

**UKUROVA NİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

DOKTORA TEZİ

Esra KOYUNCU

**YENİ ÜRÜN GELİŞTİRME PROJELERİNDE ÇOKLU FAALİYET
ÇAKIŞTIRMALI DURUM İÇİN ÇİZELGELEME**

ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

ADANA, 2011

**ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**YENİ ÜRÜN GELİŞTİRME PROJELERİNDE ÇOKLU FAALİYET
ÇAKIŞTIRMALI DURUM İÇİN ÇİZELGELEME**

**Esra KOYUNCU
DOKTORA TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI**

Bu Tez 04/03/2011 Tarihinde Aşağıdaki Jüri Üyeleri Tarafından
Oybirliği/Oyçokluğu ile Kabul Edilmiştir.

.....
Prof. Dr. Rızvan EROL
DANIŞMAN

.....
Prof. Dr. Türkay DERELİ
ÜYE

.....
Yrd. Doç. Dr. Selçuk ÇOLAK
ÜYE

.....
Yrd. Doç. Dr. S. Noyan OĞULATA
ÜYE

.....
Yrd. Doç. Dr. Mustafa GÖÇKEN
ÜYE

Bu Tez Enstitümüz Endüstri Mühendisliği Anabilim Dalında hazırlanmıştır.

Kod No:

**Prof. Dr. İlhami YEĞİNGİL
Enstitü Müdürü**

**Bu Çalışma Ç.ü. Araştırma Projeleri Birimi Tarafından Desteklenmiştir.
Proje No: MMF2007D9**

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge, şekil ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ÖZ

DOKTORA TEZİ

YENİ ÜRÜN GELİŞTİRME PROJELERİNDE ÇOKLU FAALİYET ÇAKIŞTIRMALI DURUM İÇİN ÇİZELGELME

Esra KOYUNCU

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

Danışman :Prof. Dr. Rızvan EROL
Yıl: 2011, Sayfa: 159
Jüri :Prof. Dr. Rızvan EROL
Prof.Dr. Türkay DERELİ
Yrd. Dr. S.Noyan OĞULATA
Yrd. Doç. Dr. Selçuk ÇOLAK
Yrd. Doç. Dr. Mustafa GÖÇKEN

Yeni ürün geliştirme projelerinde ürünün müşteriye bir an önce sunulması, daha yüksek satış miktarı sağlanabilmesi ve fiyat avantajı açısından oldukça önemlidir. Bu nedenle, yeni ürün geliştirme projelerinde (YÜG) bitiş sürelerinin kısaltılabilmesi için bir takım teknikler önerilmiştir. Bu çalışmada, iyi planlandığı takdirde proje bitiş süresini kısalttığı bilinen ve bu amaçla en çok kullanılan tekniklerden biri olan faaliyetlerin üst üste çakıştırılması uygulamasını içeren bir proje çizelgeleme ve kaynak tahsisi modeli ortaya konmuştur. Geliştirilen model, kaynak kısıtları ve faaliyetler arasındaki öncelik ilişkilerini dikkate alarak proje bitiş süresini en küçükleyecek şekilde faaliyetleri çizelgelerken aynı zamanda yeniden çalışma miktarını dikkate alarak proje bitiş süresini en küçükleyecek üst üste çakıştırma miktarını da belirlemektedir.

Ele alınan problemin çözüm karmaşıklığının olması nedeniyle, farklı ağ karmaşıklığı, kaynak kısıtlılığı, ve faaliyet sayısına sahip projeler için problemi makul sürelerde çözebilecek Parçacık Sürü Optimizasyonu (PSO) temelli bir yaklaşım geliştirilmiştir. Geliştirilen algorithmada, parçacıkların temsil edilmesinde öncelik temelli gösterim kullanılmıştır. Parçacıkların çizelgeye dönüştürülmesi içinse probleme uyarlanmış yani üst üste çakıştırmaya imkan verecek şekilde güncellenmiş seri çizelge oluşturma planı kullanılmıştır. Geliştirilen sezgisel yöntem, farklı ağ karmaşıklığı, kaynak kısıtlılığı ve proje büyüklüğüne sahip proje ağları ile test edilmiştir. Ayrıca, bir elektronik ürün geliştirme projesi için farklı alt seviye faaliyet duyarlılığı ve üst seviye faaliyet gelişim parametresi değerleri dikkate alınarak bir senaryo analizi gerçekleştirilmiştir.

Anahtar Kelimeler: YÜG Projesi, Üst Üste Çakıştırma, Kaynak Kısıtlı Proje Çizelgeleme, Parçacık Sürü Optimizasyonu

ABSTRACT

PhD THESIS

SCHEDULING FOR NEW PRODUCT DEVELOPMENT PROJECT INCLUDING MULTIPLE ACTIVITY OVERLAPPING

Esra KOYUNCU

**ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF INDUSTRIAL ENGINEERING**

Supervisor : Prof. Dr. Rızvan EROL
Year: 2011, Page: 145
Jury : Prof. Dr. Rızvan EROL
Prof. Dr. Türkay DERELİ
Asst. Prof. Dr. S. Noyan OĞULATA
Asst. Prof. Dr. Selçuk ÇOLAK
Asst. Prof. Dr. Mustafa GÖÇKEN

Faster time to market for new products is important to gain price advantage and higher sales volume. Therefore, many techniques have been suggested to reduce project duration. Activity overlapping is commonly regarded as the most promising strategy to reduce product development times. In this study, a mathematical model is proposed for scheduling new product development (NPD) projects, which includes calculating optimal overlap duration between predetermined design activities, taking into account rework duration. The mathematical model aims to minimize project duration considering resource and precedence constraints and also optimal overlapping strategy.

Since the problem is known as NP-hard problem, a Particle Swarm Optimization (PSO) based approach is proposed to solve the problem for the projects having different number of activity, network complexity and resource strength. Activities priorities for scheduling are represented by particles and revised serial scheme including overlapping is utilized to transform the particle-represented priorities to a feasible schedule according to the precedence and resource constraints. The algorithm is tested using test problems which have different number of activity, network complexity and resource strength. Also, a scenario analysis is conducted taking into account downstream activity sensitivity and upstream activity evolution parameter for an electronic product development project.

Key Words: NPD project, Activity overlapping, Resource Constrained Project Scheduling Problem (RCPS), Particle Swarm Optimization.

