optimizasyonu | | |
|---|---|---|---|---|---|---|---|---|
| Süre | Bütçe | GBekle | Süre | Bütçe | GBekle | Süre | Bütçe | GBekle |
| 158 | 145463,5 | 5599 | 617 | 71491,2 | 20234 | 165 | 145330 | 5168 |
| 158 | 149705,1 | 5447 | 635 | 71491,2 | 21490 | 164 | 148101,3 | 5187 |
| 158 | 148092,6 | 5566 | 600 | 71491,2 | 20874 | 176 | 148490,8 | 5196 |

Çizelge 6.6. (devam) PGA-B* kullanarak proje süresi, proje bütçesi ve görev bekleme süresinin (GBekle) karşılaştırılması

| Süre Optimizasyonu | | | Bütçe Optimizasyonu | | | GBekle Optimizasyonu | | |
|---|---|---|---|---|---|---|---|---|
| Süre | Bütçe | GBekle | Süre | Bütçe | GBekle | Süre | Bütçe | GBekle |
| 160 | 145249,3 | 5676 | 657 | 71491,2 | 21023 | 168 | 145302,2 | 5198 |
| 161 | 148678,8 | 5607 | 596 | 71491,2 | 20818 | 165 | 146354,9 | 5199 |
| 161 | 147387,1 | 5781 | 601 | 72006,1 | 20412 | 180 | 147852,1 | 5201 |
| 161 | 148357,1 | 5439 | 688 | 72006,1 | 20987 | 163 | 147394 | 5242 |
| 162 | 148744,3 | 5769 | 653 | 72215,7 | 20466 | 176 | 144290,7 | 5248 |
| 162 | 149357,1 | 5746 | 640 | 72435,7 | 21287 | 181 | 147001,4 | 5272 |
| 162 | 149058,8 | 5611 | 620 | 72838,9 | 20892 | 184 | 147307,1 | 5274 |

* 10 ayrı çalıştırma

Çizelge 6.7. PHibrit-B * kullanarak proje süresi, proje bütçesi ve görev bekleme süresinin (GBekle) karşılaştırılması

| Süre Optimizasyonu | | | Bütçe Optimizasyonu | | | GBekle Optimizasyonu | | |
|---|---|---|---|---|---|---|---|---|
| Süre | Bütçe | GBekle | Süre | Bütçe | GBekle | Süre | Bütçe | GBekle |
| 161 | 142736,6 | 5848 | 601 | 70146,6 | 19988 | 172 | 143978,5 | 5013 |
| 161 | 141844,1 | 5716 | 660 | 70576 | 20551 | 175 | 143751,4 | 5022 |
| 161 | 143865,7 | 5718 | 601 | 70690 | 20480 | 175 | 141092,5 | 5025 |
| 161 | 147223,6 | 5775 | 601 | 70769,8 | 20085 | 165 | 144844,5 | 5032 |
| 161 | 141328,1 | 5580 | 601 | 70769,8 | 20954 | 193 | 140902,1 | 5083 |
| 161 | 143679,8 | 5875 | 626 | 70811,8 | 20397 | 184 | 145537,8 | 5085 |
| 161 | 144155,2 | 5723 | 601 | 71215 | 19020 | 180 | 144498,1 | 5097 |
| 161 | 143965,2 | 5768 | 601 | 71229,4 | 20215 | 172 | 144598,8 | 5108 |
| 161 | 145325,4 | 5658 | 604 | 71229,4 | 19822 | 205 | 142243,1 | 5142 |
| 162 | 143438,4 | 5571 | 604 | 71483,2 | 19806 | 176 | 142326,1 | 5171 |

* 10 ayrı çalıştırma

## 6.3. İstatistiksel Değerlendirme

ÇY-KKPÇP'nin çözümünde kullanılan amaç fonksiyonları saat ve para birimi olmak üzere iki farklı ölçüm birimine sahiptir. Elde edilen sonuçların farklı birimlerle ölçülüyor olması,

standart sapma kullanılarak yapılan değerlendirmelerde sonuçların doğrudan karşılaştırılmasını zorlaştırmaktadır.

Varyasyon katsayısı, standart sapmanın aritmetik ortalamaya oranını ifade eder ve bu değer genellikle yüzde olarak ifade edilir. Bu katsayının kullanımı, farklı birimlerde ölçülen sonuçların birbiriyle karşılaştırılmasını mümkün kılar. Böylelikle sonuçların dağılımının ne kadar geniş olduğu ölçüm birimlerinden bağımsız olarak ele alınabilir. Bu sayede de projenin farklı yönleri değerlendirilirken elde edilen sonuçların tutarlılığı ve değişkenliği daha objektif bir şekilde değerlendirilebilir [106].

Düşük bir varyasyon katsayısı, geliştirilen yöntemin çeşitli durumlarda benzer sonuçlar verdiğini ve dolayısıyla daha öngörülebilir ve güvenilir olduğunu gösterir. Yüksek bir varyasyon katsayısı ise, yöntemin farklı durumlarda farklı sonuçlar verebileceğini ve bu durumun potansiyel olarak daha fazla risk taşıdığını gösterir [107]. Bu bilgi, proje yöneticilerine ve karar vericilere, hangi yöntemin projelerinin ihtiyaçlarına daha iyi uyum sağladığını seçmelerinde yardımcı olur.

100_20_46_15 proje verisinin kullanıldığı testler sonucunda elde edilen varyasyon katsayıları Çizelge 6.8'de belirtilmiştir. Katsayılar değerlendirildiğinde, PHibrit-B yönteminin proje süresi optimizasyonunda, PGA yönteminin ise bütçe ve boşta bekleme süresi optimizasyonunda diğer yöntemlere göre daha tutarlı sonuçlar ürettiği görülmektedir.

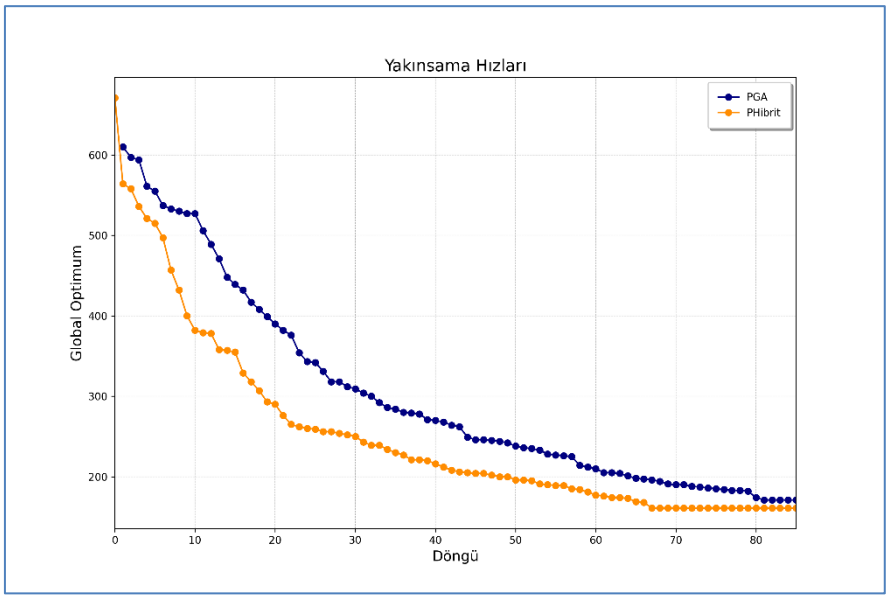
Çizelge 6.8. Yöntemlerin istatistiksel sonuçları

| | | Ortalama | Standart Sapma | Varyasyon Katsayısı |
|---|---|---|---|---|
| Süre | PGA | 169,5 | 3,26 | 1,92 |
| | PHibrit | 167,7 | 2,49 | 1,48 |
| | PGA-B | 160,30 | 1,61 | 1,00 |
| | PHibrit-B | 161,10 | 0,30 | 0,18 |
| Bütçe | PGA | 71848,88 | 249,66 | 0,35 |
| | PHibrit | 80402,9 | 777,58 | 0,97 |
| | PGA-B | 71895,85 | 461,09 | 0,64 |
| | PHibrit-B | 70892,10 | 376,04 | 0,53 |
| GBekle | PGA | 5322,5 | 30,52 | 0,57 |
| | PHibrit | 5244,4 | 41,54 | 0,79 |
| | PGA-B | 5218,50 | 35,38 | 0,67 |
| | PHibrit-B | 5077,80 | 51,34 | 1,01 |

## 6.4. Yöntemlerin Performans Değerlendirmesi

Geliştirilen yöntemlerin global optimuma yakınsama hızları 100_20_46_15 proje verisi kullanılarak ve proje tamamlanma süresi optimize edilerek analiz edilmiştir.

Şekil 6.1 PGA ve PHibrit yöntemlerinin döngüler boyunca global optimum değerine olan yakınsama davranışlarını göstermektedir. Şekil incelendiğinde PHibrit yönteminin döngüler ilerledikçe PGA yöntemine oranla daha hızlı bir şekilde global optimuma yaklaştığı görülmektedir. Bu durum, PHibrit yönteminin yerel minimumlardan kaçınma kabiliyetinin PGA'ya göre daha iyi olduğunu ortaya koymaktadır.

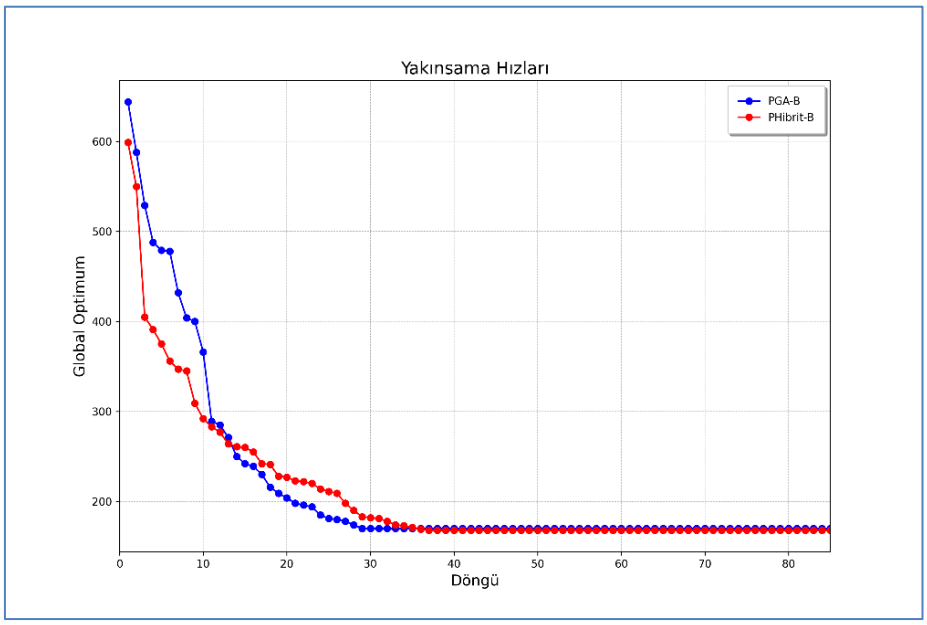



Şekil 6.1. PGA ve PHibrit yöntemlerinin yakınsama hızı karşılaştırması

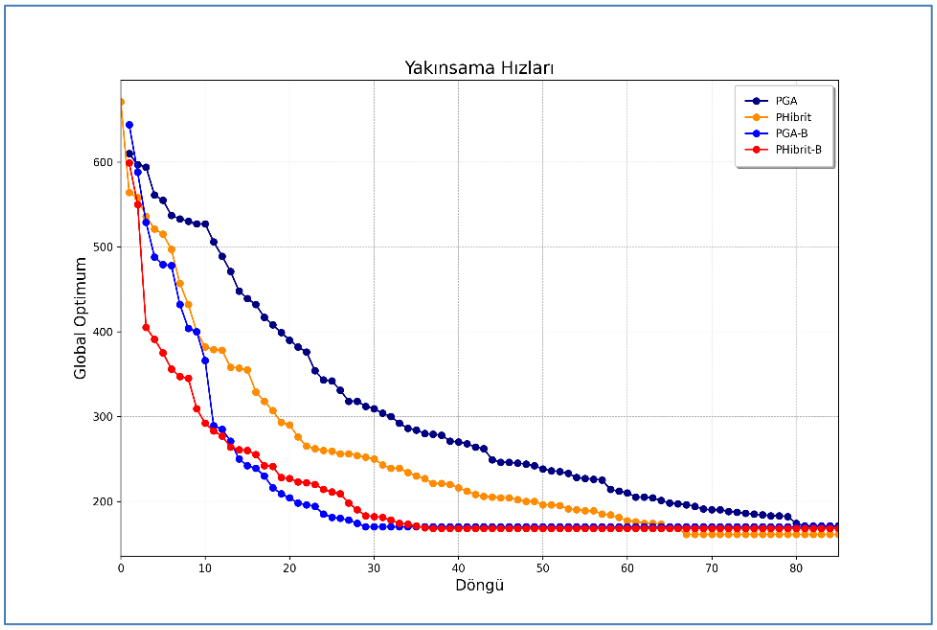
Benzer şekilde, Şekil 6.2'de yer alan PGA-B ve PHibrit-B yöntemlerinin yakınsama davranışları incelendiğinde, PHibrit-B yönteminin PGA-B yöntemine göre yerel minimumlardan kaçınma konusunda daha kabiliyetli olduğu görülmektedir.

Yöntemlerin yakınsama hızlarının bir arada gösterildiği Şekil 6.3'te PHibrit ve PHibrit-B algoritmalarının optimum değerlere daha düşük seviyelerde ulaşarak ve yerel minimumlardan daha iyi kaçınarak etkili optimizasyon davranışları sergiledikleri, PGA yönteminin yakınsama hızının diğer yöntemlere daha yavaş olduğu, PGA-B yönteminin ise hızlı yakınsama sonucunda yerel minimuma takıldığı görülmektedir.

Sonuç olarak, uzun vadeli ve kararlı çözüm aranan problemler için PHibrit ve PHibrit-B yöntemleri, kısa vadeli ve hızlı çözüm aranan problemler için bulanık mantık entegreli PGA-B ve PHibrit-B yöntemleri çözüme ulaşma konusunda daha başarılı sonuçlar üretmektedir.




Şekil 6.2. PGA-B ve PHibrit-B yöntemlerinin yakınsama hızı karşılaştırması




Şekil 6.3. Yöntemlerin yakınsama hızı karşılaştırması

## 6.5. Üç Boyutlu Baskın Olmayan Çözüm Kümeleri

Tez çalışmasında proje süresi, proje bütçesi ve insan kaynaklarının boşta bekleme süresi amaç fonksiyonları kullanılarak yapılan optimizasyon işlemleri sonrasında, elde edilen çözüm kümelerinin üç boyutlu baskın olmayan çözümleri elde edilmiştir. Elde edilen çözüm kümeleri ile proje yöneticisine farklı senaryolar içerisinden dengeli seçim yapabilme olanağı getirilmiştir. Örneğin bir çözüm seti, proje süresini minimize ederken bütçe ve insan kaynaklarının boşta bekleme süresini de optimize eden seçenekler sunabilir. Bu sayede, proje yöneticileri, bütçeyi aşmadan ve insan kaynaklarını verimli kullanarak projeyi en kısa sürede tamamlayacak stratejileri belirleyebilirler.

Dört farklı yöntem her amaç fonksiyonunun optimizasyonu için 100_20_46_15 proje verisi kullanılarak ayrı ayrı çalıştırılmış ve elde edilen üç boyutlu baskın olmayan sonuç kümeleri Çizelge 6.9, Çizelge 6.10, Çizelge 6.11 ve Çizelge 6.12'de belirtilmiştir.