ÖNSÖZ

Firmalar açısından ürün geliştirme prosesinin önemi her zaman kabul görmüştür ancak geçmişten günümüze kadar olan süreçte firmaların bu proseste önem verdikleri noktalar farklılaşmıştır. 1960’lı yıllarda, ürün geliştirmede itici güç teknolojik gelişmeler olmuştur. Üretkenlik artmış, geniş pazarlarda büyük miktarlarda üretim yapabilen firmalar rakiplerini geride bırakmışlardır. Bu nedenle ürün geliştirme çalışmaları daha çok mevcut teknolojileri geliştirmeye odaklanmıştır. 1970’li yıllarda, petrol krizinin de etkisiyle işletmeler, maliyetlerini düşürme yoluna gitmişlerdir. Düşük maliyetli ürünler geliştiren işletmeler pazarda başarı sağlamışlardır. 1980’li yıllarda teknolojik gelişmelerin ürün geliştirme üzerindeki etkisi daha çok artmış ve ürünün kalite boyutu önem kazanmıştır. 1990’lı yıllarda işletmeler, daha çok hız mücadelesine girmişlerdir. Pazara rakiplerde önce ürünü sunmanın avantajlarını görmüşler ve pazar koşullarını belirlemek için önemli stratejiler geliştirmişlerdir.

Küresel rekabetin arttığı 2000’li yıllarda ise, bir çok firma ürün geliştirme zamanlarını daha kısa yapmaya odaklanmıştır. Çünkü ürün geliştirme zamanının kısaltılması, firmanın pazar payı ile ilgili avantajlarını ve uzun dönem rekabetçi avantajlarını arttırmaktadır. Firmalar, ürünü pazara ne kadar erken sunarsa, ürün yaşam çevrimi boyunca daha yüksek gelir elde etmekte ve pazar karakteristiklerinin değişmesi durumunda bile pazara sunulan ürünün pazarda tutunma şansı daha yüksek olmaktadır. Bu avantajları nedeniyle, ürün geliştirme süresini kısaltılabilmesi amacıyla birtakım araç ve teknikler ortaya konmuştur. Bu teknikler içerisinde en çok kullanılanlardan birisi, yeni ürün geliştirme projelerinde yer alan faaliyetlerin bilgi transferi şeklinde ilişkiye sahip olmaları nedeniyle uygulanabilir olan “faaliyetlerin üst üste çakıştırılmasıdır (overlapping)” .

Bu çalışmada, belirli faaliyetler arasındaki üst üste çakıştırma miktarının, proje bitiş süresini en küçükleyecek şekilde hesaplanmasını da içeren bir çizelgeleme ve kaynak tahsisi modeli geliştirilmiştir. Problemin çözüm karmaşıklığının olması nedeniyle, büyük çapta problemlerin makul sürelerde çözülebilmesi için probleme uyarlanmış PSO algoritması önerilmiştir.

TEŞEKKÜR

Çalışmalarım sırasında yardımlarını esirgemeyen ve yönlendirici fikirleri ile yol gösteren danışman hocam Sayın Prof. Dr. Rızvan EROL'a teşekkür ederim.

Doktora Tez İzleme Komitesi üyesi Sayın Prof. Dr. Türkay DERELİ ve Sayın Yrd. Doç. Dr.S. Noyan OĞULATA'ya doktora tezi raporlarımı değerlendirerek yapmış oldukları yönlendirici ve olumlu katkılarından dolayı teşekkür ederim. Doktora tezi jüri üyelerinden Sayın Selçuk ÇOLAK ve Sayın Mustafa GÖÇKEN'e yapıcı ve yönlendirici fikirleriyle katkıda bulundukları için teşekkürlerimi sunarım.

Doktora çalışmalarım esnasında maddi destek veren TUBİTAK Yurt İçi Doktora Burs Programı ve Ç.Ü. Bilimsel Araştırma Projeleri Birimi'ne (Proje no: MMF 2007D9) içten teşekkürlerimi sunarım.

Bütün fedakarlıkları ve destekleri için annem Aliye KARAKAŞ ve babam Hüseyin KARAKAŞ'a çok teşekkür ederim. Çalışmam sırasında göstermiş oldukları anlayış, destek ve verdikleri moral için kardeşim Tuğba KARAKAŞ, ve eşim Melik KOYUNCU'ya çok teşekkür ederim. Dostluğu ile bana moral ve güç veren Z.Figen ANTMEN'e teşekkür ederim. Tüm yorgunluğumu unutturan ve tekrar çalışabilmem için gerekli enerjiyi veren neşe kaynağım kızım Defne KOYUNCU'ya teşekkür ederim.

İÇİNDEKİLER

SAYFA

ÖZ	I
ABSTRACT	II
ÖNSÖZ	III
TEŞEKKÜR.....	IV
İÇİNDEKİLER	V
TABLOLAR DİZİNİ	VII
ŞEKİLLER DİZİNİ.....	IX
1. GİRİŞ	1
1.1. Yeni Ürün Tanımı ve Ürün Geliştirme Nedenleri.....	1
1.2. Yeni Ürün Geliştirme Süreci	4
1.3. Yeni Ürün Geliştirme Projelerinde Üst Üste Çakıştırma Uygulaması.....	8
1.4. Problemin Tanımı.....	10
1.5. Çalışmanın Amacı ve Kapsamı	12
1.6. Varsayımlar	13
1.7. Çalışmanın Adımları	14
1.8. Çalışmanın Orijinal Katkısı.....	15
2. ÖNCEKİ ÇALIŞMALAR.....	17
2.1. Kaynak Kısıtlı Proje Çizelgeleme Çalışmaları.....	17
2.1.1. KKPC ile İlgili Belirsizlik İçermeyen Çalışmalar	19
2.1.1.1. Kesin Çözüm Yöntemlerini Kullanan Çalışmalar.....	19
2.1.1.2. Sezgisel Yaklaşımların Kullanıldığı Çalışmalar	20
2.1.2. KKPC’de Faaliyet Sürelerindeki Belirsizliği Dikkate Alan Çalışmalar.....	27
2.2. Yeni Ürün Geliştirme Projeleri İle İlgili Modelleme Çalışmaları.....	28
2.2.1. Yeni Ürün Geliştirme Projelerinde Çizelgeleme Çalışmaları.....	29
2.2.2. Yeni Ürün Geliştirme Projelerinde Faaliyetlerin Üst Üste Çakıştırılması ile İlgili Çalışmalar	32
2.3. Literatürün Değerlendirilmesi	33
3. MATERYAL VE METOD	35

3.1. Materyal.....	35
3.2. Metod	38
3.2.1. Problemin Formülasyonu	38
3.2.1.1. Sistemin Özellikleri.....	38
3.2.1.2. Varsayımlar	42
3.2.1.3. Girdi Değişkenleri ve Parametreler.....	44
3.2.1.4. Performans Değişkenleri	45
3.2.2. Matematik Programlama Modeli	45
3.2.2.1. Notasyon	45
3.2.2.2. Matematiksel Model.....	47
3.2.3. Kullanılan İleri Sezgisel Karar Verilmesi	50
3.2.4. Parçacık Sürü Optimizasyonu Algoritması.....	52
3.2.4.1. KKPÇP’de Parçacıkların Temsili	53
3.2.5.2. Çizelge Oluşturma Planları	57
3.2.5. Önerilen PSO Algoritması	58
3.2.6. Geliştirilen Algoritmaların Programlanması.....	66
3.2.6.1. Algoritmaların Ara Yüzlerin Tanıtılması.....	67
3.2.6.2. Verilerin Programa Aktarılması.....	73
4. ARAŞTIRMA BULGULARI VE TARTIŞMA	77
4.1. Analitik Modelin Çözüm Süresinin İncelenmesi	77
4.2. PSO Algoritmasının Performansının Test Edilmesi.....	81
4.3. Üst Üste Çakıştırma Yapılacak Alternatif Faaliyetlerin Belirlenmesi	89
4.4. Üst Üste Çakıştırma İçeren PSO Algoritmasının Değerlendirilmesi	99
4.5. Senaryo Analizleri	116
5. SONUÇLAR VE ÖNERİLER	127
5.1. Çalışmanın Özeti	127
5.2. Sonuçlar	129
5.3. Sonraki Çalışmalar İçin Öneriler.....	133
KAYNAKLAR	135
ÖZGEÇMİŞ	143
EKLER.....	145

TABLolar DİZİNİ**SAYFA**

Tablo 1.1.	Ürün geliştirme süreçleri.....	5
Tablo 3.1.	Test amaçlı kullanılan proje verilerinin özellikleri	37
Tablo 4.1.	30 faaliyetli proje ağlarının alternatif tekli üst üste karşılaştırma seçenekleri göre kesin çözüm durumları.....	79
Tablo 4.2.	30 faaliyetli problemlerin üst üste karşılaştırma içermeyen PSO algoritması ile elde edilen sonuçları.....	83
Tablo 4.3.	60 faaliyetli problemlerin üst üste karşılaştırma içermeyen PSO algoritması ile elde edilen sonuçları.....	85
Tablo 4.4.	90 faaliyetli problemlerin üst üste karşılaştırma içermeyen PSO algoritması ile elde edilen sonuçları.....	86
Tablo 4.5.	120 faaliyetli problemlerin üst üste karşılaştırma içermeyen PSO algoritması ile elde edilen sonuçları.....	87
Tablo 4.6.	30 faaliyetli test problemleri için tekli üst üste karşılaştırma sonuçları.....	101
Tablo 4.7.	30 faaliyetli test problemleri için çoklu üst üste karşılaştırma sonuçları	103
Tablo 4.8.	30 faaliyetli test problemleri için üst üste karşılaştırma ile elde edilen süre kazançları	104
Tablo 4.9.	60 faaliyetli test problemleri için tekli üst üste karşılaştırma sonuçları.....	106
Tablo 4.10.	60 faaliyetli test problemleri için çoklu üst üste karşılaştırma sonuçları	107
Tablo 4.11.	60 faaliyetli test problemleri için üst üste karşılaştırma ile elde edilen süre kazançları	108
Tablo 4.12.	90 faaliyetli test problemleri için tekli üst üste karşılaştırma sonuçları.....	109
Tablo 4.13.	90 faaliyetli test problemleri için çoklu üst üste karşılaştırma sonuçları	111
Tablo 4.14.	90 faaliyetli test problemleri için üst üste karşılaştırma ile elde edilen süre kazançları	112
Tablo 4.15.	120 faaliyetli test problemleri için tekli üst üste karşılaştırma sonuçları	113

Tablo 4.16. 120 faaliyetli test problemleri için çoklu üst üste çakıştırma sonuçları	114
Tablo 4.17. 120 faaliyetli test problemleri için üst üste çakıştırma ile elde edilen süre kazançları	115
Tablo 4.18. Senaryo durumları	117
Tablo 4.19. Elektronik ürün geliştirme projesinin farklı senaryo analizlerine göre tekli üst üste çakıştırma sonuçları	119
Tablo 4.20. Elektronik ürün geliştirme projesinin farklı senaryo analizlerine göre çoklu üst üste çakıştırma sonuçları	125

ŞEKİLLER DİZİNİ

SAYFA

Şekil 1.1. Faaliyetlerin üst üste çakıştırılması.....	9
Şekil 2.1. Kaynak kısıtlı proje çizelgeleme problemleri ile ilgili çalışmalar.	18
Şekil 2.2. Ürün geliştirme projelerinde modelleme çalışmaları.....	29
Şekil 3.1. Faaliyet listesi gösterim örneği	58
Şekil 3.2. Öncelik temelli gösterim örneği.....	58
Şekil 3.3. Öncelik temelli gösterimli ve seri çizelgeleme kullanan PSO Algoritması.....	62
Şekil 3.4. Üst üste çakıştırma içermeyen programın ara yüzü-Dosyalar bölümü- ...	67
Şekil 3.5. Üst üste çakıştırma içermeyen programın ara yüzü-Faaliyetler bölümü-	68
Şekil 3.6. Üst üste çakıştırma içermeyen programın ara yüzü-Parametreler bölümü-	69
Şekil 3.7. Üst üste çakıştırma içermeyen programın ara yüzü-Kaynaklar bölümü-	69
Şekil 3.8. Üst üste çakıştırma içermeyen programın ara yüzü -Çözücü bölümü-	70
Şekil 3.9. Üst üste çakıştırma içeren programın ara yüzü -Dosyalar bölümü-.....	71
Şekil 3.10. Üst üste çakıştırma içeren programın ara yüzü -Çakışan faaliyetler bölümü-	71
Şekil 3.11. Üst üste çakıştırma uygulamasını içeren programın ara yüzü- Parametreler bölümü-.....	72
Şekil 3.12. Çoklu üst üste çakıştırma içeren programın ara yüzü-Çakışan faaliyetler bölümü-	73
Şekil 3.13. Veri dosyalarının oluşturulması	74
Şekil 3.14. Verilerin girilmesi	75
Şekil 3.15. Mevcut verilerin programa aktarılması.....	76
Şekil 4.1. 1 numaralı problem için çözüm ekranı.....	84
Şekil 4.2. 120 faaliyetli projeler için yinelemenin tamamlanma süreleri.....	88
Şekil 4.3. Düzenli kritik yol örneği- 3 numaralı problemin GANNT şeması	90