Çizelge 6.9. PGA kullanarak elde edilen baskın olmayan çözümler

| Sıra | Süre | Bütçe | GBekle | Sıra | Süre | Bütçe | GBekle |
|---|---|---|---|---|---|---|---|
| 1 | 466 | 87266,7 | 15182 | 14 | 182 | 141008,9 | 5891 |
| 2 | 188 | 140402,5 | 5982 | 15 | 626 | 71784,4 | 19808 |
| 3 | 434 | 108576,5 | 12883 | 16 | 614 | 71576,7 | 20193 |
| 4 | 239 | 137532,8 | 6866 | 17 | 270 | 140315 | 6609 |
| 5 | 409 | 97692 | 12957 | 18 | 346 | 131360,6 | 10557 |
| 6 | 487 | 79303,6 | 16379 | 19 | 225 | 135373,7 | 7317 |
| 7 | 592 | 74246,7 | 18680 | 20 | 514 | 77966,5 | 17487 |
| 8 | 305 | 138277,5 | 6803 | 21 | 555 | 74939,5 | 19167 |
| 9 | 165 | 147707,6 | 5732 | 22 | 582 | 73211,4 | 19398 |
| 10 | 596 | 71694,8 | 20744 | 23 | 303 | 133767,1 | 8256 |
| 11 | 521 | 76163,1 | 17143 | 24 | 218 | 145247,9 | 5680 |
| 12 | 605 | 72975,9 | 19648 | 25 | 178 | 146586,5 | 5342 |
| 13 | 211 | 145396,5 | 5620 | 26 | 178 | 147677,3 | 5286 |

Çizelge 6.10. PHibrit kullanarak elde edilen baskın olmayan çözümler

| Sıra | Süre | Bütçe | GBekle | Sıra | Süre | Bütçe | GBekle |
|---|---|---|---|---|---|---|---|
| 1 | 491 | 102097,6 | 12910 | 18 | 488 | 127561,7 | 13460 |
| 2 | 584 | 70648,5 | 19838 | 19 | 387 | 132183,6 | 8742 |
| 3 | 204 | 141111,9 | 5334 | 20 | 204 | 141662,8 | 5268 |
| 4 | 199 | 139237,1 | 6605 | 21 | 519 | 87683 | 15692 |
| 5 | 164 | 142063,3 | 5741 | 22 | 480 | 74972 | 17142 |
| 6 | 487 | 78027,2 | 16644 | 23 | 505 | 74351,9 | 17422 |
| 7 | 505 | 73907,8 | 17547 | 24 | 601 | 70458,1 | 20325 |
| 8 | 204 | 142141 | 5228 | 25 | 204 | 142734,8 | 5203 |
| 9 | 205 | 135411 | 6318 | 26 | 481 | 94419,6 | 13867 |
| 10 | 465 | 79865,6 | 16232 | 27 | 499 | 76480,6 | 16796 |
| 11 | 487 | 79199 | 16355 | 28 | 622 | 70061,1 | 20578 |
| 12 | 318 | 133723,2 | 7412 | 29 | 366 | 133613,6 | 8274 |
| 13 | 303 | 135396,8 | 6898 | 30 | 241 | 139178,5 | 5809 |
| 14 | 519 | 92915,3 | 14751 | 31 | 517 | 72814,7 | 18104 |
| 15 | 435 | 88779,1 | 15602 | 32 | 584 | 71018,9 | 18854 |
| 16 | 472 | 79992 | 15968 | 33 | 590 | 70726,5 | 18942 |
| 17 | 517 | 73511,9 | 17645 | 34 | 193 | 146071,3 | 5572 |

Çizelge 6.11. PGA-B kullanarak elde edilen baskın olmayan çözümler

| Sıra | Süre | Bütçe | GBekle | Sıra | Süre | Bütçe | GBekle |
|---|---|---|---|---|---|---|---|
| 1 | 521 | 83362 | 15961 | 17 | 347 | 131710,5 | 9767 |
| 2 | 611 | 72006,1 | 20702 | 18 | 509 | 93350,5 | 14157 |
| 3 | 180 | 143740,2 | 5532 | 19 | 483 | 88178,1 | 15604 |
| 4 | 211 | 141290,2 | 6720 | 20 | 571 | 73397,5 | 18649 |
| 5 | 662 | 72180,1 | 19701 | 21 | 637 | 72715,7 | 20265 |
| 6 | 648 | 72112,1 | 19755 | 22 | 222 | 140699,3 | 6187 |
| 7 | 371 | 129714,9 | 10372 | 23 | 160 | 147970,8 | 5236 |
| 8 | 323 | 138971 | 7260 | 24 | 495 | 91146,9 | 14885 |
| 9 | 575 | 81135,9 | 17582 | 25 | 514 | 83937,9 | 16407 |
| 10 | 478 | 90943,8 | 15305 | 26 | 515 | 81580,6 | 16798 |
| 11 | 600 | 79242,9 | 17920 | 27 | 515 | 81537,2 | 16950 |

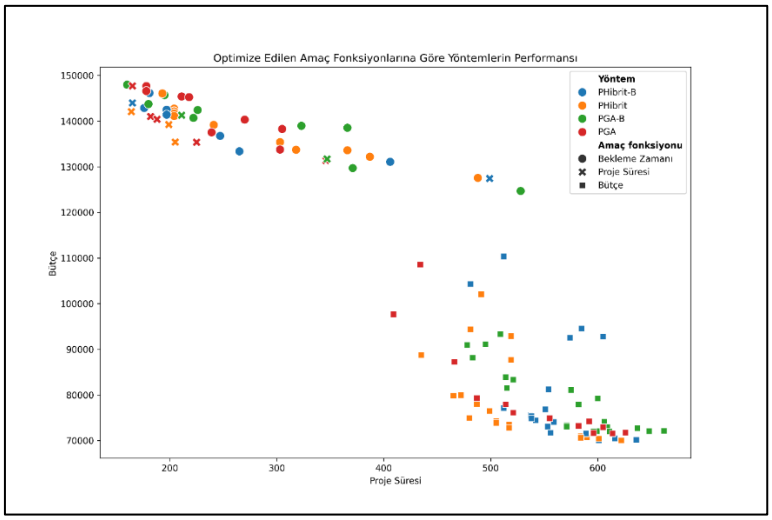
Çizelge 6.11. (devam) PGA-B kullanarak elde edilen baskın olmayan çözümler

| Sıra | Süre | Bütçe | GBekle | Sıra | Süre | Bütçe | GBekle |
|---|---|---|---|---|---|---|---|
| 12 | 582 | 77940,8 | 18073 | 28 | 609 | 72991,5 | 19126 |
| 13 | 606 | 74172,4 | 17845 | 29 | 599 | 72112,1 | 20437 |
| 14 | 571 | 73087,3 | 19010 | 30 | 528 | 124687 | 12792 |
| 15 | 596 | 72011,3 | 20494 | 31 | 226 | 142427,5 | 5973 |
| 16 | 366 | 138547,1 | 8705 | 32 | 195 | 145703,6 | 5381 |

Çizelge 6.12. PHibrit-B kullanarak elde edilen baskın olmayan çözümler

| Sıra | Süre | Bütçe | GBekle | Sıra | Süre | Bütçe | GBekle |
|---|---|---|---|---|---|---|---|
| 1 | 585 | 94573,8 | 15405 | 15 | 538 | 74888,4 | 18623 |
| 2 | 554 | 81230,9 | 16911 | 16 | 556 | 71746,4 | 19192 |
| 3 | 265 | 133380,3 | 7077 | 17 | 406 | 131078,9 | 9813 |
| 4 | 512 | 110353,7 | 12232 | 18 | 481 | 104302,7 | 13206 |
| 5 | 574 | 92524,1 | 15833 | 19 | 551 | 76876,7 | 17219 |
| 6 | 537 | 75583,1 | 17580 | 20 | 542 | 74454,4 | 18476 |
| 7 | 589 | 71576 | 19531 | 21 | 197 | 142496,6 | 5377 |
| 8 | 636 | 70207,5 | 19877 | 22 | 181 | 146153,3 | 5120 |
| 9 | 197 | 141411,7 | 5637 | 23 | 165 | 143977 | 5843 |
| 10 | 553 | 73098,5 | 18880 | 24 | 512 | 77156,3 | 16999 |
| 11 | 601 | 70061,1 | 19988 | 25 | 605 | 92822,8 | 14759 |
| 12 | 176 | 142881,1 | 5227 | 26 | 559 | 74105,1 | 18631 |
| 13 | 499 | 127422,8 | 13161 | 27 | 616 | 70521,5 | 19454 |
| 14 | 538 | 75399,1 | 18016 | 28 | 247 | 136762 | 6091 |

Baskın olmayan çözümlerin bir arada bulunduğu Şekil 6.4 incelendiğinde, proje yöneticilerinin karar verme süreçlerinde belirli durumlarda hangi optimizasyon yöntemini tercih edebileceklerine dair bir sonuca ulaşılabilmektedir: Proje süresi optimizasyonunda PGA ve PHibrit yöntemleri daha dengeli sonuçlar üretirken, bütçe optimizasyonu için PHibrit ve PHibrit-B yöntemlerinin daha avantajlı olduğu görülmektedir.




Şekil 6.4. Yöntemlerin performansı

## 6.6. Literatür Karşılaştırması

Çizelge 6.13'te ÇY-KKPÇP çözümünde aynı veri setini kullanan literatürdeki yöntemlerin elde ettiği sonuçlar yer almaktadır [22,31,72,108]. Literatürde problem sadece proje tamamlanma süresi optimize edilecek şekilde çalışıldığından, tez çalışması kapsamında geliştirilen modellerin bu yöntemlerle karşılaştırılması, proje tamamlanma süresi kapsamında yapılmıştır.

Çizelge 6.13. Literatürdeki yöntemlerin sonuçları

| | Literatürdeki Yöntemler | | | | |
|---|---|---|---|---|---|
| Veri Seti | GreedyDO | HAntCO | GRASP | CSM | DEM |
| 100_5_22_15 | 630 | 504 | 503 | 488 | 485 |
| 100_10_26_15 | 370 | 266 | 251 | 247 | 244 |
| 100_10_47_9 | 549 | 297 | 263 | 268 | 270 |
| 100_20_46_15 | 394 | 194 | 170 | 188 | 181 |
| 100_20_65_9 | 408 | 180 | 135 | 174 | ND |
| 200_10_50_15 | 763 | 529 | 524 | 500 | 496 |
| 200_20_54_15 | 488 | 336 | 304 | 329 | 301 |
| 200_20_55_9 | 999 | 313 | 257 | 304 | ND |
| 200_40_91_15 | 519 | 207 | 153 | 197 | ND |
| 200_40_133_15 | 512 | 214 | 163 | ND | 205 |

Geliştirilen yöntemlerin sonuçları Çizelge 6.14'te sunulmaktadır. Sonuçlar değerlendirildiğinde, geliştirilen yöntemlerin literatürdeki sonuçlardan daha başarılı olduğu, en başarılı sonuçların da PHibrit-B modeli ile elde edildiği görülmektedir. Bu sonuçlar, PHibrit-B modelinin, çeşitlilik sağlayarak arama uzayında daha etkin bir arama gerçekleştirdiğini ve yerel minimumlardan kaçınarak daha iyi çözümlere ulaşabildiğini göstermektedir. PGA-B yönteminin durum uzayının daha küçük olduğu problemlerde çözümde başarılı olduğu görülmektedir.

Çizelge 6.14. Proje sürelerinin literatürle karşılaştırılması

| Veri seti | LMD* | PGA | PGA-B | PHibrit | PHibrit-B |
|---|---|---|---|---|---|
| 100_5_22_15 | 485 | 485 | **480** | 485 | 484 |
| 100_10_26_15 | 244 | 240 | **232** | 238 | 234 |
| 100_10_47_9 | 263 | 260 | 257 | 259 | **256** |
| 100_20_46_15 | 170 | 164 | **158** | 164 | 161 |
| 100_20_65_9 | 135 | 143 | 139 | **135** | **135** |
| 200_10_50_15 | 496 | 492 | 487 | 487 | **486** |
| 200_20_54_15 | 301 | 280 | 275 | 287 | **262** |
| 200_20_55_9 | 257 | 253 | **251** | 256 | 254 |
| 200_40_91_15 | 153 | 165 | 150 | 153 | **144** |
| 200_40_133_15 | 163 | 174 | 161 | 165 | **159** |

* LMD: GreedyDO, HAntCO, GRASP, CSM ve DEM yöntemlerinin minimumları.

# 7. SONUÇLAR

Bu çalışmada, proje süresi, proje bütçesi ve insan kaynaklarının görevlere atanırken oluşabilecek boşta bekleme süresini azaltmak amacıyla ÇY-KKPÇP'nin çözümü için hibrit yöntemler önerilmiştir.

PGA yönteminde GA ve YAKA, PGA-B yönteminde GA, YAKA ve bulanık mantık, PHibrit yönteminde BO, genetik algoritma ve YAKA, PHibrit-B yönteminde ise BO, GA, YAKA ve bulanık mantık kullanılmıştır. Yöntemlerin tümünde olası durgunluğu önlemek için YAKA'nın kâşif arı stratejisinden yararlanılmıştır. Kaotik lojistik haritanın rastgele sayı üreteci olarak kullanımı ile algoritmaların rastgeleliği arttırılmıştır. Proje yöneticilerinin elde edilen proje takvimi sonuçlarından mevcut şartlar altında hangilerinin kullanılabileceği kararını verebilmeleri için optimize edilen amaç fonksiyonlarının baskın olmayan çözüm kümeleri belirlenmiştir.

Tez çalışması kapsamında geliştirilen 4 farklı yöntem, gerçek proje verilerine uygun şekilde tasarlanmış olan iMOPSE veri setinden seçilen 10 adet proje verisi kullanılarak test edilmiş ve sonuçları analiz edilmiştir. Her bir yöntemin üç farklı amaç fonksiyonunun optimizasyonunda elde ettiği sonuçlar karşılaştırılmış olup, yöntemlerin global optimum değerlerine yakınsama davranışları incelenmiştir.

Deneysel çalışmalar, geliştirilen yöntemlerin yazılım proje çizelgeleme problemlerine uygulanabilirliğini ortaya koyarak, hangi yöntemin hangi koşullarda etkili sonuçlar ürettiğini göstermiştir.

Yapılan çalışma ile yazılım projelerinde proje başarısına önemli katkı sağlayan proje çizelgesi oluşturma işleminin manuel yöntemlerle değil de daha hızlı ve etkin sonuçlar ortaya koyan optimizasyon yöntemleri ile gerçekleştirilmesi sağlanmıştır. Böylelikle proje yöneticilerinin gerçekleştireceği çizelgeleme sürecinde insan kaynağı faktöründen kaynaklanacak hataların minimize edilmesi sağlanmış, optimum sonuçlar üretilerek şirketlerin zaman ve maliyet tasarrufları arttırılmıştır. Ayrıca, geliştirilen optimizasyon algoritmaları sayesinde proje çizelgesi daha gerçekçi ve uygulanabilir hale getirilerek

projelerin zamanında tamamlanma olasılığı arttırılmış böylelikle proje bütçesinin daha yönetilebilir olması sağlanmıştır.

Geliştirilen yöntemler gerçek proje verilerine yakın bilgiler içeren iMOPSE veri seti kullanılarak test edilmiş ve yöntemlerin gerçek hayatta etkin sonuçlar üretecek şekilde uygulanabilir olduğu ortaya konulmuştur.

Ortaya konan çalışma ile proje yöneticilerine, şirketin içinde bulunduğu şartlar ve beklentiler dikkate alınarak uygun yazılım proje çizelgesi oluşturma imkânı ve ortamı sağlanmıştır.

Olası senaryolara uygun olarak tercih edilebilecek yöntemler aşağıda önerilmektedir:

- Dengeli sonuçların elde edilmek istendiği projelerde proje tamamlanma süresini optimize etmek için PHibrit-B yöntemi, proje bütçesi ve insan kaynaklarının boşta bekleme sürelerini optimize etmek için de PGA yöntemi kullanılır.
- Hızlı çözüm gerektiren durumlarda PGA-B ve PHibrit-B yöntemleri kullanılır.
- Optimum sonuçlar istendiğinde PHibrit ve PHibrit-B yöntemleri uygulanır.
- Amaç fonksiyonlarının bütünü ekseninde dengeli kararlar verilmesi gereken durumlarda proje süresi optimizasyonunda PGA ve PHibrit, proje bütçesi ve insan kaynaklarının boşta bekleme süresi optimizasyonunda PHibrit ve PHibrit yöntemleri tercih edilir.

Bu çerçevede oluşturulacak verimli proje çizelgeleri sayesinde, literatüre katkı sunulması ve ilerleyen dönemlerdeki çalışmalar için yol gösterici olması amaçlanmıştır.

Tez çalışması sürecinde karşılaşılan ve literatüre katkı anlamında ilerleyen dönemde yapılacak çalışmalar için geliştirilen öneriler şu şekildedir:

- Yapılacak çalışmalarda kullanılabilmesi için yazılım projeleri saha çalışmalarından elde edilen veri setlerine ihtiyaç duyulmaktadır. Özellikle optimizasyon algoritmalarının test edilmesinde kullanılabilecek veri setlerinin hazırlanması ve bilim dünyasında paylaşılması önerilmektedir.
- Proje sürecinde ortaya çıkabilecek değişikliklere rağmen, başlangıçta belirlenen hedeflerden sapma oranı düşük seviyede tutulmalıdır. Yapılacak optimizasyon

çalışmalarıyla, proje çizelgeleme süreçlerinin olası değişikliklere dinamik olarak uyum sağlayarak beslenmesi ve proje maliyetine yönelik analizlerin yapılması önerilmektedir.

- Durum uzayının çok daha büyük olduğu projelerde etkin sonuçlar elde edilebilmesi için yöntemin paralel süreçler ve güncel optimizasyon algoritmaları ile desteklenmesi önerilmektedir.
- İnsan kaynaklarının yetkinlik seviyelerinin görev sürelerine etkilerini sürece dahil eden optimizasyon çalışmaları gerçekleştirilmelidir.
- Bu kapsamda çalışmaların devam ettirilmesi ve yöntemlerin sadece yazılım proje çizelgeleme kapsamında değil, genel proje çizelgeleme problemlerine de uygulanabilirliğinin araştırılması önerilmektedir.