Şekil 4.4. Parçalı kritik yol içeren problemlerde üst üste çakıştırma yapılacak faaliyetlerin belirlenmesi-Adım 1-.....	91
Şekil 4.5. Parçalı kritik yol içeren problemlerde üst üste çakıştırma yapılacak faaliyetlerin belirlenmesi-Adım 2-.....	92
Şekil 4.6. Parçalı kritik yol içeren problemlerde üst üste çakıştırma yapılacak faaliyetlerin belirlenmesi-Adım 3-.....	92
Şekil 4.7. 1 numaralı problemde 11 ve 27 numaralı faaliyetler arasında üst üste çakıştırma içeren GANNT şeması	93
Şekil 4.8. Parçalı kritik yol içeren problemlerde üst üste çakıştırma yapılacak faaliyetlerin belirlenmesi-Adım 4-.....	94
Şekil 4.9. 1 numaralı problem için 11-27 ve 2-7 numaralı faaliyetler arasında üst üste çakıştırmayı içeren GANNT şeması	95
Şekil 4.10. Parçalı kritik yol için 2. örnek-8 numaralı problemin GANNT şeması-.....	95
Şekil 4.11. Bir noktada kesişen paralel kritik yol örneği-18 numaralı problemin GANNT şeması-.....	97
Şekil 4.12. Paralel kritik yol örneği -22 numaralı problemin GANNT Şeması	98
Şekil 4.13. Bir elektronik ürün geliştirme projesine ait GANNT şeması.....	118
Şekil 4.14. Senaryo 2 durumunda, elektronik ürün geliştirme projesinin 2 ve 3 numaralı faaliyetleri arasında üst üste çakıştırma içeren GANNT şeması.....	120
Şekil 4.15. Senaryo 3 durumunda, elektronik ürün geliştirme projesinin 2 ve 3 numaralı faaliyetleri arasında üst üste çakıştırma içeren GANNT şeması.....	121
Şekil 4.16. Senaryo 5 durumunda, elektronik ürün geliştirme projesinin 2 ve 3 numaralı faaliyetleri arasında üst üste çakıştırma içeren GANNT şeması.....	122
Şekil 4.17. Elektronik ürün geliştirme projesinin 2 ve 3 numaralı faaliyetleri arasında üst üste çakıştırma içeren çizelgenin farklı senaryolara göre sonuçları	123

Şekil 4.18. Elektronik ürün geliştirme projesinin 3 ve 5 numaralı faaliyetleri
arasındaki üst üste çakıştırmanın senaryo analizleri sonuçları 124

1. GİRİŞ

Ele alınan problemin, “yeni ürün geliştirme projelerinde çizelgeleme” olması nedeniyle bu bölümde öncelikle, yeni ürün tanımı ve yeni ürün geliştirme süreçleri hakkında genel bilgiler verilmektedir. Daha sonraki bölümlerde ise sırasıyla, problemin tanımı, çalışmanın amacı ve kapsamı, çalışmada dikkate alınan varsayımlar, çalışmanın adımları ve çalışmanın orijinal katkıları yer almaktadır.

1.1. Yeni Ürün Tanımı ve Ürün Geliştirme Nedenleri

Yeni ürün geliştirme sürecinin anlaşılabilmesi için öncelikli olarak yeni ürün kavramının iyice anlaşılmış olması gereklidir. Yeni ürün denildiğinde ilk olarak fiziksel nesneler aklımıza gelirken, yeni hizmetlerin ve hizmet paketlerinin de birer yeni ürün olduğu unutulmamalıdır. Yeni iş makinelerinin ve beyaz eşya ya da cep telefonlarının yanı sıra yazılımlar, teknik hizmetler ve servis hizmetleri de yeni ürün kapsamına alınabilir. Literatürde değişik şekillerde tanımlanan ürünlerdeki yenilik için, genel kabul görmüş ortak nokta, pazarda var olan ürünlerden ne kadar farklı olduğuna dayanmaktadır. En genel anlamıyla yeni ürün daha önce pazarda satılmamış veya işletme tarafından ilk defa tasarımı yapılarak üretilen bir üründür. Ayrıca, ürünün tasarımı üzerinde değişiklikler yapılarak üretilen ürünler için de yeni ürün kavramı kullanılmaktadır (Crawford ve Di Benedetto, 2006; Trott, 2002; Akyüz 2007’den). Yeni ürünler, önceki ürün kuşağı ile karşılaştırıldığında malzemesi, bileşenleri, yerine getirdiği işlevler açısından teknolojik farklar gösteren ürünlerdir. Yeni ürünlerdeki farklılaşma, küçük iyileştirmelerden büyük yeniliklere doğru derecelendirilmektedir. Örneğin, yeni bir ürün elde etmek için ürünün herhangi bir karakteristiğini değiştirmek yeterlidir.

Yeni ürün kavramının çok farklı şekillerde tanımı yapılmakla birlikte bu tanımlardaki ortak özellikler olarak literatürde yer alan yeni ürün sınıflandırması şu şekildedir (Ulrich ve Eppinger, 2003; Cooper ve Edgett, 1999; Mahajan ve Wind, 1992; Akyüz 2007’den):

- Buluşlar ve teknolojik gelişmelerden yararlanarak ortaya çıkarılan tamamen yeni ürünler,
- İşletme için yeni, pazar için yeni olmayan ürünler,
- İşletme için eski, pazar için yeni olan ürünler,
- Mevcut ürün üzerinde yapılan iyileştirmeler ve gelişmeler sonucu oluşan ürünler,
- Mevcut ürünün ambalajında veya sunum özelliklerinde yapılan değişiklikler sonucunda oluşan ürünler (Tasarım, fiyat, satış koşulları, ambalaj vb.),
- Mevcut ürünlerin fiziki boyutlarında yapılan değişiklikler sonucu ortaya çıkan ürünler,
- Mevcut ürün(ler)e yeni kullanım alanları bulma,
- Taklit ürünler,
- Bir ürünün yeni nesillerinin üretimi,

Yeni ürün tanımının anlaşılmasından sonra ürün geliştirmenin tanımını verebilir. Yapılan önceki çalışmalarda, birçok tanım yer almaktadır ancak genel olarak, ürün geliştirme, müşterinin ne istediğinin belirlenmesi ile başlayarak; ürünün ortaya çıkması, satışı ve müşteriye sunulması ile son bulan bir dizi faaliyetin yerine getirilmesidir (Chase ve ark., 2006; Brown ve ark., 2004; Akyüz 2007'den). Ürün geliştirme çok çeşitli şekillerde tanımlanmakla birlikte bütün tanımların ortak özelliği, geliştirilen ürünlerin müşteriler için yeni olmasıdır.

Yapılan araştırmalar, ürün geliştirme kavramı (Product Development-PD) ile yeni ürün geliştirme (New Product Development-NPD) kavramının aynı anlamlarda kullanıldığını göstermiştir (Ulrich ve Eppinger, 2003; Belliveau ve ark., 2002; Cooper ve Edgett, 1999; Akyüz 2007'den).

Ürün geliştirme, işletmelerin varlıklarını sürdürebilmeleri için temel unsurdur. Küreselleşen rekabet, geliştirilen yeni teknolojiler, ürün yaşam çevrimlerinin kısalması ve sürekli değişen müşteri talepleri, işletmelerin ürün geliştirme çalışmalarını bir zorunluluk haline getirmiştir (Goetsch ve Davis, 2006; Akyüz 2007'den). Ürün yaşam sürelerinin kısalması, ürün geliştirme çalışmalarının önemini artırmıştır. Yaklaşık 30 yıl önce ürün yaşam süreleri ortalama 10-15 yıl iken, günümüzde ileri teknoloji ürünlerinin ürün yaşam süreleri 1-3 yıl arasında

değişmektedir. Buna bağlı olarak ileri teknoloji ürünleri geliştirme süreleri ortalama 6 aya kadar düşmüştür (Hörte, 2006: Akyüz 2007'den).

İşletmelerin ürün geliştirme çalışmalarına önem verme nedenleri; küresel rekabet, ürün yaşam çevrimin (PLC) kısalması, bilginin yayılma hızı, işletmeler ya da ülkeler arası işbirliği, müşteri odaklılığın artması, küçük projelerin istenen başarıyı getirmemesi, teknolojik ilerlemelerin çok büyük oranda etkili olması, müşteri ihtiyaçlarının değişmesi, pazarın ve rekabetin giderek büyümesi, maliyetlerin artması, şeklinde sıralanabilir (Goetsch ve Davis, 2006; Minderhoud ve Fraser, 2005; Luo ve ark., 2005; Sun ve Wing, 2004; Bobrow ve Shafer, 1987: Akyüz 2007'den).

Ürün geliştirme nedenlerinden bazıları kısaca aşağıdaki gibi gruplandırılarak açıklanabilir:

- **Küreselleşme ve Rekabet:** Dünya ticaretinin artması ile rekabet yoğunlaşmış ve uluslararası rakipler ortaya çıkmıştır. Bu durum mevcut veya yeni pazarlarda her zaman rakipler olmasına neden olmaktadır. Bu nedenle rekabette avantajlı olmak zorunludur. Bunu sağlamanın yolu fark yaratan ürünler geliştirmek ve bunu oldukça hızlı yapmaktır. Rekabetin ulusal olmaktan çıkarak uluslararası hale gelmesi, ürün kalitesinin dünya kalitesi standartlarına ulaşması ve rakiplerin sayılarının artması ürün geliştirme çalışmalarında modern ürün geliştirme araç ve tekniklerinin kullanılması ihtiyacını artırmıştır (Lehmann ve Winer, 2002; Shina, 1994; Wheelwright ve Clark, 1992: Akyüz 2007'den).
- **Firmanın Pazarda Büyüme İsteği:** Bir firmanın büyümesi ancak yeni bir ürünün üretilmesiyle gerçekleştirilebilir. Çünkü mevcut ürünler, zamanla eskir, tüketici ihtiyaçlarına cevap veremez hale gelir. Bir firma, büyümek istiyorsa, yeni ürünler üretmek ve yeni pazarlar bulmak zorundadır. Aksi takdirde uzun dönemde büyüme, yeni ürünler ya da hizmetler pazara sunulmadan gerçekleştirilemez (Gönüllüoğlu, 2005; Gao ve ark.,2000; Kutar ve ark.,2004: Yıldız 2007'den).
- **Ürün Yaşam Çevriminin Kısalması:** Ürün yaşam çevrimi, ürünün satış hacmi ve karının zaman içindeki değişimlerini açıklayan bir kavramdır. Yeni bir ürün pazara girdiğinde satışlar artar ve belli bir süre sonunda giderek azalır. Bunun

sonucunda ekonomik karlılığın sürdürülebilmesi için yeni ürün geliştirme çabaları önem kazanır (Lehman ve Winer, 2002: Akyüz 2007'den).

- ***Firma Kaynaklarının Verimli Şekilde Kullanılması:*** Firmalar, tam kapasite ile üretim yapmak isterler. Ancak, üretilen ürünlere talebin düşük olması, tam kapasite çalışmayı engeller. Firmanın ürettiği ürünlere yeni çeşitler eklemesi, karını arttırırken, sabit giderlerini de azaltması anlamına gelmektedir. Öte yandan, yeni ürünler için yapılacak reklam ve tanıtım faaliyetleri, firmaya ait diğer ürünlerin de ön plana çıkmasına ve hatırlanmasına imkan sağlayacaktır (Gao ve ark.,2000; Kutar ve ark.,2004: Akyüz 2007'den).

1.2. Yeni Ürün Geliştirme Süreci

Ürün geliştirme süreçleri (Product Development Processes); müşteri ve pazar ihtiyaçlarının belirlendiği, yeni kavramların oluşturulduğu ve müşterilere bir değer sunmak amacıyla gerçekleştirilen çalışmaların toplamından oluşan faaliyetlerdir. Ürünlerin karmaşıklığı, rekabetçi çevre, teknolojiye değişim, hangi ürünlerin geliştirileceği gibi birçok faktör ürün geliştirme süreçlerinin işletmelere göre farklılıklar göstermesine neden olmuştur. Bu değişim, farklı seviyelerde farklı ürün geliştirme görevleri ortaya çıkarmıştır (Green ve Bonollo, 2002: Akyüz 2007'den).

Süreçlerin iyi planlanması ürün geliştirme çalışmalarının başarısında önemli olmaktadır. Ürün geliştirme süreçleri; bir fikrin oluşumu ve ortaya çıkmasından başlayarak müşteriye sunulmaya hazır bir değere dönüştürülünceye kadar geçirdiği bütün işlemlerden oluşur. Tasarım süreci ise fikir aşaması ve kavram aşamasını tamamlamış, ürünün kağıt üzerinde ya da model üzerinde görünür hale gelmeye başladığı aşamadır. Ürün, bu aşamada çeşitli işlem adımlarından geçerek bir sonraki süreç olan üretim sürecine doğru yol almaktadır. Ürün geliştirme adımları olarak ürün geliştirme süreçlerinin her birinde yapılan işlemlerin sıralaması belirtilir (Carford ve Di Benedetto, 2006; Green ve Bonollo,2002 Akyüz 2007'den).

Ürün geliştirme süreçlerini ve süreçlerdeki işlemleri genel olarak Tablo 1.1.'de görüldüğü gibi özetlemek mümkündür:

Tablo 1.1. Ürün geliştirme süreçleri (Chase ve ark., 2006; Kahn, 2005: Akyüz 2007'den)

Süreçler	İşlemler
Fikirlerin Oluşturulması	Fikir dikkate değer mi? Geliştirilen ürünlerin temelinde fikir yatar.
Fikirlerin Değerlendirilmesi	Ürün fikri, işletme amaçları, stratejileri ve kaynakları ile tutarlı mı? Toplanan fikirlerin değerlendirilerek uygun olanlarının seçilmesidir.
Kavram geliştirme	Müşterilerin tamam diyeceği iyi bir ürün kavramı var mı? Ürün fikirlerinin, tüketicilerin diliyle anlamlı terimler haline gelmesidir.
Stratejilerini belirleme	Nasıl bir strateji izlenmeli? Ürün geliştirme stratejileri belirlenir.
Ekonomik analiz	Bu ürün işletmenin kar hedeflerini karşılayacak mı? Ürünün işletmenin karlılığına, satışlarına ve maliyetlerine katkıları hesaplanır.
Ürünün Geliştirilmesi	Bu ürünü teknik olarak yapabilir miyiz? Ürün teknik ve pazarlama açısından mükemmel bir ürüne dönüştürülür.
Pazar (pazarlama) test aşaması	Ürünün satışları beklentilerimizi karşılar mı? Ürünün pazar potansiyelini saptanır, satış politikalarının etkinliğini ölçülür.
Pazara Sunum	Ürünü, ne zaman, nasıl, nerede ve kime sunacağız? Ambalaj ve dağıtım özellikleri belirlenir.