# KAYNAKLAR


1. Boehm, B. W. (1984). Software Engineering Economics. *IEEE Transactions on Software Engineering*, SE-10(1), 4–21.

2. İnternet: Standish Group. *Chaos Report 1994*. URL: https://web.archive.org/web/20240509084507/https://www.standishgroup.com/sample_research_files/chaos_report_1994.pdf, Son Erişim Tarihi: 09.05.2024.

3. İnternet: Portman, H. *Review Standish Group – CHAOS 2020: Beyond Infinity | Henny Portman's Blog*. (n.d.). URL: https://web.archive.org/web/20240509090632/https://hennyportman.wordpress.com/2021/01/06/review-standish-group-chaos-2020-beyond-infinity/, Son Erişim Tarihi: 09.05.2024.

4. Pohl, K., ve Rupp, C. (2015). *Requirements engineering fundamentals: A study guide for the certified professional for requirements engineering exam, foundation level, IREB compliant* (Second Edition). ABD: Rocky Nook,3.

5. Sommerville, I. (2016). *Software engineering* (Tenth Edition). İngiltere: Pearson, 73-76.

6. Pressman, R. S. (2010). *Software engineering: A practitioner's approach* (Seventh Edition). ABD: McGraw-Hill, 119-123.

7. İnternet: *MS-RCPSP iMOPSE datasets*. URL: https://web.archive.org/web/20240509112928/http://imopse.ii.pwr.wroc.pl/download.html, Son Erişim Tarihi: 09.05.2024.

8. Chang, C. K., Christensen, M. J., ve Zhang, T. (2001). Genetic Algorithms for Project Management. *Annals of Software Engineering*, 11, 107–139.

9. Chang, C. K., Jiang, H., Di, Y., Zhu, D., ve Ge, Y. (2008). Time-line based model for software project scheduling with genetic algorithms. *Information and Software Technology*, 50(11), 1142–1154.

10. Akbari, R., Zeighami, V., ve Ziarati, K. (2011). Artificial Bee colony for resource constrained project scheduling problem. *International Journal of Industrial Engineering Computations*, 2, 45–60.

11. Stylianou, C., ve Andreou, A. S. (2011). *Intelligent Software Project Scheduling and Team Staffing with Genetic Algorithms. In Iliadis, L., Maglogiannis, I., Papadopoulos, H. (eds) Artificial Intelligence Applications and Innovations.* IFIP Advances in Information and Communication Technology, 169–178. Springer, Berlin, Heidelberg.

12. Nayak, V., A. Suthar, H., ve Gadit, J. (2012). Implementation of Artificial Bee Colony Algorithm. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 1(3), 112-120.

13. Xiao, J., Ao, X.-T., ve Tang, Y. (2013). Solving software project scheduling problems with ant colony optimization. *Computers and Operations Research*, 40(1), 33-46.

14. Chen, W.-N., ve Zhang, J. (2013). Ant Colony Optimization for Software Project Scheduling and Staffing with an Event-Based Scheduler. *IEEE Transactions on Software Engineering*, 39(1), 1-17.

15. Suri, B., ve Jajoria, P. (2013). *Using ant colony optimization in software development project scheduling*. 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2101–2106, Mysore, India.

16. Bansal, J. C., Sharma, H., ve Jadon, S. S. (2013). Artificial bee colony algorithm: A survey. *International Journal of Advanced Intelligence Paradigms*, *5*(1/2), 123-159.

17. Gharehchopogh, F. S., ve Dizaji, Z. A. (2014). A New Approach in Software Cost Estimation with Hybrid of Bee Colony and Chaos Optimizations Algorithms. *MAGNT Research Report*, 2(6), 1263–1271.

18. Luna, F., González-Álvarez, D. L., Chicano, F., ve Vega-Rodríguez, M. A. (2014). The software project scheduling problem: A scalability analysis of multi-objective metaheuristics. *Applied Soft Computing*, 15, 136–148.

19. Mirjalili, S., Mirjalili, S. M., ve Lewis, A. (2014). Grey Wolf Optimizer. *Advances in Engineering Software*, 69, 46–61.

20. Mohd Pauzi, N. F. (2015). *Flowshop scheduling using artificial bee colony (ABC) algorithm with varying onlooker bees approaches* [Masters, Universiti Tun Hussein Onn Malaysia].

21. Seo, D., Shin, D., ve Bae, D. (2015). *Quality Based Software Project Staffing and Scheduling with Cost Bound*. 2015 Asia-Pacific Software Engineering Conference (APSEC), 269–276, New Delhi, India.

22. Myszkowski, P. B., Skowroński, M., ve Sikora, K. (2015). *A new benchmark dataset for Multi-Skill Resource-Constrained Project Scheduling Problem*. 2015 Federated Conference on Computer Science and Information Systems (FedCSIS), 129–138, Lodz, Poland.

23. Myszkowski, P. B., Skowroński, M. E., Olech, Ł. P., Oślizło, K. (2015). Hybrid ant colony optimization in solving multi-skill resource-constrained project scheduling problem. *Soft Computing*, 19(12), 3599–3619.

24. Mirjalili, S., Saremi, S., Mirjalili, S. M., Coelho, L. dos S. (2016). Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization. *Expert Systems with Applications*, 47, 106–119.

25. Pauzi, N. F. B. M., Bareduan, S. A. (2016). Scheduling analysis for flowship using artifical bee colony (ABC) algorithm with varying onlooker approaches. *ARPN Journal of Engineering and Applied Sciences*, 11(10), 6472–6477.

26. Mathew, J., Paul, B., Dileeplal, J., ve Mathew, T. (2016). Multi Objective Optimization for Scheduling Repetitive Projects Using GA. *Procedia Technology*, 25, 1072–1079.

27. Long, W., Xu, S. (2016). *A novel grey wolf optimizer for global optimization problems.* 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 1266–1270, Xi'an, China.

28. Hameed, M., Khalid, H., Qamar, U., Abass, S. K. (2017). *Optimizing software project management staffing and work-force deployment processes using swarm intelligence*. 2017 Computing Conference, 78–84, London, UK.

29. Gaidhane, P. J., Nigam, M. J. (2018). A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems. *Journal of Computational Science*, 27, 284–302.

30. Gai, W., Qu, C., Liu, J., Zhang, J. (2018). *An improved grey wolf algorithm for global optimization*. 2018 Chinese Control And Decision Conference (CCDC), 2494-2498, Shenyang, China.

31. Myszkowski, P. B., Olech, Ł. P., Laszczyk, M., Skowroński, M. E. (2018). Hybrid Differential Evolution and Greedy Algorithm (DEGR) for solving Multi-Skill Resource-Constrained Project Scheduling Problem. *Applied Soft Computing*, 62, 1–14.

32. Bibiks, K., Hu, Y., Li, J., Pillai, P., ve Smith, A. (2018). Improved discrete cuckoo search for the resource-constrained project scheduling problem. *Applied Soft Computing*, 69, 493–503.

33. Afshar-Nadjafi, B. (2018). A solution procedure for preemptive multi-mode project scheduling problem with mode changeability to resumption. *Applied Computing and Informatics*, 14(2), 192-201.

34. Nemati-Lafmejani, R., Davari-Ardakani, H., Najafzad, H. (2019). Multi-mode resource constrained project scheduling and contractor selection: Mathematical formulation and metaheuristic algorithms. *Applied Soft Computing*, 81, 1-22.

35. Guo, M. W., Wang, J. S., Zhu, L. F., Guo, S. S., Xie, W. (2020). An Improved Grey Wolf Optimizer Based on Tracking and Seeking Modes to Solve Function Optimization Problems. *IEEE Access*, 8, 69861–69893.

36. Lu, C., Gao, L., Li, X., Hu, C., Yan, X., Gong, W. (2020). Chaotic-based grey wolf optimizer for numerical and engineering optimization problems. *Memetic Computing*, 12, 371-398.

37. Tian, Y., Xiong, T., Liu, Z., Deng, P., Wan, L. (2020). *Novel Feedback-based Operators in Solving Multi-skill Resource-Constrained Project Scheduling Problem*. 2020 Chinese Control And Decision Conference (CCDC), 296–301, Hefei, China.

38. Rauf, M., Guan, Z., Yue, L., Guo, Z., Mumtaz, J., Ullah, S. (2020). Integrated Planning and Scheduling of Multiple Manufacturing Projects Under Resource Constraints Using Raccoon Family Optimization Algorithm. *IEEE Access*, 8, 151279–151295.

39. Dang Quoc, H., Nguyen The, L., Nguyen Doan, C., Xiong, N. (2020). Effective Evolutionary Algorithm for Solving the Real-Resource-Constrained Scheduling Problem. *Journal of Advanced Transportation*, 2020, 1–11.

40. Xie, F., Li, H., Xu, Z. (2021). Multi-mode resource-constrained project scheduling with uncertain activity cost. *Expert Systems with Applications*, 168, 1-12.

41. Snauwaert, J., Vanhoucke, M. (2021). A new algorithm for resource-constrained project scheduling with breadth and depth of skills. *European Journal of Operational Research*, 292(1), 43–59.

42. Ghamginzadeh, A., Najafi, A. A., Khalilzadeh, M. (2021). Multi-Objective Multi-Skill Resource-Constrained Project Scheduling Problem Under Time Uncertainty. *International Journal of Fuzzy Systems*, 23(2), 518–534.

43. Cai, J., Peng, Z., Liao, S., Ding, S. (2022). A multi-mode multi-skill project scheduling reformulation for reconnaissance mission planning. *Science China Information Sciences*, 65(6), 1-2.

44. Wang, M., Liu, G., Lin, X. (2022). Dynamic Optimization of the Multi-Skilled Resource-Constrained Project Scheduling Problem with Uncertainty in Resource Availability. *Mathematics*, 10, 1-20.

45. Mahmud, F., Sarker, R., ve Essam, D. (2022). *Multi-Operator based Genetic Algorithm for Resource Constrained Project Scheduling*. 2022 14th International Conference on Software, Knowledge, Information Management and Applications (SKIMA), 292–297, Phnom Penh, Cambodia.

46. Akbar, S., Ahmad, I., Khan, R., Lopes, I. O., Ullah, R. (2022). Multi-Skills Resource Constrained and Personality Traits Based Project Scheduling. *IEEE Access*, 10, 131419–131429.

47. Ou, C.-H., Li, Y.-H., Chen, C.-Y., Wu, C.-H., Tsai, Y.-C., Yan, Z.-Y., Chang, C.-R. (2023). *Quantum-Inspired Optimization for Task Scheduling in Software Development Projects*. 2023 IEEE International Conference on Quantum Computing and Engineering (QCE), 348–349, Bellevue, WA, USA.

48. Nigar, N., Shahzad, M. K., Islam, S., Oki, O., ve Lukose, J. M. (2023). Multi-Objective Dynamic Software Project Scheduling: A Novel Approach to Handle Employee's Addition. *IEEE Access*, 11, 39792–39806.

49. Nigar, N., Shahzad, M. K., Islam, S., Oki, O., ve Lukose, J. M. (2023). A Novel Multi-Objective Evolutionary Algorithm to Address Turnover in the Software Project Scheduling Problem Based on Best Fit Skills Criterion. *IEEE Access*, 11, 89742–89756.

50. Akbar, S., Zubair, M., Khan, R., Ul Akbar, U., Ullah, R., ve Zheng, Z. (2024). Weighted Multi-Skill Resource Constrained Project Scheduling: A Greedy and Parallel Scheduling Approach. *IEEE Access*, 12, 29824–29836.

51. Kerzner, H. (2009). *Project management: A systems approach to planning, scheduling, and controlling* (Tenth Edition). ABD:Wiley, 19.

52. Schwalbe, K. (2016). *Information Technology Project Management* (Eighth Edition). Cengage Learning, 234-238.

53. Kelley, J. E. (1961). Critical-Path Planning and Scheduling: Mathematical Basis. *Operations Research*, 9(3), 296–320.

54. O. Bellenguez-Morineau ve E. Néron. (2007). A Branch-and-Bound method for solving Multi-Skill Project Scheduling Problem. *RAIRO-Operation Resources*, 41(2), 155–170.

55. Pellerin, R., Perrier, N., ve Berthaut, F. (2020). A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 280(2), 395–416.

56. Blazewicz, J., Lenstra, J. K., ve Kan, A. H. G. R. (1983). Scheduling subject to resource constraints: Classification and complexity. *Discrete Applied Mathematics*, 5(1), 11–24.

57. Chen, T., ve Zhou, G. (2013). Research on Project Scheduling Problem with Resource Constraints. *Journal of Software*, 8(8), 2058–2063.

58. Mahmud, F., Zaman, F., Ahrari, A., Sarker, R., ve Essam, D. (2021). Genetic Algorithm for Singular Resource Constrained Project Scheduling Problems. *IEEE Access*, 9, 131767–131779.

59. Chen, R., Liang, C., Gu, D., ve Leung, J. Y.-T. (2017). A multi-objective model for multi-project scheduling and multi-skilled staff assignment for IT product development considering competency evolution. *International Journal of Production Research*, 55(21), 6207–6234.

60. Neron, E. (2002). *Lower Bounds for the Multi-Skill Project Scheduling Problem*. Proceeding of the Eighth Inter-national Workshop on Project Management and Scheduling, 274–277, Spain, 2002.

61. Myszkowski, P. B., Skowronski, M. E., ve Podlodowski, Ł. (2013). *Novel heuristic solutions for Multi-Skill Resource-Constrained Project Scheduling Problem*. Proceedings of the 2013 Federated Conference on Computer Science and Information Systems, 159-166, Krakow, Poland.

62. Hartmann, S., ve Briskorn, D. (2010). A survey of variants and extensions of the resource-constrained project scheduling problem. *European Journal of Operational Research*, 207(1), 1–14.

63. Zhang, H., Li, H., ve Tam, C. M. (2006). Particle swarm optimization for resource-constrained project scheduling. *International Journal of Project Management*, 24(1), 83–92.

64. Uysal, F. (2014). *Hybrid meta-heuristic algorithms for the resource constrained multi-project scheduling problem*, Doctoral Thesis, Civil Engineering of Middle East Technical University, Türkiye.

65. Tian, Y., Xiong, T., Liu, Z., Mei, Y., ve Wan, L. (2022). Multi-Objective multi-skill resource-constrained project scheduling problem with skill switches: Model and evolutionary approaches. *Computers and Industrial Engineering*, 167, 1-20.

66. Habibi, F., Barzinpour, F., ve Sadjadi, S. J. (2018). Resource-constrained project scheduling problem: Review of past and recent developments. *Journal of Project Management*, 3, 55–88.

67. Schwindt, C., ve Zimmermann, J. (2015). *Handbook on Project Management and Scheduling Vol.1* (First Edition). Springer International Publishing : Imprint: Springer, xxv-xxix.

68. Project Management Institute (Ed.). (2013). *A guide to the project management body of knowledge (PMBOK guide)* (Fifth edition). Project Management Institute, 279-284.

69. Santos, M. A., Tereso, A. P. (2011). On the Multi-mode, Multi-skill Resource Constrained Project Scheduling Problem – A Software Application. In A. Gaspar-Cunha, R. Takahashi, G. Schaefer, and L. Costa (Eds.). *Soft Computing in Industrial Applications*, 96, 239–248.

70. Kolisch, R., Sprecher, A. (1997). PSPLIB - A project scheduling problem library. *European Journal of Operational Research*, 96(1), 205–216.

*71.* Brucker, P., Drexl, A., Mo, R., Pesch, E. (1999). Resource-constrained project scheduling: Notation, classification, models, and methods. *European Journal of Operational Research*, 112(1), 3-41.

72. Myszkowski, P. B., Laszczyk, M., Nikulin, I., ve Skowroński, M. (2019). iMOPSE: A library for bicriteria optimization in Multi-Skill Resource-Constrained Project Scheduling Problem. *Soft Computing*, 23(10), 3397–3410.

73. Jaberi, M., ve Jaberi, M. (2014). A Multi-objective Resource-constrained Project-scheduling Problem Using Mean Field Annealing Neural Networks. *Journal of Mathematics and Computer Science*, 09(03), 228–239.

74. Kaya, S., ve Fi, N. (2016). Çok Amaçlı Optimizasyon Problemlerinde Pareto Optimal Kullanımı. *Social Sciences Research Journal*, 5(2), 9–18.

75. Fahmy, A. M. (2016). Optimization Algorithms in Project Scheduling. *Optimization Algorithms:Methods and Applications*. InTechOpen. 171-193.

76. Meziane, F., Vadera, S. (Eds.). (2010). *Artificial Intelligence Applications for Improved Software Engineering Development: New Prospects*. USA: Information Science Reference, 278-299.

77. Nocedal, J., Wright, S. J. (1999). *Numerical optimization*. USA: Springer, 2-9.

78. Yavuz, N. (2006). *Kaotik ortamlarda güvenli veri transferi*, Yüksek Lisans Tezi, Bilgisayar Mühendisliği Ana Bilim Dalı, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü, Türkiye.