- i) **Fikirlerin Toplanması:** Ürün geliştirme çalışmalarının başlangıç noktasını yeni fikirler oluşturur. Birçok yeni fikrin toplanarak bunlar içinden uygun olanların seçilmesi söz konusudur. Yeni ürün fikirleri müşteriler, rakipler, araştırma kurumları gibi birçok kaynaktan gelebilir (Zahay ve ark., 2004; Kotler, 2000 Akyüz 2007'den). Ürün geliştirmede fikirlerin çok olması ve bunlar içinden seçim yapılması başarıyı artıran bir unsurdur. Bu nedenle işletmeler, ürün geliştirme çalışmalarında fikirlerin oluşturulmasına ve bunların toplanmasına önem vermektedir (Reid ve Sanders, 2005: Akyüz 2007'den). Bu fikirlerin, işletmenin genel politikaları ve süreçleri ile uygun olması önemlidir (Gren ve Bonollo, 2002). Fikirlerin büyük bir çoğunluğu işletme içinden gelir. İşletme içi fikir kaynakları; Araştırma Geliştirme departmanı çalışanları, işletme çalışanları, pazarlama bölümü elemanları, teknik uzmanlar ve diğer yöneticilerdir. İşletme dışı fikir kaynakları ise, müşteriler, rakip ürünler,

araştırma kurumları, üniversiteler ve tedarikçilerdir (Akyüz, 2007; Ribbens, 2000).

- ii) Fikir Değerlendirme:** Yeni ürün hakkında toplanan birçok fikrin değerlendirilerek uygun olmayanların elenmesidir. Burada, ürünün pazar durumu, boyutları, rakiplerin durumu, ürün geliştirme maliyetleri, üretim maliyetleri, üretilebilirlik ve ürün geliri gibi birçok faktör dikkate alınarak fikirler değerlendirilir (Kotler, 2000; Akyüz 2007'den). Değerlendirmeler sonucunda uygun görülen fikirler üzerinde kavram geliştirme çalışmalarına başlanır (Crawford ve Di Benedetto, 2006 Akyüz 2007'den)
- iii) Kavram Geliştirme:** Cazip kavramlar geliştirilerek, tecrübe edilebilecek ürün kavramları haline getirilir. Ürünü kimlerin kullanacağı ve bu ürünün sağlayacağı başlıca yararların neler olacağı sorularına cevap oluşturulur (Chase ve ark., 2006; Akyüz 2007'den). Ayrıca bu sorulara verilecek cevaplar ürünün hangi hususta rekabet edeceğini göstermektedir. Daha sonra gerçekleştirilecek adım ise ürün kavramını bir marka kavramına dönüştürmektir (Kotler, 2000; Ribbens, 2000; Akyüz 2007'den).
- iv) Strateji Belirleme:** Bir işletme, ürün geliştirme çalışmalarına başlamadan önce ürün geliştirme stratejilerini belirlemelidir. Ürün geliştirme stratejileri, işletmenin yeni ürünlerinin özelliklerinin ve pazara sunma zamanının belirlenmesini sağlar. Ürün geliştirme stratejileri üzerine yapılan araştırmalara göre; ürün geliştirme stratejileri belirlenirken; yeni ürün geliştirme çalışmalarında yeniliğin ve yaratıcılığın geliştirilmesine, ürün geliştirme stratejilerinin işletme amaçlarına uygun olmasına, rekabet avantajı sağlamasına ve çevreye uyum sağlamada kolaylıklar sağlamasına dikkat edilir (Kumar ve Phrommathed, 2005; Ribbens, 2000; Lamb ve ark., 1996; Akyüz 2007'den).
- v) Ekonomik Analiz:** Fikir aşamasında geliştirilen fikirlerin, yapılabilir olup olmadığını değerlendirmektir. Bu aşamada aynı zamanda, teknolojik ve yapısal araştırma da yapılmaktadır (Sahin, 2005; Akyüz 2007'den). Geliştirilmesi düşünülen ürünlerin; satışları, maliyetleri ve karlılıkları incelenerek yeni ürünlerin, işletmenin talep ve isteklerine uygun olup olmadığı araştırılır. Bu

analizlerin sonucuna göre ürün geliştirme sürecine girilir (Chase ve ark., 2006; Green ve Bonollo, 2002; Akyüz, 2007'den).

vi) Ürün Geliştirme: Ürün kavramı, ekonomik ve ticari analizlerden geçtikten sonra araştırma geliştirme bölümüne gönderilir. Buraya kadar ürün, kelimelerle veya sembollerle tanımlanırken bu noktadan itibaren somut hale gelmeye başlar. Burada hedef, müşteri ihtiyaçlarını geçerli bir modele (protatip) dönüştürmektir. Bu işlem, birçok farklı tekniğin kullanılması ile yapılmaktadır. Bu araç ve teknikler, pazar araştırmaları sonucunda ortaya çıkarılmış müşteri istek ve beklentilerini alarak bunları mühendislerin kullanabileceği mühendislik özellikleri listesine dönüştürmektedir. Bu araç ve teknikler, işletmenin diğer bölümleri ile iletişimi de geliştirmektedir (Green ve Bonollo, 2002; Kotler, 2000; Akyüz 2007'den). Araştırma ve geliştirme bölümünde, ürünün bir veya daha fazla fiziki olarak çeşitlemesi yapılarak ön tasarımı yapılır. Amaç, geliştirilen ürünün ekonomik üretilmesini sağlamak ve müşterilerin beklentilerine cevap verecek kalite ve fiziksel özellikleri kazandırmaktır. Ürünün bir modeli hazırlanır. Model hazırlama esnasında maliyetlerin çok yüksek olduğu görülse de, daha sonra ortaya çıkabilecek olumsuzlukların maliyetlerinden çok değildir (Green ve Bonollo, 2002; Akyüz 2007'den). Ürünü mükemmelleştirmek için çok yoğun kaynak, çaba ve zaman gerekmektedir. Yapılan araştırmalara göre, ürün geliştirme zamanının azaltılmasının anahtarı, kalite için tasarım tekniklerinin daha ileri götürülmesidir. Burada önemli olan nokta ürün geliştirme ekiplerinin, ürün geliştirme çalışmalarında müşteri ihtiyaçlarını mümkün olan en kısa sürede karşılayabilecek bir ürün tasarımına ulaşmalarıdır. İyi bir tasarım ilerleyen süreçlerde ortaya çıkması muhtemel problemleri önlemektedir (Thomkovik ve Miller, 2000; Akyüz 2007'den).

vii) Test: Testlerin amacı, geliştirilen ürünün doğruluğunu test etmektir. Bu amaçla ürünün modeli hazırlanır. Ürünün standartlara uygunluğu test edilir. Pazarlama açısından ürünün test edilmesinde amaç ise, ürünün uzun dönemde beklenen satışları sağlayıp sağlamayacağını belirlemektir (Chase ve ark., 2006; Akyüz

2007'den). Model testleri olarak; alfa testi, beta testi ve gama testi olarak çeşitli şekillerde test yöntemleri kullanılmaktadır (Ribbens, 2000: Akyüz 2007'den).

viii) Pazara Sunum: Bu son aşamada ürün, pazara sunuma hazır hale getirilmiş demektir. Ürün geliştirme süreci yeni ürünün pazara sunumu ile son bulur. Bu bölümden itibaren, pazarlama bölümünün çalışmaları giderek önem kazanır. Ürün pazara firmanın pazarlama stratejilerince uygun olarak sunulur (Fisher, 1998: Akyüz 2007'den).