79. Lu, H., Wang, X., Fei, Z., Qiu, M. (2014). The Effects of Using Chaotic Map on Improving the Performance of Multiobjective Evolutionary Algorithms. *Mathematical Problems in Engineering*, 2014, 1–16.

80. Tian, D. (2015). Particle Swarm Optimization with Chaotic Maps and Gaussian Mutation for Function Optimization. *International Journal of Grid and Distributed Computing*, 8(4), 123-134.

81. Yan, G., Li, C. (2011). An Effective Refinement Artificial Bee Colony Optimization Algorithm Based On Chaotic Search and Application for PID Control Tuning. *Journal of Computational Information Systems*, 9(2011), 3309-3316.

82. Tai, H. K., Jusoh, S. A., ve Siu, S. W. I. (2018). Chaos-embedded particle swarm optimization approach for protein-ligand docking and virtual screening. *Journal of Cheminformatics*, 10(1), 62.

83. Ozoren, A. R. (2011). *Random Number Generation Using Chaotic Dynamical Maps,* Master of Science Thesis, Industrial Engineering of Boğaziçi University, Türkiye.

84. Wu, S., Wan, H.-D., Shukla, S. K., ve Li, B. (2011). Chaos-based improved immune algorithm (CBIIA) for resource-constrained project scheduling problems. *Expert Systems with Applications*, 38, 3387–3395.

85. May, R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, 261(5560), 459–467.

86. Alligood, K. T., Sauer, T. D., ve Yorke, J. A. (1996). *Chaos: An introduction to dynamical systems* (Repr.). USA:Springer, 17-22.

87. François, M., Defour, D., ve Negre, C. (2014). A Fast Chaos-Based Pseudo-Random Bit Generator Using Binary64 Floating-Point Arithmetic. *Informatica*, 38, 115-124.

88. Schaffer, J. G. (1985). *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms*. Proceedings of the 1st International Conference on Genetic Algorithms, 93–100, New Jersey, USA.

89. Goldberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley Publishing Company, 2-17.

90. Mitchell, M. (1996). *An introduction to genetic algorithms* (First Edition). MIT Press, 128-130.

91. Srinivas, M., ve Patnaik, L. M. (1994). Genetic algorithms: A survey. *Computer*, 27(6), 17–26.

92. Gargiulo, F., ve Quagliarella, D. (2012). *Genetic Algorithms for the Resource Constrained Project Scheduling Problem*. 2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI), 39–47, Budapest, Hungary.

93. Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization. *Technical Report*, 10.

94. Karaboga, D. (2017). *Yapay Zeka Optimizasyon Algoritmaları* (Dördüncü Baskı). Nobel Akademik Yayıncılık, 201-210.

95. Karaboga, D., Gorkemli, B., Ozturk, C., ve Karaboga, N. (2014). A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artificial Intelligence Review*, 42(1), 21–57.

96. Karaboga, D., ve Akay, B. (2009). A comparative study of Artificial Bee Colony algorithm. *Applied Mathematics and Computation*, 214(1), 108–132.

97. Karaboga, D., ve Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459–471.

98. Pian, J., Wang, G., ve Li, B. (2018). An Improved ABC Algorithm Based on Initial Population and Neighborhood Search. *IFAC-PapersOnLine*, 51(18), 251–256.

99. Tu, Q., Chen, X., Liu, X. (2019). Hierarchy Strengthened Grey Wolf Optimizer for Numerical Optimization and Feature Selection. *IEEE Access*, 7, 78012–78028.

100. Zhou, X., Miao, F., Ma, H. (2018). Genetic Algorithm with an Improved Initial Population Technique for Automatic Clustering of Low-Dimensional Data. *Information*, 9(4), 101.

101. Javadi, G., Aminian, E. (2017). *A New Method for Tuning Mutation and Crossover Rate in Genetic Algorithm.* Proceedings of the 9th International Conference on Machine Learning and Computing, 217–220, Singapore.

102. Ngatchou, P., Zarei, A., El-Sharkawi, A. (2005). *Pareto Multi Objective Optimization.* Proceedings of the 13th International Conference on, Intelligent Systems Application to Power Systems, 84–91, Virginia, USA.

103. Wang, R. (2016). An Improved Nondominated Sorting Genetic Algorithm for Multiobjective Problem. *Mathematical Problems in Engineering*, 2016, 1–7.

104. Özlen, M., Azizoğlu, M. (2009). Multi-objective integer programming: A general approach for generating all non-dominated solutions. *European Journal of Operational Research*, 199, 25-35.

105. Ozlen, M., Burton, B. A., MacRae, C. A. G. (2014). Multi-Objective Integer Programming: An Improved Recursive Algorithm. *Journal of Optimization Theory and Applications*, 160, 470-482.

106. Orhunbilge, N. (2000). *Tanımsal istatistik olasılık ve olasılık dağılımları*. İstanbul Üniversitesi, 119-120.

107. Frank, H., ve Althoen, S. C. (1994). *Statistics: Concepts and applications* (First Edition). USA: Cambridge University Press, 52-59.

108. 108. Quoc, H. D., The, L. N., Doan, C. N., ve Thanh, T. P. (2020). *New Effective Differential Evolution Algorithm for the Project Scheduling Problem.* 2020 2nd International Conference on Computer Communication and the Internet (ICCCI), 150–157, Nagoya, Japan.

# EKLER

EK-1. Proje süresi optimizasyonu (100_20_46_14 proje verisi)

Çizelge 1.1. PHibrit-B yöntemi

| Görev No | IK No | Yetkinlik | Yetkinlik Seviyesi | Proje Süresi | Proje Bütçesi | Boşta Bekleme Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 72 | 0 | 13 | 2 | 37,0 | 1372,7 | 0,0 | 0 |
| 6 | 10 | 0 | 2 | 10,0 | 2148,7 | 0,0 | 1 |
| 88 | 4 | 8 | 2 | 25,0 | 3156,2 | 0,0 | 2 |
| 36 | 0 | 13 | 2 | 77,0 | 4640,2 | 37,0 | 3 |
| 42 | 10 | 0 | 2 | 24,0 | 5726,6 | 47,0 | 4 |
| 1 | 7 | 3 | 0 | 37,0 | 9067,7 | 47,0 | 5 |
| 45 | 12 | 14 | 2 | 27,0 | 11152,1 | 47,0 | 6 |
| 77 | 2 | 5 | 2 | 31,0 | 13241,5 | 47,0 | 7 |
| 24 | 18 | 13 | 2 | 40,0 | 16177,5 | 47,0 | 8 |
| 25 | 2 | 8 | 1 | 65,0 | 18469,1 | 78,0 | 9 |
| 85 | 15 | 9 | 1 | 21,0 | 19418,3 | 78,0 | 10 |
| 50 | 6 | 10 | 0 | 38,0 | 22598,9 | 78,0 | 11 |
| 52 | 2 | 5 | 2 | 84,0 | 23879,5 | 143,0 | 12 |
| 17 | 5 | 14 | 2 | 17,0 | 24959,0 | 143,0 | 13 |
| 5 | 9 | 0 | 1 | 19,0 | 26511,3 | 143,0 | 14 |
| 37 | 12 | 14 | 2 | 67,0 | 29599,3 | 170,0 | 15 |
| 99 | 4 | 0 | 2 | 43,0 | 30324,7 | 195,0 | 16 |
| 74 | 14 | 8 | 2 | 40,0 | 32116,7 | 195,0 | 17 |
| 93 | 13 | 3 | 2 | 38,0 | 32815,9 | 195,0 | 18 |
| 57 | 11 | 5 | 2 | 33,0 | 33792,7 | 195,0 | 19 |
| 71 | 12 | 8 | 0 | 94,0 | 35877,1 | 262,0 | 20 |
| 8 | 4 | 4 | 2 | 62,0 | 36642,8 | 305,0 | 21 |
| 90 | 16 | 12 | 1 | 40,0 | 37490,8 | 305,0 | 22 |
| 21 | 1 | 1 | 2 | 27,0 | 38225,2 | 305,0 | 23 |
| 31 | 10 | 0 | 2 | 57,0 | 40786,0 | 329,0 | 24 |
| 20 | 9 | 0 | 1 | 57,0 | 43890,6 | 348,0 | 25 |
| 2 | 0 | 7 | 2 | 112,0 | 45189,1 | 425,0 | 26 |
| 38 | 1 | 12 | 2 | 40,0 | 45542,7 | 452,0 | 27 |
| 49 | 3 | 11 | 2 | 29,0 | 45832,7 | 452,0 | 28 |
| 44 | 17 | 2 | 0 | 37,0 | 48655,8 | 452,0 | 29 |
| 19 | 11 | 3 | 1 | 71,0 | 49780,6 | 485,0 | 30 |
| 12 | 6 | 2 | 2 | 63,0 | 51873,1 | 523,0 | 31 |
| 75 | 8 | 4 | 1 | 26,0 | 53503,3 | 523,0 | 32 |
| 26 | 19 | 3 | 1 | 36,0 | 55810,9 | 523,0 | 33 |
| 33 | 18 | 13 | 2 | 60,0 | 57278,9 | 563,0 | 34 |
| 55 | 10 | 4 | 2 | 88,0 | 59684,5 | 620,0 | 35 |
| 3 | 4 | 8 | 2 | 98,0 | 61135,3 | 682,0 | 36 |
| 40 | 2 | 5 | 2 | 106,0 | 62618,1 | 766,0 | 37 |
| 23 | 17 | 3 | 1 | 57,0 | 64144,1 | 803,0 | 38 |
| 43 | 19 | 6 | 2 | 55,0 | 65362,0 | 839,0 | 39 |

EK-1. (devam) Proje süresi optimizasyonu (100_20_46_14 proje verisi)

Çizelge 1.1.(devam) PHibrit-B yöntemi

| Görev No | IK No | Yetkinlik | Yetkinlik Seviyesi | Proje Süresi | Proje Bütçesi | Boşta Bekleme Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 22 | 1 | 12 | 2 | 53,0 | 65715,6 | 879,0 | 40 |
| 28 | 7 | 2 | 2 | 50,0 | 66889,5 | 916,0 | 41 |
| 15 | 19 | 11 | 0 | 86,0 | 68876,6 | 971,0 | 42 |
| 16 | 5 | 7 | 1 | 56,0 | 71353,1 | 988,0 | 43 |
| 10 | 15 | 10 | 0 | 39,0 | 72166,7 | 1009,0 | 44 |
| 0 | 17 | 8 | 1 | 76,0 | 73616,4 | 1066,0 | 45 |
| 62 | 4 | 9 | 1 | 113,0 | 74220,9 | 1164,0 | 46 |
| 39 | 9 | 10 | 2 | 82,0 | 76263,4 | 1221,0 | 47 |
| 66 | 15 | 10 | 0 | 74,0 | 77845,4 | 1260,0 | 48 |
| 76 | 9 | 6 | 0 | 114,0 | 80459,8 | 1342,0 | 49 |
| 92 | 7 | 1 | 2 | 63,0 | 81633,7 | 1392,0 | 50 |
| 78 | 8 | 6 | 2 | 57,0 | 83577,4 | 1418,0 | 51 |
| 30 | 1 | 0 | 1 | 65,0 | 83903,8 | 1471,0 | 52 |
| 34 | 3 | 11 | 2 | 41,0 | 84023,8 | 1500,0 | 53 |
| 9 | 9 | 0 | 1 | 142,0 | 86311,4 | 1614,0 | 54 |
| 98 | 10 | 9 | 1 | 111,0 | 88096,2 | 1702,0 | 55 |
| 87 | 4 | 0 | 2 | 134,0 | 88942,5 | 1815,0 | 56 |
| 81 | 5 | 14 | 2 | 92,0 | 91228,5 | 1871,0 | 57 |
| 48 | 3 | 11 | 2 | 55,0 | 91368,5 | 1912,0 | 58 |
| 79 | 17 | 13 | 1 | 91,0 | 92513,0 | 1988,0 | 59 |
| 27 | 14 | 6 | 2 | 49,0 | 92916,2 | 2028,0 | 60 |
| 18 | 16 | 10 | 1 | 73,0 | 93615,8 | 2068,0 | 61 |
| 4 | 2 | 5 | 2 | 115,0 | 94222,4 | 2174,0 | 62 |
| 11 | 13 | 3 | 2 | 75,0 | 94903,2 | 2212,0 | 63 |
| 7 | 0 | 7 | 2 | 143,0 | 96053,3 | 2324,0 | 64 |
| 14 | 6 | 8 | 1 | 102,0 | 99317,6 | 2387,0 | 65 |
| 86 | 7 | 11 | 2 | 94,0 | 102116,9 | 2450,0 | 66 |
| 47 | 18 | 13 | 2 | 98,0 | 104759,3 | 2512,0 | 67 |
| 46 | 7 | 1 | 2 | 109,0 | 106113,8 | 2606,0 | 68 |
| 68 | 8 | 5 | 0 | 118,0 | 108120,2 | 2692,0 | 69 |
| 53 | 1 | 12 | 2 | 86,0 | 108691,4 | 2757,0 | 70 |
| 35 | 14 | 8 | 2 | 83,0 | 110214,6 | 2806,0 | 71 |
| 94 | 14 | 6 | 2 | 97,0 | 110841,8 | 2889,0 | 72 |
| 61 | 14 | 8 | 2 | 128,0 | 112230,6 | 2986,0 | 73 |
| 69 | 12 | 6 | 0 | 149,0 | 113851,8 | 3114,0 | 74 |
| 63 | 3 | 1 | 1 | 70,0 | 114001,8 | 3169,0 | 75 |
| 29 | 11 | 7 | 1 | 120,0 | 115008,2 | 3255,0 | 76 |
| 73 | 6 | 2 | 2 | 135,0 | 117770,3 | 3357,0 | 77 |
| 56 | 15 | 3 | 1 | 107,0 | 118855,1 | 3440,0 | 78 |
| 80 | 13 | 3 | 2 | 111,0 | 119517,5 | 3515,0 | 79 |

EK-1. (devam) Proje süresi optimizasyonu (100_20_46_14 proje verisi)

Çizelge 1.1.(devam) PHibrit-B yöntemi

| Görev No | IK No | Yetkinlik | Yetkinlik Seviyesi | Proje Süresi | Proje Bütçesi | Boşta Bekleme Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 32 | 7 | 11 | 2 | 123,0 | 120781,7 | 3624,0 | 80 |
| 96 | 13 | 8 | 0 | 146,0 | 121425,7 | 3735,0 | 81 |
| 41 | 8 | 6 | 2 | 157,0 | 123871,0 | 3853,0 | 82 |
| 58 | 18 | 13 | 2 | 130,0 | 126219,8 | 3951,0 | 83 |
| 89 | 10 | 8 | 1 | 139,0 | 128392,6 | 4062,0 | 84 |
| 13 | 0 | 7 | 2 | 165,0 | 129208,8 | 4205,0 | 85 |
| 54 | 2 | 3 | 1 | 133,0 | 130422,0 | 4320,0 | 86 |
| 64 | 18 | 13 | 2 | 158,0 | 132477,2 | 4450,0 | 87 |
| 70 | 5 | 14 | 2 | 126,0 | 134636,2 | 4542,0 | 88 |
| 84 | 7 | 1 | 2 | 158,0 | 137796,7 | 4665,0 | 89 |
| 91 | 1 | 5 | 1 | 147,0 | 138259,1 | 4795,0 | 90 |
| 82 | 16 | 9 | 1 | 104,0 | 138916,3 | 4868,0 | 91 |
| 60 | 12 | 14 | 2 | 165,0 | 140151,5 | 5017,0 | 92 |
| 67 | 11 | 10 | 2 | 152,0 | 141098,7 | 5137,0 | 93 |
| 97 | 11 | 14 | 1 | 172,0 | 141690,7 | 5289,0 | 94 |
| 83 | 3 | 11 | 2 | 146,0 | 141950,7 | 5409,0 | 95 |
| 51 | 14 | 6 | 2 | 139,0 | 142443,5 | 5537,0 | 96 |
| 59 | 3 | 11 | 2 | 162,0 | 142603,5 | 5683,0 | 97 |
| 95 | 16 | 12 | 1 | 165,0 | 143430,3 | 5809,0 | 98 |
| 65 | 10 | 4 | 2 | 161,0 | 145137,5 | 5948,0 | 99 |

EK-2. Proje bütçesi optimizasyonu (100_20_46_14 proje verisi)

Çizelge 2.1. PHibrit-B yöntemi

| Görev No | IK No | Yetkinlik | Yetkinlik Seviyesi | Proje Süresi | Proje Bütçesi | Boşta Bekleme Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 33 | 0 | 13 | 2 | 20,0 | 742,0 | 0,0 | 0 |
| 48 | 3 | 11 | 2 | 14,0 | 882,0 | 0,0 | 1 |
| 26 | 13 | 3 | 2 | 36,0 | 1544,4 | 0,0 | 2 |
| 92 | 3 | 1 | 1 | 27,0 | 1674,4 | 14,0 | 3 |
| 42 | 1 | 0 | 1 | 14,0 | 2055,2 | 14,0 | 4 |
| 71 | 3 | 8 | 0 | 54,0 | 2325,2 | 41,0 | 5 |
| 17 | 13 | 14 | 2 | 53,0 | 2638,0 | 77,0 | 6 |
| 44 | 13 | 2 | 0 | 90,0 | 3318,8 | 130,0 | 7 |
| 57 | 11 | 5 | 2 | 33,0 | 4295,6 | 130,0 | 8 |
| 85 | 16 | 9 | 1 | 21,0 | 4740,8 | 130,0 | 9 |
| 9 | 1 | 0 | 1 | 42,0 | 5502,4 | 144,0 | 10 |
| 74 | 4 | 8 | 2 | 40,0 | 7114,4 | 144,0 | 11 |
| 77 | 11 | 5 | 2 | 64,0 | 8032,0 | 177,0 | 12 |
| 22 | 16 | 12 | 1 | 34,0 | 8307,6 | 198,0 | 13 |
| 24 | 0 | 13 | 2 | 60,0 | 9791,6 | 218,0 | 14 |
| 86 | 3 | 11 | 2 | 85,0 | 10101,6 | 272,0 | 15 |
| 50 | 3 | 10 | 0 | 123,0 | 10481,6 | 357,0 | 16 |
| 37 | 13 | 14 | 2 | 130,0 | 11217,6 | 447,0 | 17 |
| 31 | 1 | 0 | 1 | 75,0 | 12115,2 | 489,0 | 18 |
| 87 | 1 | 0 | 1 | 96,0 | 12686,4 | 564,0 | 19 |
| 79 | 3 | 13 | 1 | 138,0 | 12836,4 | 687,0 | 20 |
| 40 | 11 | 5 | 2 | 86,0 | 13487,6 | 751,0 | 21 |
| 14 | 3 | 8 | 0 | 177,0 | 13877,6 | 889,0 | 22 |
| 81 | 13 | 14 | 2 | 166,0 | 14540,0 | 1019,0 | 23 |
| 75 | 0 | 4 | 2 | 86,0 | 15504,6 | 1079,0 | 24 |
| 98 | 1 | 9 | 0 | 119,0 | 16130,2 | 1175,0 | 25 |
| 43 | 14 | 6 | 2 | 19,0 | 16981,4 | 1175,0 | 26 |
| 99 | 1 | 0 | 1 | 137,0 | 17471,0 | 1294,0 | 27 |
| 23 | 13 | 3 | 2 | 186,0 | 17839,0 | 1460,0 | 28 |
| 49 | 3 | 11 | 2 | 206,0 | 18129,0 | 1637,0 | 29 |
| 34 | 3 | 11 | 2 | 218,0 | 18249,0 | 1843,0 | 30 |
| 45 | 13 | 14 | 2 | 213,0 | 18745,8 | 2029,0 | 31 |
| 25 | 3 | 8 | 0 | 252,0 | 19085,8 | 2247,0 | 32 |
| 1 | 13 | 3 | 2 | 250,0 | 19766,6 | 2460,0 | 33 |
| 11 | 13 | 3 | 2 | 287,0 | 20447,4 | 2710,0 | 34 |
| 90 | 16 | 12 | 1 | 74,0 | 21295,4 | 2744,0 | 35 |
| 36 | 0 | 13 | 2 | 126,0 | 22779,4 | 2830,0 | 36 |
| 88 | 4 | 8 | 2 | 65,0 | 23786,9 | 2870,0 | 37 |
| 3 | 4 | 8 | 2 | 101,0 | 25237,7 | 2935,0 | 38 |
| 12 | 6 | 2 | 2 | 25,0 | 27330,2 | 2935,0 | 39 |

EK-2. (devam) Proje bütçesi optimizasyonu (100_20_46_14 proje verisi)

Çizelge 2.1. (devam) PHibrit-B yöntemi

| Görev No | IK No | Yetkinlik | Yetkinlik Seviyesi | Proje Süresi | Proje Bütçesi | Boşta Bekleme Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 30 | 1 | 0 | 1 | 149,0 | 27656,6 | 3072,0 | 40 |
| 27 | 14 | 6 | 2 | 28,0 | 28059,8 | 3091,0 | 41 |
| 76 | 14 | 6 | 2 | 60,0 | 29493,4 | 3119,0 | 42 |
| 15 | 3 | 11 | 2 | 283,0 | 29803,4 | 3371,0 | 43 |
| 78 | 14 | 6 | 2 | 91,0 | 31192,2 | 3431,0 | 44 |
| 20 | 1 | 0 | 1 | 187,0 | 32225,8 | 3580,0 | 45 |
| 38 | 16 | 12 | 1 | 87,0 | 32501,4 | 3654,0 | 46 |
| 18 | 13 | 10 | 2 | 320,0 | 33108,6 | 3941,0 | 47 |
| 4 | 3 | 5 | 1 | 292,0 | 33198,6 | 4224,0 | 48 |
| 2 | 0 | 7 | 2 | 161,0 | 34497,1 | 4350,0 | 49 |
| 55 | 0 | 4 | 2 | 192,0 | 35647,2 | 4511,0 | 50 |
| 10 | 3 | 10 | 0 | 310,0 | 35827,2 | 4803,0 | 51 |
| 7 | 0 | 7 | 2 | 223,0 | 36977,3 | 4995,0 | 52 |
| 0 | 4 | 8 | 2 | 120,0 | 37743,0 | 5096,0 | 53 |
| 39 | 13 | 10 | 2 | 345,0 | 38203,0 | 5416,0 | 54 |
| 21 | 1 | 1 | 2 | 214,0 | 38937,4 | 5603,0 | 55 |
| 93 | 13 | 3 | 2 | 383,0 | 39636,6 | 5948,0 | 56 |
| 62 | 1 | 9 | 0 | 229,0 | 40044,6 | 6162,0 | 57 |
| 5 | 1 | 0 | 1 | 248,0 | 40561,4 | 6391,0 | 58 |
| 16 | 11 | 7 | 1 | 125,0 | 41715,8 | 6477,0 | 59 |
| 72 | 0 | 13 | 2 | 260,0 | 43088,5 | 6700,0 | 60 |
| 6 | 4 | 0 | 2 | 130,0 | 43491,5 | 6820,0 | 61 |
| 66 | 3 | 10 | 0 | 345,0 | 43841,5 | 7130,0 | 62 |
| 19 | 13 | 3 | 2 | 421,0 | 44540,7 | 7513,0 | 63 |
| 8 | 0 | 4 | 2 | 279,0 | 45245,6 | 7773,0 | 64 |
| 52 | 3 | 5 | 1 | 364,0 | 45435,6 | 8118,0 | 65 |
| 28 | 6 | 2 | 2 | 38,0 | 46523,7 | 8143,0 | 66 |
| 80 | 13 | 3 | 2 | 457,0 | 47186,1 | 8564,0 | 67 |
| 53 | 1 | 12 | 2 | 273,0 | 47757,3 | 8816,0 | 68 |
| 82 | 16 | 9 | 1 | 452,0 | 48414,5 | 9237,0 | 69 |
| 67 | 13 | 10 | 2 | 489,0 | 49003,3 | 9694,0 | 70 |
| 58 | 0 | 13 | 2 | 311,0 | 50190,5 | 9973,0 | 71 |
| 60 | 13 | 14 | 2 | 505,0 | 50484,9 | 10462,0 | 72 |
| 91 | 3 | 5 | 1 | 381,0 | 50654,9 | 10826,0 | 73 |
| 54 | 13 | 3 | 2 | 523,0 | 50986,1 | 11331,0 | 74 |
| 65 | 0 | 4 | 2 | 545,0 | 51802,3 | 11854,0 | 75 |
| 41 | 14 | 6 | 2 | 322,0 | 53549,5 | 12137,0 | 76 |
| 61 | 4 | 8 | 2 | 161,0 | 54798,8 | 12267,0 | 77 |
| 97 | 13 | 14 | 2 | 543,0 | 55166,8 | 12790,0 | 78 |
| 46 | 1 | 1 | 2 | 288,0 | 55574,8 | 13063,0 | 79 |

EK-2. (devam) Proje bütçesi optimizasyonu (100_20_46_14 proje verisi)

Çizelge 2.1. (devam) PHibrit-B yöntemi

| Görev No | IK No | Yetkinlik | Yetkinlik Seviyesi | Proje Süresi | Proje Bütçesi | Boşta Bekleme Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 51 | 14 | 6 | 2 | 333,0 | 56067,6 | 13385,0 | 80 |
| 89 | 4 | 8 | 2 | 288,0 | 57196,0 | 13645,0 | 81 |
| 47 | 0 | 13 | 2 | 581,0 | 58531,6 | 14190,0 | 82 |
| 35 | 3 | 8 | 0 | 415,0 | 58871,6 | 14571,0 | 83 |
| 84 | 1 | 1 | 2 | 323,0 | 59823,6 | 14859,0 | 84 |
| 94 | 14 | 6 | 2 | 347,0 | 60450,8 | 15192,0 | 85 |
| 32 | 3 | 11 | 2 | 429,0 | 60590,8 | 15607,0 | 86 |
| 70 | 13 | 14 | 2 | 577,0 | 61216,4 | 16150,0 | 87 |
| 29 | 11 | 7 | 1 | 317,0 | 62222,8 | 16433,0 | 88 |
| 69 | 15 | 6 | 0 | 182,0 | 63172,0 | 16594,0 | 89 |
| 56 | 13 | 3 | 2 | 601,0 | 63613,6 | 17171,0 | 90 |
| 83 | 3 | 11 | 2 | 455,0 | 63873,6 | 17600,0 | 91 |
| 13 | 0 | 7 | 2 | 603,0 | 64689,8 | 18181,0 | 92 |
| 73 | 6 | 2 | 2 | 219,0 | 67451,9 | 18367,0 | 93 |
| 63 | 3 | 1 | 1 | 470,0 | 67601,9 | 18822,0 | 94 |
| 59 | 3 | 11 | 2 | 486,0 | 67761,9 | 19292,0 | 95 |
| 96 | 3 | 8 | 0 | 636,0 | 68111,9 | 19893,0 | 96 |
| 68 | 3 | 5 | 1 | 668,0 | 68431,9 | 20529,0 | 97 |
| 64 | 0 | 13 | 2 | 631,0 | 69470,7 | 21132,0 | 98 |
| 95 | 16 | 12 | 1 | 616,0 | 70297,5 | 21709,0 | 99 |

EK-3. Boşta bekleme süresi optimizasyonu (100_20_46_14 proje verisi)

Çizelge 3.1. PHibrit-B yöntemi

| Görev No | IK No | Yetkinlik | Yetkinlik Seviyesi | Proje Süresi | Proje Bütçesi | Boşta Bekleme Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 76 | 12 | 6 | 0 | 32,0 | 2470,4 | 0,0 | 0 |
| 30 | 4 | 0 | 2 | 12,0 | 2954,0 | 0,0 | 1 |
| 7 | 0 | 7 | 2 | 31,0 | 4104,1 | 0,0 | 2 |
| 62 | 15 | 9 | 1 | 15,0 | 4782,1 | 0,0 | 3 |
| 48 | 17 | 11 | 0 | 14,0 | 5850,3 | 0,0 | 4 |
| 1 | 2 | 3 | 1 | 37,0 | 8344,1 | 0,0 | 5 |
| 99 | 10 | 0 | 2 | 18,0 | 9740,9 | 0,0 | 6 |
| 6 | 4 | 0 | 2 | 22,0 | 10143,9 | 12,0 | 7 |
| 5 | 1 | 0 | 1 | 19,0 | 10660,7 | 12,0 | 8 |
| 20 | 9 | 0 | 1 | 38,0 | 13765,3 | 12,0 | 9 |
| 27 | 19 | 6 | 2 | 9,0 | 14342,2 | 12,0 | 10 |
| 25 | 3 | 8 | 0 | 34,0 | 14682,2 | 12,0 | 11 |
| 40 | 2 | 5 | 2 | 59,0 | 16165,0 | 49,0 | 12 |
| 85 | 15 | 9 | 1 | 36,0 | 17114,2 | 64,0 | 13 |
| 10 | 15 | 10 | 0 | 54,0 | 17927,8 | 100,0 | 14 |
| 3 | 4 | 8 | 2 | 58,0 | 19378,6 | 122,0 | 15 |
| 79 | 17 | 13 | 1 | 29,0 | 20523,1 | 136,0 | 16 |
| 28 | 6 | 2 | 2 | 13,0 | 21611,2 | 136,0 | 17 |
| 4 | 5 | 5 | 0 | 9,0 | 22182,7 | 136,0 | 18 |
| 34 | 7 | 11 | 2 | 12,0 | 23266,3 | 136,0 | 19 |
| 42 | 8 | 0 | 0 | 14,0 | 24144,1 | 136,0 | 20 |
| 49 | 3 | 11 | 2 | 63,0 | 24434,1 | 170,0 | 21 |
| 16 | 5 | 7 | 1 | 48,0 | 26910,6 | 179,0 | 22 |
| 15 | 19 | 11 | 0 | 40,0 | 28897,7 | 188,0 | 23 |
| 86 | 3 | 11 | 2 | 94,0 | 29207,7 | 251,0 | 24 |
| 87 | 10 | 0 | 2 | 39,0 | 30837,3 | 269,0 | 25 |
| 22 | 16 | 12 | 1 | 13,0 | 31112,9 | 269,0 | 26 |
| 11 | 13 | 3 | 2 | 37,0 | 31793,7 | 269,0 | 27 |
| 9 | 9 | 0 | 1 | 66,0 | 34081,3 | 307,0 | 28 |
| 24 | 18 | 13 | 2 | 40,0 | 37017,3 | 307,0 | 29 |
| 44 | 6 | 2 | 2 | 50,0 | 40114,2 | 320,0 | 30 |
| 33 | 0 | 13 | 2 | 51,0 | 40856,2 | 351,0 | 31 |
| 19 | 17 | 3 | 1 | 67,0 | 43755,6 | 380,0 | 32 |
| 36 | 18 | 13 | 2 | 80,0 | 46691,6 | 420,0 | 33 |
| 52 | 1 | 5 | 1 | 38,0 | 47208,4 | 439,0 | 34 |
| 92 | 10 | 1 | 1 | 52,0 | 48217,2 | 478,0 | 35 |
| 72 | 18 | 13 | 2 | 117,0 | 50933,0 | 558,0 | 36 |
| 12 | 7 | 2 | 2 | 37,0 | 53190,5 | 570,0 | 37 |
| 78 | 7 | 6 | 0 | 68,0 | 55989,8 | 607,0 | 38 |
| 75 | 8 | 4 | 1 | 40,0 | 57620,0 | 621,0 | 39 |

EK-3. (devam) Boşta bekleme süresi optimizasyonu (100_20_46_14 proje verisi)