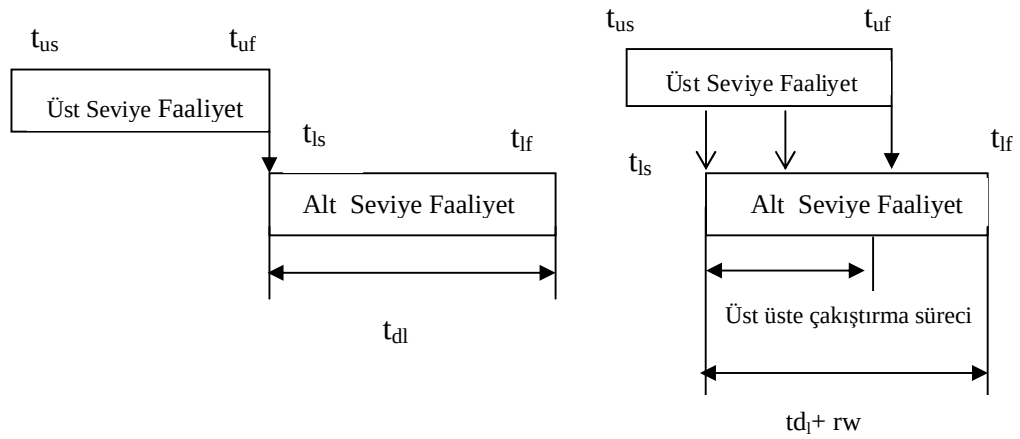
1.3. Yeni Ürün Geliştirme Projelerinde Üst Üste Çakıştırma Uygulaması

Ürün geliştirme, ürün fikrinin ortaya çıkmasından ürünün piyasaya çıkartılmasına kadar birçok aşamaları içermektedir. Bu nedenle çok fazla bilgi kaynaklarından girdiler gerektiren, disiplinler arası bir süreçtir. Sonuç olarak, günümüzün büyük ürün geliştirme projeleri, yüzlerce mühendis, yönetici, teknisyen ve uzmanın yer aldığı karmaşık faaliyetler dizinini içermektedir. Bu durumda, yeni ürün geliştirme projelerinin en önemli özelliği bazı faaliyetler arasında bilgi transferleri şeklinde ilişkiler bulunmasıdır (Chen ve ark., 2003).

Yeni ürün geliştirme projelerindeki bazı faaliyetler arasındaki bilgi transferi ilişkisinin olması proje bitiş süresinin kısaltılabilmesi amacıyla kullanılabilecek özel uygulamaların geliştirilebilmesine olanak sağlamıştır. Rekabetin oldukça yoğun olduğu günümüzde, ürün geliştirme süresinin kısaltılması firmalar açısından kritik öneme sahiptir. Ürün tasarım süresinin önemi, Nichols (1990) tarafından yapılan çalışma ile oldukça net bir şekilde gösterilmiştir. Bu çalışmaya göre ürün geliştirmede yaklaşık %50 daha fazla harcama yapılması durumunda kardaki kayıp %4 iken eğer ürün pazara 6 ay daha geç çıkartılırsa bu kayıp karın yaklaşık üçte biri kadar olmaktadır. Bu durumda ürün geliştirme sürecini hızlandırmak amacıyla eş zamanlı mühendislik yaklaşımına dayalı bir uygulama olan faaliyetlerin üst üste çakıştırılması uygulamasının ortaya çıkmasına neden olmuştur.

Geleneksel seri tasarım yaklaşımında, alt seviye faaliyetlere ancak bu faaliyetin gerçekleştirilmesi için tam bilgiye sahip olunduğunda başlanmaktadır. Eş

zamanlı yaklaşımda ise bir faaliyet kendinden önce gelen faaliyete ait temel yada kısmi bilgi ile başlatılmakta ve sonrasında üst seviye faaliyetten bilgi geldikçe güncellenmektedir. Bu durum eş zamanlı mühendislikte “üst üste çakıştırma (task overlapping)” olarak tanımlanmıştır. Faaliyetlerin üst üste çakıştırılması Şekil 1.1. ile gösterilmiştir;



Şekil 1.1. Faaliyetlerin üst üste çakıştırılması

Burada;

t_{us} : Üst seviye faaliyetin başlama zamanı

t_{uf} : Üst seviye faaliyetin bitiş zamanı

t_{ls} : Alt seviye faaliyetin başlama zamanı

t_{lf} : Alt seviye faaliyetin bitiş zamanı

t_{dl} : Alt seviye faaliyetin süresi

rw : Üst üste çakıştırmadan kaynaklanan yeniden çalışma miktarı

Bu yöntemin proje süresini kısaltıcı etkisi olmasına rağmen sürecin çok iyi planlanmaması durumunda ters yönde etki yaparak süreyi uzatabileceği ve aynı zaman da tasarımın kalitesini düşürebileceği bilinmektedir. Bunun nedeni ise, kısmi ve yetersiz bilgi ile bir faaliyetin başlatılmasının riskli bir durum olmasıdır. Yeterli olmayan bir bilgi ile çok çabuk başlatılan veya çok hızlı ilerletilen bir alt seviye faaliyete, zaman içerisinde üst seviye faaliyetten gelen yeni bilgiler neticesinde güncelleme ve faaliyetin yeniden yapılması çok zaman alabilir ve bu durumda

faaliyetin süresinin uzamasına neden olabilir. Özellikle eş zamanlı ürün geliştirmenin kullanıldığı durumlarda bir faaliyetin süresindeki uzama ise domino etkisi yaparak tüm projenin bitişini çok önemli oranda etkileyebilir. Bu nedenle, üst üste çakıştırma sürecinin çok iyi planlaması gerekir (Abdelselam, 2003). Burada önemli olan, ne zaman ve nasıl (ne miktarda) üst üste çakıştırmanın avantajlı olacağının belirlenmesidir.