Çizelge 3.1. (devam) PHibrit-B yöntemi

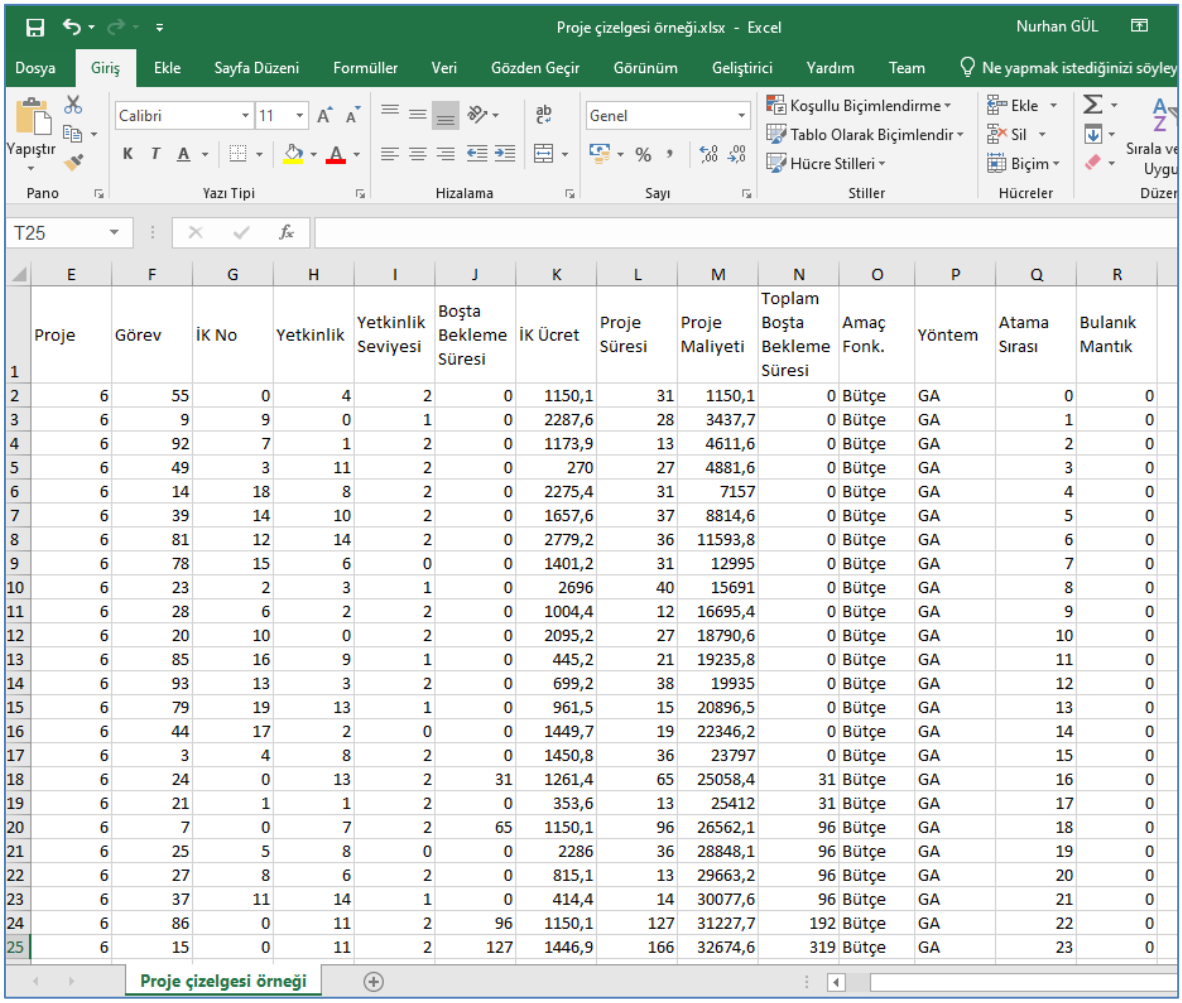
| Görev No | IK No | Yetkinlik | Yetkinlik Seviyesi | Proje Süresi | Proje Bütçesi | Boşta Bekleme Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 93 | 9 | 3 | 0 | 104,0 | 60724,6 | 687,0 | 40 |
| 21 | 1 | 1 | 2 | 65,0 | 61459,0 | 725,0 | 41 |
| 50 | 7 | 10 | 1 | 106,0 | 64890,4 | 793,0 | 42 |
| 39 | 12 | 10 | 2 | 57,0 | 66820,4 | 825,0 | 43 |
| 38 | 16 | 12 | 1 | 26,0 | 67096,0 | 838,0 | 44 |
| 71 | 19 | 8 | 0 | 67,0 | 68826,7 | 878,0 | 45 |
| 14 | 12 | 8 | 0 | 96,0 | 71837,5 | 935,0 | 46 |
| 66 | 15 | 10 | 0 | 89,0 | 73419,5 | 989,0 | 47 |
| 57 | 2 | 5 | 2 | 92,0 | 75643,7 | 1048,0 | 48 |
| 8 | 2 | 4 | 0 | 111,0 | 76924,3 | 1140,0 | 49 |
| 98 | 16 | 9 | 1 | 49,0 | 77411,9 | 1166,0 | 50 |
| 26 | 13 | 3 | 2 | 73,0 | 78074,3 | 1203,0 | 51 |
| 43 | 8 | 6 | 2 | 59,0 | 79265,6 | 1243,0 | 52 |
| 23 | 11 | 3 | 1 | 20,0 | 79857,6 | 1243,0 | 53 |
| 37 | 9 | 14 | 0 | 144,0 | 83125,6 | 1347,0 | 54 |
| 17 | 14 | 14 | 1 | 17,0 | 83887,2 | 1347,0 | 55 |
| 0 | 6 | 8 | 1 | 69,0 | 85477,5 | 1397,0 | 56 |
| 81 | 8 | 14 | 2 | 95,0 | 87734,7 | 1456,0 | 57 |
| 31 | 4 | 0 | 2 | 91,0 | 89064,6 | 1514,0 | 58 |
| 77 | 11 | 5 | 2 | 51,0 | 89982,2 | 1534,0 | 59 |
| 2 | 0 | 7 | 2 | 86,0 | 91280,7 | 1585,0 | 60 |
| 45 | 5 | 14 | 2 | 75,0 | 92995,2 | 1633,0 | 61 |
| 18 | 13 | 10 | 2 | 106,0 | 93602,4 | 1706,0 | 62 |
| 74 | 4 | 8 | 2 | 131,0 | 95214,4 | 1797,0 | 63 |
| 90 | 16 | 12 | 1 | 89,0 | 96062,4 | 1846,0 | 64 |
| 88 | 14 | 8 | 2 | 42,0 | 97182,4 | 1863,0 | 65 |
| 55 | 10 | 4 | 2 | 83,0 | 99588,0 | 1915,0 | 66 |
| 54 | 11 | 3 | 1 | 69,0 | 100120,8 | 1966,0 | 67 |
| 13 | 0 | 7 | 2 | 108,0 | 100937,0 | 2052,0 | 68 |
| 46 | 17 | 1 | 2 | 82,0 | 102081,5 | 2119,0 | 69 |
| 65 | 5 | 4 | 0 | 97,0 | 103478,5 | 2194,0 | 70 |
| 63 | 1 | 1 | 2 | 80,0 | 103886,5 | 2259,0 | 71 |
| 51 | 8 | 6 | 2 | 107,0 | 104576,2 | 2355,0 | 72 |
| 89 | 4 | 8 | 2 | 159,0 | 105704,6 | 2486,0 | 73 |
| 41 | 19 | 6 | 2 | 106,0 | 108204,5 | 2553,0 | 74 |
| 68 | 1 | 5 | 1 | 112,0 | 109074,9 | 2633,0 | 75 |
| 47 | 18 | 13 | 2 | 153,0 | 111717,3 | 2750,0 | 76 |
| 60 | 5 | 14 | 2 | 113,0 | 112733,3 | 2847,0 | 77 |
| 84 | 17 | 1 | 2 | 117,0 | 115403,8 | 2929,0 | 78 |
| 80 | 13 | 3 | 2 | 142,0 | 116066,2 | 3035,0 | 79 |

EK-3. (devam) Boşta bekleme süresi optimizasyonu (100_20_46_14 proje verisi)

Çizelge 3.1. (devam) PHibrit-B yöntemi

| Görev No | IK No | Yetkinlik | Yetkinlik Seviyesi | Proje Süresi | Proje Bütçesi | Boşta Bekleme Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 35 | 10 | 8 | 1 | 117,0 | 118704,6 | 3118,0 | 80 |
| 58 | 0 | 13 | 2 | 140,0 | 119891,8 | 3226,0 | 81 |
| 61 | 14 | 8 | 2 | 73,0 | 121280,6 | 3268,0 | 82 |
| 70 | 6 | 14 | 1 | 103,0 | 124126,4 | 3337,0 | 83 |
| 69 | 12 | 6 | 0 | 117,0 | 125747,6 | 3433,0 | 84 |
| 95 | 16 | 12 | 1 | 142,0 | 126574,4 | 3536,0 | 85 |
| 56 | 19 | 3 | 1 | 141,0 | 128112,8 | 3653,0 | 86 |
| 96 | 10 | 8 | 1 | 176,0 | 130828,8 | 3794,0 | 87 |
| 32 | 3 | 11 | 2 | 108,0 | 130968,8 | 3888,0 | 88 |
| 64 | 18 | 13 | 2 | 181,0 | 133024,0 | 4041,0 | 89 |
| 67 | 14 | 10 | 2 | 105,0 | 134457,6 | 4114,0 | 90 |
| 59 | 3 | 11 | 2 | 124,0 | 134617,6 | 4222,0 | 91 |
| 94 | 14 | 6 | 2 | 119,0 | 135244,8 | 4327,0 | 92 |
| 53 | 1 | 12 | 2 | 133,0 | 135816,0 | 4439,0 | 93 |
| 91 | 2 | 5 | 2 | 157,0 | 136961,8 | 4579,0 | 94 |
| 82 | 15 | 9 | 1 | 120,0 | 138363,0 | 4668,0 | 95 |
| 73 | 6 | 2 | 2 | 136,0 | 141125,1 | 4771,0 | 96 |
| 29 | 11 | 7 | 1 | 103,0 | 142131,5 | 4840,0 | 97 |
| 97 | 6 | 14 | 1 | 156,0 | 143805,5 | 4976,0 | 98 |
| 83 | 7 | 11 | 2 | 170,0 | 146153,3 | 5120,0 | 99 |

EK-4. Yazılım Çıktısı




| | E | F | G | H | I | J | K | L | M | N | O | P | Q | R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Proje | Görev | İK No | Yetkinlik | Yetkinlik Seviyesi | Boşta Bekleme Süresi | İK Ücret | Proje Süresi | Proje Maliyeti | Toplam Boşta Bekleme Süresi | Amaç Fonk. | Yöntem | Atama Sırası | Bulanık Mantık |
| 2 | 6 | 55 | 0 | 4 | 2 | 0 | 1150,1 | 31 | 1150,1 | 0 | Bütçe | GA | 0 | 0 |
| 3 | 6 | 9 | 9 | 0 | 1 | 0 | 2287,6 | 28 | 3437,7 | 0 | Bütçe | GA | 1 | 0 |
| 4 | 6 | 92 | 7 | 1 | 2 | 0 | 1173,9 | 13 | 4611,6 | 0 | Bütçe | GA | 2 | 0 |
| 5 | 6 | 49 | 3 | 11 | 2 | 0 | 270 | 27 | 4881,6 | 0 | Bütçe | GA | 3 | 0 |
| 6 | 6 | 14 | 18 | 8 | 2 | 0 | 2275,4 | 31 | 7157 | 0 | Bütçe | GA | 4 | 0 |
| 7 | 6 | 39 | 14 | 10 | 2 | 0 | 1657,6 | 37 | 8814,6 | 0 | Bütçe | GA | 5 | 0 |
| 8 | 6 | 81 | 12 | 14 | 2 | 0 | 2779,2 | 36 | 11593,8 | 0 | Bütçe | GA | 6 | 0 |
| 9 | 6 | 78 | 15 | 6 | 0 | 0 | 1401,2 | 31 | 12995 | 0 | Bütçe | GA | 7 | 0 |
| 10 | 6 | 23 | 2 | 3 | 1 | 0 | 2696 | 40 | 15691 | 0 | Bütçe | GA | 8 | 0 |
| 11 | 6 | 28 | 6 | 2 | 2 | 0 | 1004,4 | 12 | 16695,4 | 0 | Bütçe | GA | 9 | 0 |
| 12 | 6 | 20 | 10 | 0 | 2 | 0 | 2095,2 | 27 | 18790,6 | 0 | Bütçe | GA | 10 | 0 |
| 13 | 6 | 85 | 16 | 9 | 1 | 0 | 445,2 | 21 | 19235,8 | 0 | Bütçe | GA | 11 | 0 |
| 14 | 6 | 93 | 13 | 3 | 2 | 0 | 699,2 | 38 | 19935 | 0 | Bütçe | GA | 12 | 0 |
| 15 | 6 | 79 | 19 | 13 | 1 | 0 | 961,5 | 15 | 20896,5 | 0 | Bütçe | GA | 13 | 0 |
| 16 | 6 | 44 | 17 | 2 | 0 | 0 | 1449,7 | 19 | 22346,2 | 0 | Bütçe | GA | 14 | 0 |
| 17 | 6 | 3 | 4 | 8 | 2 | 0 | 1450,8 | 36 | 23797 | 0 | Bütçe | GA | 15 | 0 |
| 18 | 6 | 24 | 0 | 13 | 2 | 31 | 1261,4 | 65 | 25058,4 | 31 | Bütçe | GA | 16 | 0 |
| 19 | 6 | 21 | 1 | 1 | 2 | 0 | 353,6 | 13 | 25412 | 31 | Bütçe | GA | 17 | 0 |
| 20 | 6 | 7 | 0 | 7 | 2 | 65 | 1150,1 | 96 | 26562,1 | 96 | Bütçe | GA | 18 | 0 |
| 21 | 6 | 25 | 5 | 8 | 0 | 0 | 2286 | 36 | 28848,1 | 96 | Bütçe | GA | 19 | 0 |
| 22 | 6 | 27 | 8 | 6 | 2 | 0 | 815,1 | 13 | 29663,2 | 96 | Bütçe | GA | 20 | 0 |
| 23 | 6 | 37 | 11 | 14 | 1 | 0 | 414,4 | 14 | 30077,6 | 96 | Bütçe | GA | 21 | 0 |
| 24 | 6 | 86 | 0 | 11 | 2 | 96 | 1150,1 | 127 | 31227,7 | 192 | Bütçe | GA | 22 | 0 |
| 25 | 6 | 15 | 0 | 11 | 2 | 127 | 1446,9 | 166 | 32674,6 | 319 | Bütçe | GA | 23 | 0 |

Şekil 4.1.Excel formunda proje çizelgesi

EK-5. Şekil 6.3'te yer alan yakınsama hızları

Çizelge 5.1. PGA yöntemi

| Döngü | Değer | Döngü | Değer | Döngü | Değer | Döngü | Değer |
|---|---|---|---|---|---|---|---|
| 1 | 610 | 22 | 376 | 43 | 262 | 64 | 201 |
| 2 | 597 | 23 | 354 | 44 | 249 | 65 | 198 |
| 3 | 594 | 24 | 343 | 45 | 246 | 66 | 197 |
| 4 | 561 | 25 | 342 | 46 | 246 | 67 | 196 |
| 5 | 555 | 26 | 331 | 47 | 245 | 68 | 194 |
| 6 | 537 | 27 | 318 | 48 | 244 | 69 | 191 |
| 7 | 533 | 28 | 318 | 49 | 242 | 70 | 190 |
| 8 | 530 | 29 | 312 | 50 | 238 | 71 | 190 |
| 9 | 527 | 30 | 309 | 51 | 236 | 72 | 188 |
| 10 | 527 | 31 | 304 | 52 | 235 | 73 | 187 |
| 11 | 506 | 32 | 300 | 53 | 233 | 74 | 186 |
| 12 | 489 | 33 | 292 | 54 | 228 | 75 | 185 |
| 13 | 471 | 34 | 286 | 55 | 227 | 76 | 184 |
| 14 | 448 | 35 | 284 | 56 | 226 | 77 | 183 |
| 15 | 439 | 36 | 280 | 57 | 225 | 78 | 183 |
| 16 | 432 | 37 | 279 | 58 | 214 | 79 | 182 |
| 17 | 417 | 38 | 278 | 59 | 212 | 80 | 174 |
| 18 | 408 | 39 | 271 | 60 | 210 | 81 | 171 |
| 19 | 399 | 40 | 270 | 61 | 205 | | |
| 20 | 390 | 41 | 268 | 62 | 205 | | |
| 21 | 382 | 42 | 264 | 63 | 204 | | |

Çizelge 5.2. PHibrit yöntemi

| Döngü | Değer | Döngü | Değer | Döngü | Değer | Döngü | Değer |
|---|---|---|---|---|---|---|---|
| 0 | 671 | 18 | 307 | 36 | 227 | 54 | 190 |
| 1 | 564 | 19 | 293 | 37 | 221 | 55 | 189 |
| 2 | 558 | 20 | 290 | 38 | 221 | 56 | 189 |
| 3 | 536 | 21 | 276 | 39 | 220 | 57 | 185 |
| 4 | 521 | 22 | 265 | 40 | 216 | 58 | 184 |
| 5 | 515 | 23 | 262 | 41 | 212 | 59 | 181 |
| 6 | 497 | 24 | 260 | 42 | 208 | 60 | 177 |
| 7 | 457 | 25 | 259 | 43 | 206 | 61 | 176 |
| 8 | 432 | 26 | 256 | 44 | 205 | 62 | 174 |
| 9 | 400 | 27 | 256 | 45 | 204 | 63 | 174 |
| 10 | 382 | 28 | 254 | 46 | 204 | 64 | 173 |
| 11 | 379 | 29 | 252 | 47 | 202 | 65 | 169 |
| 12 | 378 | 30 | 250 | 48 | 200 | 66 | 168 |
| 13 | 358 | 31 | 243 | 49 | 200 | 67 | 161 |
| 14 | 357 | 32 | 239 | 50 | 196 | | |
| 15 | 355 | 33 | 239 | 51 | 196 | | |
| 16 | 329 | 34 | 234 | 52 | 195 | | |
| 17 | 318 | 35 | 230 | 53 | 191 | | |

EK-5. (devam) Şekil 6.3'te yer alan yakınsama hızları

Çizelge 5.3. PGA-B yöntemi

| Döngü | Değer | Döngü | Değer | Döngü | Değer |
|---|---|---|---|---|---|
| 1 | 644 | 11 | 289 | 21 | 198 |
| 2 | 588 | 12 | 285 | 22 | 196 |
| 3 | 529 | 13 | 271 | 23 | 194 |
| 4 | 488 | 14 | 250 | 24 | 185 |
| 5 | 479 | 15 | 242 | 25 | 181 |
| 6 | 478 | 16 | 239 | 26 | 180 |
| 7 | 432 | 17 | 230 | 27 | 178 |
| 8 | 404 | 18 | 216 | 28 | 174 |
| 9 | 400 | 19 | 209 | 29 | 170 |
| 10 | 366 | 20 | 204 | | |

Çizelge 5.4. PHibrit-B yöntemi

| Döngü | Değer | Döngü | Değer | Döngü | Değer | Döngü | Değer |
|---|---|---|---|---|---|---|---|
| 1 | 599 | 11 | 283 | 21 | 223 | 31 | 181 |
| 2 | 550 | 12 | 277 | 22 | 222 | 32 | 178 |
| 3 | 405 | 13 | 264 | 23 | 220 | 33 | 174 |
| 4 | 391 | 14 | 261 | 24 | 214 | 34 | 173 |
| 5 | 375 | 15 | 260 | 25 | 211 | 35 | 171 |
| 6 | 356 | 16 | 255 | 26 | 209 | 36 | 169 |
| 7 | 347 | 17 | 242 | 27 | 198 | 37 | 168 |
| 8 | 345 | 18 | 241 | 28 | 190 | 31 | 181 |
| 9 | 309 | 19 | 228 | 29 | 183 | 32 | 178 |
| 10 | 292 | 20 | 227 | 30 | 182 | 33 | 174 |

EK-6. Proje çizelgesi oluşturma örneği (100_20_46_14 proje verisi)

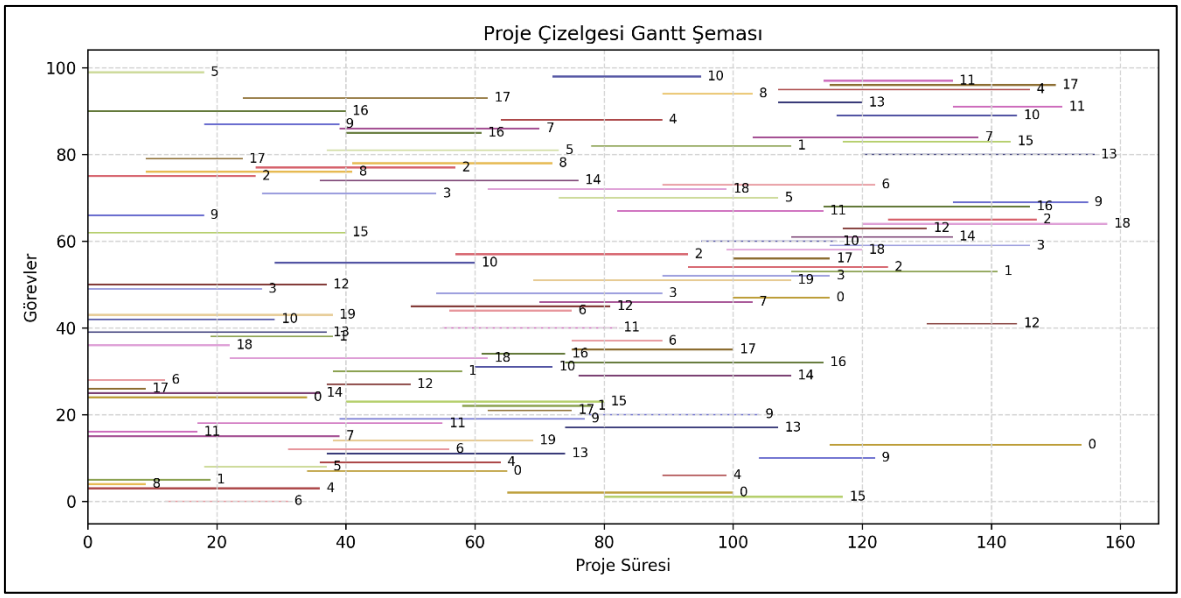
Çizelge 6.1. PGA-B yöntemi ile görev ataması

| Görev No | IK No | Proje Süresi | Atanma Sırası | Görev No | IK No | Proje Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 39 | 13 | 37 | 0 | 36 | 18 | 22 | 40 |
| 42 | 10 | 29 | 1 | 31 | 10 | 72 | 41 |
| 66 | 9 | 18 | 2 | 79 | 17 | 24 | 42 |
| 49 | 3 | 27 | 3 | 48 | 3 | 89 | 43 |
| 11 | 13 | 74 | 4 | 88 | 4 | 89 | 44 |
| 90 | 16 | 40 | 5 | 93 | 17 | 62 | 45 |
| 5 | 1 | 19 | 6 | 20 | 9 | 104 | 46 |
| 62 | 15 | 40 | 7 | 74 | 14 | 76 | 47 |
| 99 | 5 | 18 | 8 | 1 | 15 | 117 | 48 |
| 26 | 17 | 9 | 9 | 21 | 17 | 75 | 49 |
| 85 | 16 | 61 | 10 | 98 | 10 | 95 | 50 |
| 24 | 0 | 34 | 11 | 12 | 6 | 56 | 51 |
| 23 | 15 | 80 | 12 | 2 | 0 | 100 | 52 |
| 4 | 8 | 9 | 13 | 92 | 13 | 120 | 53 |
| 76 | 8 | 41 | 14 | 57 | 2 | 93 | 54 |
| 43 | 19 | 38 | 15 | 45 | 12 | 81 | 55 |
| 14 | 19 | 69 | 16 | 33 | 18 | 62 | 56 |
| 50 | 12 | 37 | 17 | 6 | 4 | 99 | 57 |
| 71 | 3 | 54 | 18 | 40 | 11 | 82 | 58 |
| 3 | 4 | 36 | 19 | 10 | 9 | 122 | 59 |
| 17 | 13 | 107 | 20 | 44 | 6 | 75 | 60 |
| 78 | 8 | 72 | 21 | 30 | 1 | 58 | 61 |
| 9 | 4 | 64 | 22 | 52 | 3 | 115 | 62 |
| 75 | 2 | 26 | 23 | 22 | 1 | 78 | 63 |
| 7 | 0 | 65 | 24 | 72 | 18 | 99 | 64 |
| 28 | 6 | 12 | 25 | 34 | 16 | 74 | 65 |
| 77 | 2 | 57 | 26 | 37 | 6 | 89 | 66 |
| 25 | 14 | 36 | 27 | 46 | 7 | 103 | 67 |
| 55 | 10 | 60 | 28 | 82 | 1 | 109 | 68 |
| 27 | 12 | 50 | 29 | 94 | 8 | 103 | 69 |
| 0 | 6 | 31 | 30 | 47 | 0 | 115 | 70 |
| 87 | 9 | 39 | 31 | 32 | 16 | 114 | 71 |
| 15 | 7 | 39 | 32 | 80 | 13 | 156 | 72 |
| 38 | 1 | 38 | 33 | 58 | 18 | 120 | 73 |
| 16 | 11 | 17 | 34 | 67 | 11 | 114 | 74 |
| 8 | 5 | 37 | 35 | 60 | 10 | 116 | 75 |
| 18 | 11 | 55 | 36 | 63 | 12 | 130 | 76 |
| 86 | 7 | 70 | 37 | 84 | 7 | 138 | 77 |
| 81 | 5 | 73 | 38 | 51 | 19 | 109 | 78 |
| 19 | 9 | 77 | 39 | 54 | 2 | 124 | 79 |

EK-6. (devam) Proje çizelgesi oluşturma örneği (100_20_46_14 proje verisi)

Çizelge 6.1. (devam) PGA-B yöntemi ile görev ataması

| Görev No | IK No | Proje Süresi | Atanma Sırası | Görev No | IK No | Proje Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 65 | 2 | 147 | 80 | 53 | 1 | 141 | 90 |
| 59 | 3 | 146 | 81 | 89 | 10 | 144 | 91 |
| 97 | 11 | 134 | 82 | 35 | 17 | 100 | 92 |
| 29 | 14 | 109 | 83 | 70 | 5 | 107 | 93 |
| 61 | 14 | 134 | 84 | 56 | 17 | 115 | 94 |
| 73 | 6 | 122 | 85 | 96 | 17 | 150 | 95 |
| 69 | 9 | 155 | 86 | 68 | 16 | 146 | 96 |
| 41 | 12 | 144 | 87 | 95 | 4 | 146 | 97 |
| 83 | 15 | 143 | 88 | 91 | 11 | 151 | 98 |
| 13 | 0 | 154 | 89 | 64 | 18 | 158 | 99 |




Şekil 6.1. PGA-B Gantt şeması

EK-7. Proje çizelgesi oluşturma örneği (100_20_46_14 proje verisi)

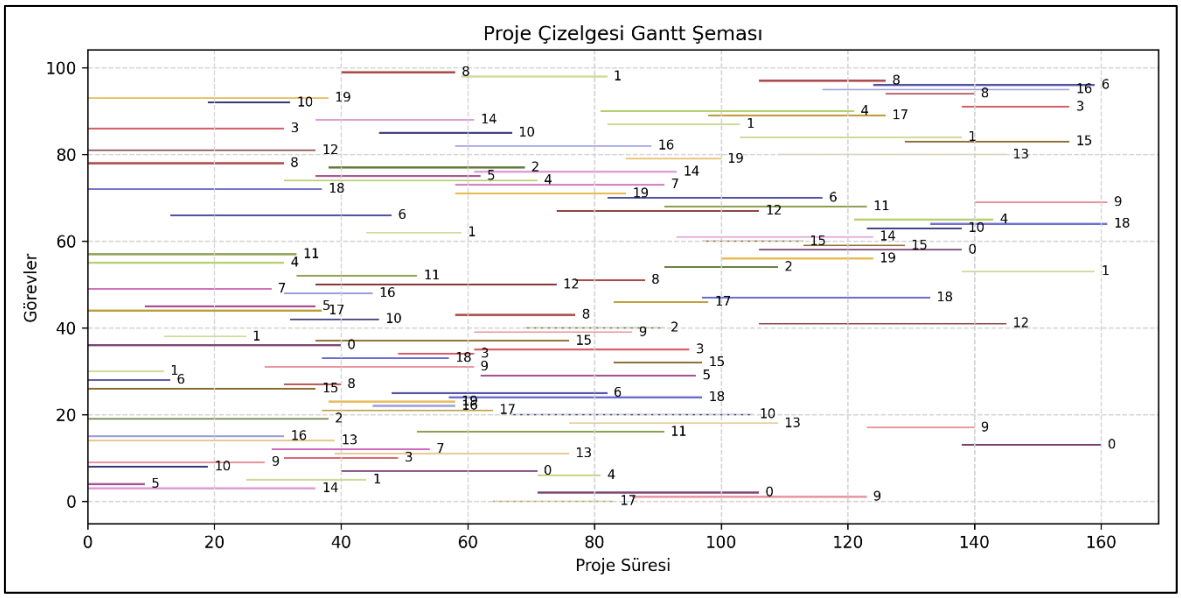
Çizelge 7.1. PHibrit-B yöntemi ile görev ataması

| Görev No | IK No | Proje Süresi | Atanma Sırası | Görev No | IK No | Proje Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 8 | 10 | 19 | 0 | 5 | 1 | 44 | 40 |
| 28 | 6 | 13 | 1 | 49 | 7 | 29 | 41 |
| 72 | 18 | 37 | 2 | 40 | 2 | 91 | 42 |
| 15 | 16 | 31 | 3 | 17 | 9 | 140 | 43 |
| 19 | 2 | 38 | 4 | 85 | 10 | 67 | 44 |
| 48 | 16 | 45 | 5 | 3 | 14 | 36 | 45 |
| 33 | 18 | 57 | 6 | 88 | 14 | 61 | 46 |
| 57 | 11 | 33 | 7 | 62 | 1 | 59 | 47 |
| 55 | 4 | 31 | 8 | 16 | 11 | 91 | 48 |
| 30 | 1 | 12 | 9 | 25 | 6 | 82 | 49 |
| 52 | 11 | 52 | 10 | 98 | 1 | 82 | 50 |
| 26 | 15 | 36 | 11 | 2 | 0 | 106 | 51 |
| 77 | 2 | 69 | 12 | 45 | 5 | 36 | 52 |
| 44 | 17 | 37 | 13 | 99 | 8 | 58 | 53 |
| 92 | 10 | 32 | 14 | 18 | 13 | 109 | 54 |
| 93 | 19 | 38 | 15 | 20 | 10 | 105 | 55 |
| 23 | 19 | 58 | 16 | 10 | 3 | 49 | 56 |
| 14 | 13 | 39 | 17 | 50 | 12 | 74 | 57 |
| 42 | 10 | 46 | 18 | 87 | 1 | 103 | 58 |
| 24 | 18 | 97 | 19 | 6 | 4 | 81 | 59 |
| 74 | 4 | 71 | 20 | 43 | 8 | 77 | 60 |
| 37 | 15 | 76 | 21 | 75 | 5 | 62 | 61 |
| 21 | 17 | 64 | 22 | 76 | 14 | 93 | 62 |
| 81 | 12 | 36 | 23 | 79 | 19 | 100 | 63 |
| 78 | 8 | 31 | 24 | 12 | 7 | 54 | 64 |
| 86 | 3 | 31 | 25 | 90 | 4 | 121 | 65 |
| 38 | 1 | 25 | 26 | 34 | 3 | 61 | 66 |
| 9 | 9 | 28 | 27 | 68 | 11 | 123 | 67 |
| 0 | 17 | 83 | 28 | 84 | 1 | 138 | 68 |
| 66 | 6 | 48 | 29 | 58 | 0 | 138 | 69 |
| 36 | 0 | 40 | 30 | 46 | 17 | 98 | 70 |
| 4 | 5 | 9 | 31 | 80 | 13 | 145 | 71 |
| 11 | 13 | 76 | 32 | 63 | 10 | 138 | 72 |
| 31 | 9 | 61 | 33 | 47 | 18 | 133 | 73 |
| 39 | 9 | 86 | 34 | 51 | 8 | 88 | 74 |
| 27 | 8 | 40 | 35 | 35 | 3 | 95 | 75 |
| 1 | 9 | 123 | 36 | 13 | 0 | 160 | 76 |
| 7 | 0 | 71 | 37 | 73 | 7 | 91 | 77 |
| 22 | 16 | 58 | 38 | 53 | 1 | 159 | 78 |
| 71 | 19 | 85 | 39 | 70 | 6 | 116 | 79 |

EK-7. (devam) Proje çizelgesi oluşturma örneği (100_20_46_14 proje verisi)

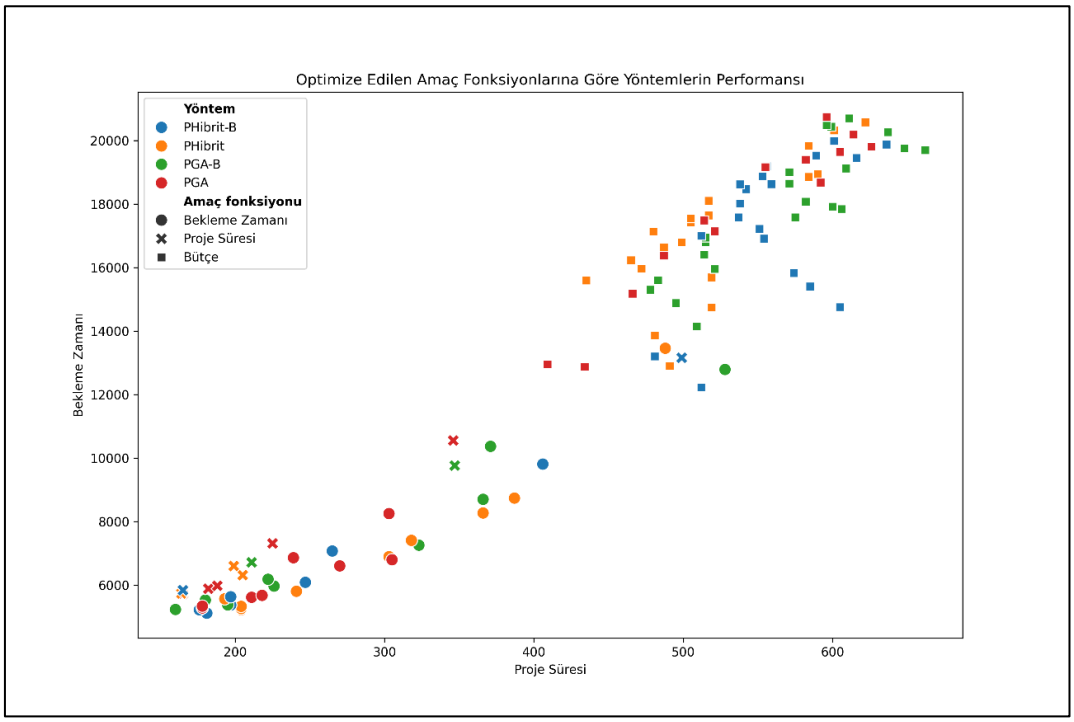
Çizelge 7.1. (devam) PHibrit-B yöntemi ile görev ataması

| Görev No | IK No | Proje Süresi | Atanma Sırası | Görev No | IK No | Proje Süresi | Atanma Sırası |
|---|---|---|---|---|---|---|---|
| 32 | 15 | 97 | 80 | 97 | 8 | 126 | 90 |
| 60 | 15 | 113 | 81 | 29 | 5 | 96 | 91 |
| 91 | 3 | 155 | 82 | 94 | 8 | 140 | 92 |
| 54 | 2 | 109 | 83 | 41 | 12 | 145 | 93 |
| 61 | 14 | 124 | 84 | 59 | 15 | 129 | 94 |
| 67 | 12 | 106 | 85 | 83 | 15 | 155 | 95 |
| 82 | 16 | 89 | 86 | 89 | 17 | 126 | 96 |
| 56 | 19 | 124 | 87 | 95 | 16 | 155 | 97 |
| 96 | 6 | 159 | 88 | 64 | 18 | 161 | 98 |
| 69 | 9 | 161 | 89 | 65 | 4 | 143 | 99 |

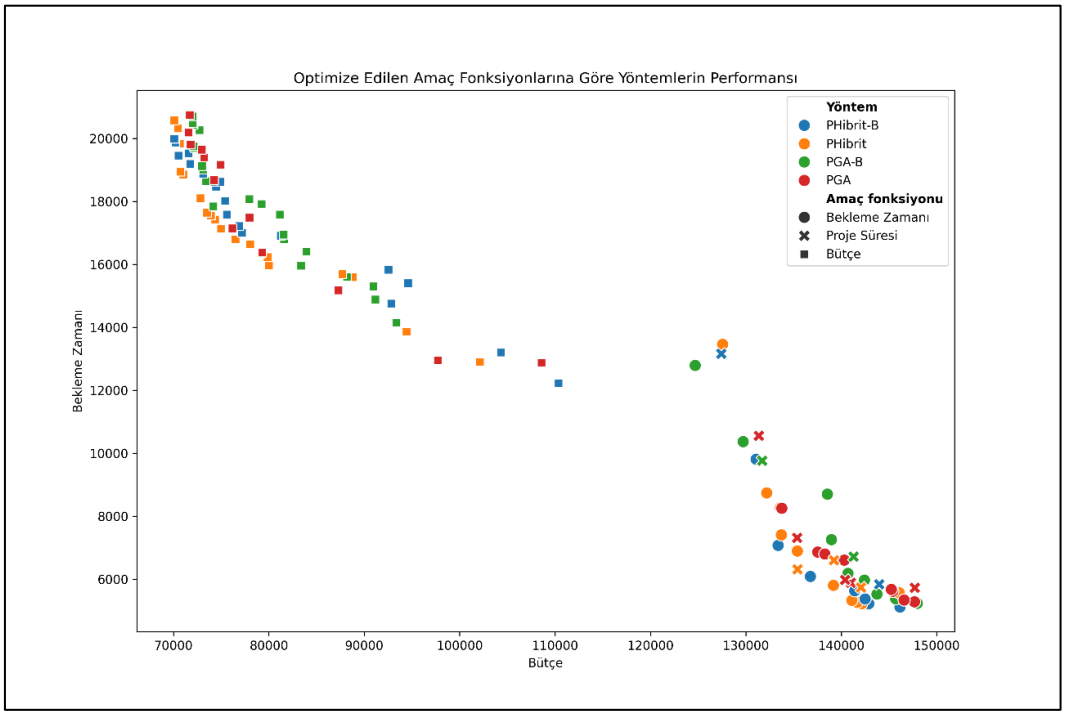



Şekil 7.1. PHibrit-B Gantt şeması

EK-8. Yöntemlerin baskın olmayan çözüm kümeleri (100_20_46_14 proje verisi)

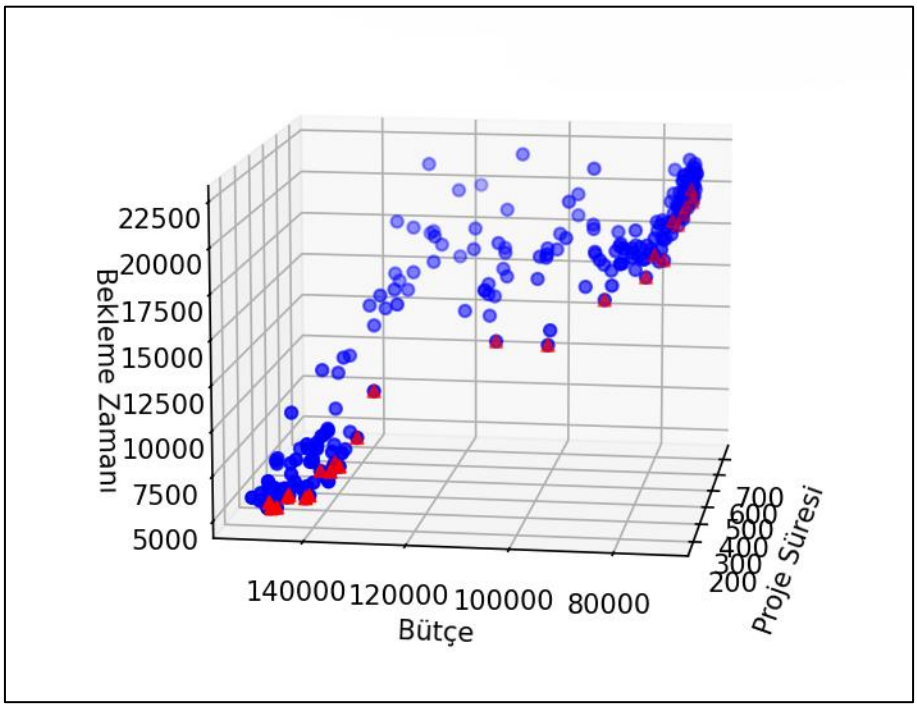



Şekil 8.2. Proje süresi-Bekleme zamanı karşılaştırması

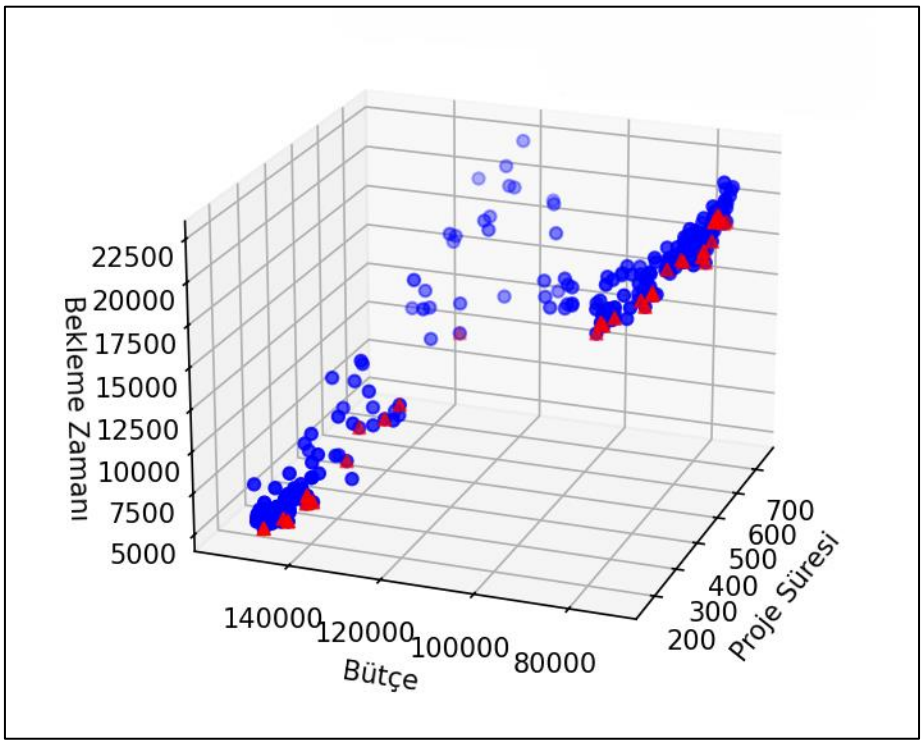



Şekil 8.3. Bütçe-Bekleme zamanı karşılaştırması

EK-9. Baskın olmayan çözüm kümesinin üç boyutlu gösterimi (100_20_46_14 proje verisi)

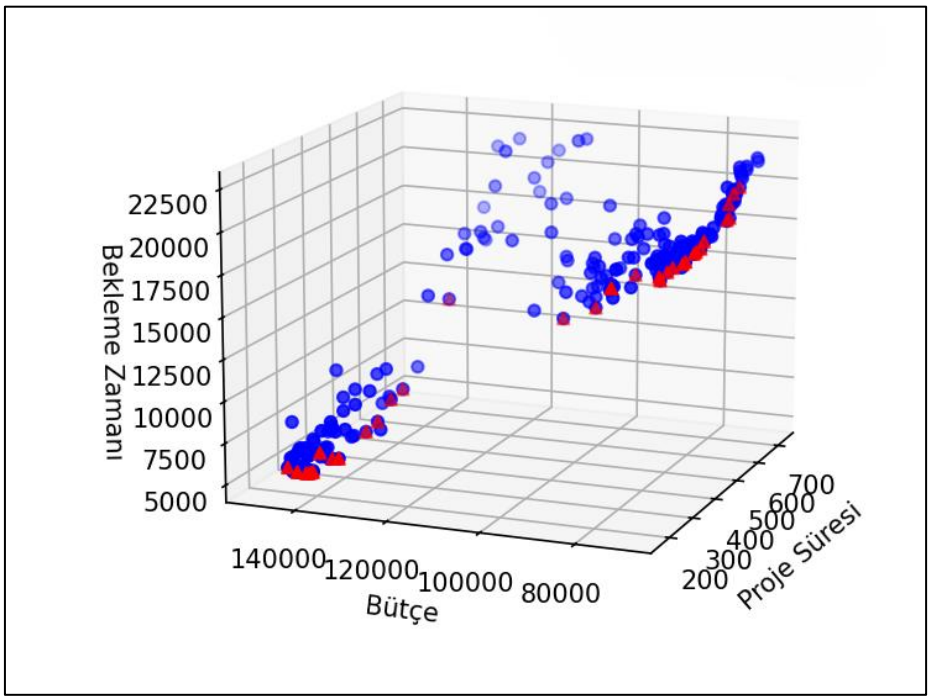



Şekil 9.1. PGA yöntemi

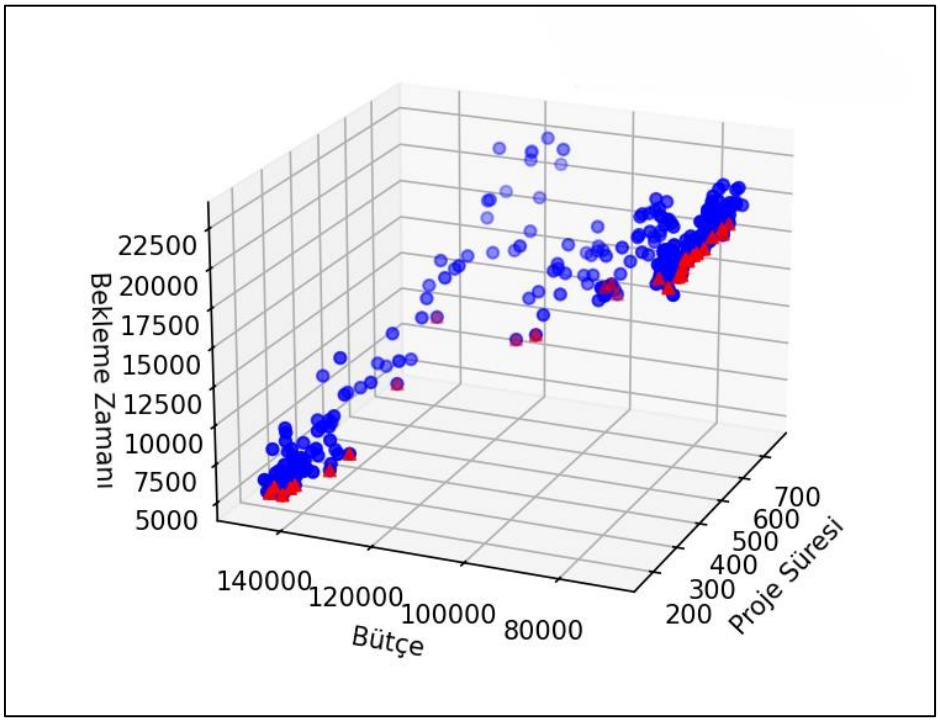



Şekil 9.2. PGA-B yöntemi

EK-9. (devam) Baskın olmayan çözüm kümesinin 3 boyutlu gösterimi (100_20_46_14 proje verisi)

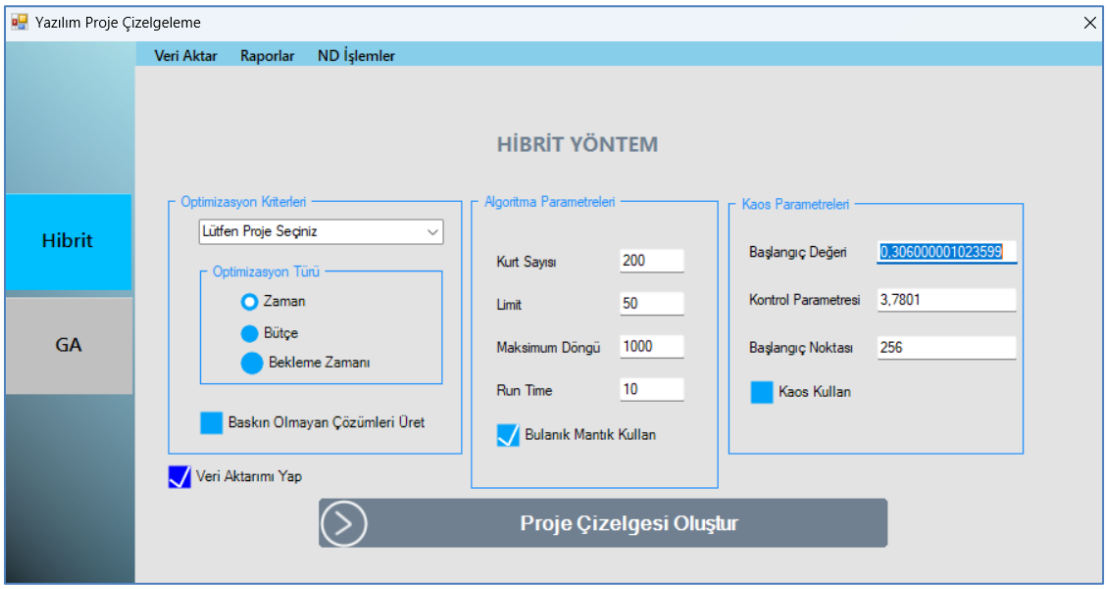



Şekil 9.3. PHibrit yöntemi

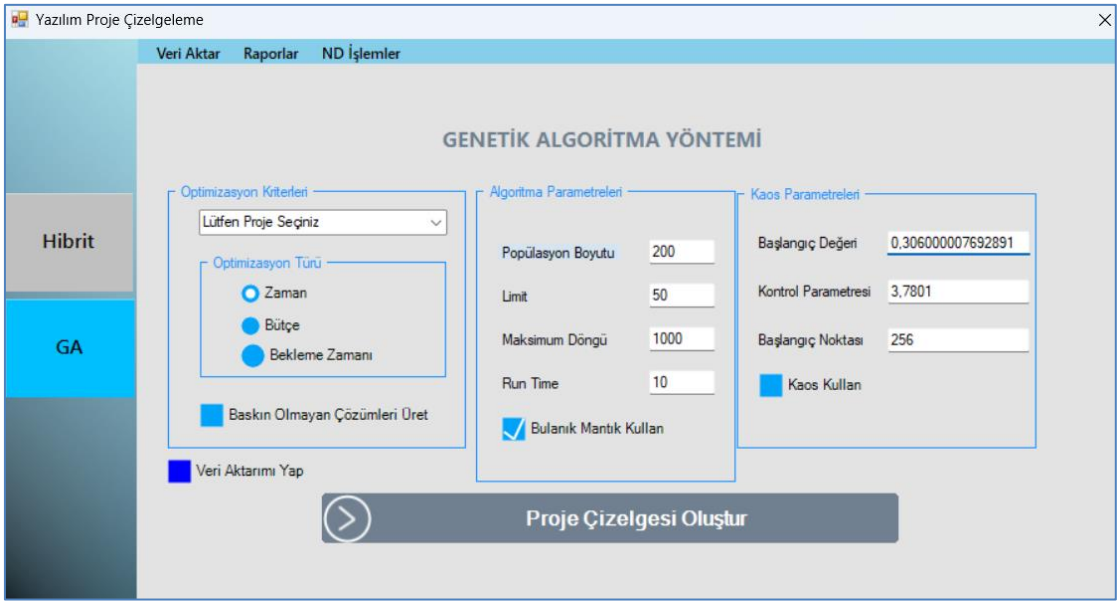



Şekil 9.4. PHibrit-B yöntemi

EK-10. Geliştirilen yazılımın ekran görüntüleri




Şekil 10.1. PHibrit ve PHibrit-B yöntemleri




Şekil 10.2. GA ve GA-B yöntemleri

*Gazili olmak ayrıcalıktır*