1.4. Problemin Tanımı

Çalışmada ele alınan problem, Yeni Ürün Geliştirme (YÜG) projelerinde kaynak kısıtlarını ve üst üste çakıştırma uygulamasını içeren bir çizelgeleme problemidir. Önceki çalışmalar incelendiğinde problemin iki şekilde ele alındığı görülmüştür. İlk grupta yer alan araştırmacılar, yeni ürün geliştirme sürecinin diğer projelerde olduğu gibi birbirleriyle ilişkili faaliyetlerden meydana geldiği ve bu faaliyetlerin gerçekleştirilebilmesi için sınırlı miktarda kaynağın mevcut olduğunu vurgulamışlardır. Özetle bu gruptaki çalışmalar ele aldıkları bu özellikler nedeniyle problemi “kaynak kısıtlı proje çizelgeleme problemi” şeklinde tanımlamışlar ve matematiksel olarak modellemişlerdir (Wang, 1999; Wang, 2002; Wang, 2004; Balasubramanian ve Grossman, 2002). Yeni ürün geliştirme projelerine özgü özellikler olarak ise sadece faaliyet sürelerindeki belirsizlikler üzerinde durmuşlardır. Bahsedilen belirsizliği ise bir kısım araştırmacılar, stokastik olarak modele dahil ederken diğer bir kısım araştırmacılar belirsizliğin bulanık modelleme kullanılarak temsil edilmesinin daha doğru olacağını belirtmişler ve faaliyet sürelerindeki belirsizliği modele bulanık olanak dahil etmişlerdir.

Diğer grupta yer alan araştırmacılar ise yeni ürün geliştirme sürecinin modellenmesinde sisteme özgü olarak nitelendirdikleri birtakım uygulamaların dikkate alınması gerektiğini vurgulamışlardır. Bu çalışmalarda, iyi bir pazar payı elde edebilmek için, firmaların yeni ürün geliştirme projelerini etkin bir şekilde yönetmesi gerektiğini ve yeni ürünü mümkün olduğu kadar kısa süre içerisinde pazara sunması gerektiğini vurgulanmıştır (Krishan ve ark.,1997; Roemer ve ark.,

2000; Lin ve ark.,2008). Bu nedenle, YÜG projelerinde, proje bitiş süresinin kısaltılmasını sağlayabilecek bir takım yeni araçlar kullanılmaya başlanmıştır. Bunların içlerinde en yaygın olarak kullanılanı “faaliyetlerin üst üste çakıştırılması (task overlapping)” olarak bilinen uygulamadır. Bu uygulamada, bazı alt seviye faaliyetler kendinden önce bitirilmesi gereken ve kendisine bilgi sağlayacak üst seviye faaliyetin bitirilmesi beklenmeden, üst seviye faaliyetten alınan ön bilgi ile başlatılmaktadır. Sonrasında ise, üst seviye faaliyetten yeni bilgi geldikçe alt seviye faaliyet bu bilgileri güncellemek için yeniden çalışılmaktadır (rework). Üst üste çakıştırma, iyi planlandığı takdirde yeni ürün geliştirme projelerinin sürelerini önemli ölçüde azalttığı bazı deneysel çalışmalar ile gösterilmiştir (Smith ve Reinersten,1995; Helms, 2004). Bu amaçla, bir kısım araştırmacılar optimal üst üste çakıştırma miktarı ve bilgi transferi zamanlarını belirleyen modelleme yaklaşımları üzerinde çalışmışlardır (Krishan ve ark.,1997; Roemer ve ark, 2000; Lin ve ark.,2008). Ancak bu yaklaşımlar genellikle çizelgelemeden bağımsız olarak ele alınmıştır.

Modern yeni ürün geliştirme projeleri birbirleriyle öncelik ilişkileri bulunan yüzlerce faaliyet içerirler ve her bir faaliyet kendine özgü bir süre zarfında birçok kaynağa (mühendis, laboratuvar) ihtiyaç duyar. Fakat her bir kaynak için bir kısıt söz konusudur. Bu nedenle, kısıtlı kaynakların belirli bir süre boyunca faaliyetlere tahsis edilmesini de içeren etkin bir proje çizelgeleme, kaynakların verimli kullanımı ve sürenin azaltılması açısından oldukça önemlidir. Ele alınan problem, klasik kaynak kısıtlı proje çizelgeleme probleminden farklı olarak, iyi planlandığı takdirde YÜG projelerinin bitiş süresinin kısaltılmasını sağladığı bilinen faaliyetlerin üst üste çakıştırılması uygulamasını da içeren bir çizelgeleme ve kaynak tahsisi problemidir. Bu bağlamda, yapılan çalışmada, yeni ürün geliştirme projelerinin çizelgelemesinde gerçek hayat problemlerine uygun çözümler bulunabilmesi için projede mevcut olan kaynak kısıtlarının dikkate alınmasının yanı sıra faaliyetlerin üst üste çakıştırılması uygulamasını da içeren bir problem tanımlanmıştır. Ele alınan problem, kaynak kısıtlarını ve faaliyetler arasındaki teknik kısıtları dikkate alarak proje süresini en küçükleyecek şekilde çizelge oluştururken aynı zamanda proje süresini en küçükleyecek şekilde, belirlenen faaliyetler arasındaki en uygun çakıştırma

miktarının hesaplanmasını da içermektedir. Faaliyetler arasında üst üste çakıştırma yapılacak miktarın uygun bir şekilde belirlenmesi, oluşacak yeniden çalışma miktarının dengelenmesi ve üst üste çakıştırmadan elde edilecek kazancın en büyüklenmesi açısından önemlidir. Aksi takdirde, yeniden çalışma miktarı çok fazla yapılırsa ve projenin bitiş süresini kısaltma yerine tam tersi bir etki yapabilmektedir. Bütün bunlar dikkate alındığında aslında problem, klasik kaynak kısıtlı proje çizelgeleme probleminden farklı olarak en uygun üst üste çakıştırma miktarının belirlenmesi gibi bir karar durumunu daha içermektedir.

1.5. Çalışmanın Amacı ve Kapsamı

Rekabetin yoğun olduğu günümüzde ürün geliştirme sürecinin kısa olması nedeniyle ürünün mümkün olduğu kadar önce piyasaya sunulması oldukça önemli bir avantaj sağlamaktadır. Bu nedenle, tasarım süreçlerinin hızlandırılması için, ürün geliştirme projelerindeki faaliyetlerin üst üste çakıştırılması yaklaşımı yaygın bir şekilde kullanılmaktadır.

Üst seviye faaliyet hakkında tam bilgiye sahip bir ekip tarafından alt seviye faaliyetlerin gerçekleştirildiği geleneksel seri yaklaşımdan farklı olarak eş zamanlı mühendislik yaklaşımının kullanıldığı bu uygulamada bir faaliyet üst seviye faaliyetten elde edilen ön ve eksik bilgi ile başlatılabilir. Başlangıçtaki bilgi, üst seviye faaliyet gerçekleştirildikçe yenilenmekte ve alt seviye faaliyete iletilmektedir. Bu nedenle, alt seviye faaliyette zaman içerisinde bir takım yeniden çalışmalar (rework) gerçekleştirilmek zorunda kalınabilmektedir (Eastman, 1980).

Bu durumlar göz önüne alındığında, başlangıçtaki bilgiler kullanılarak bir faaliyete başlanmasının doğal olarak bir takım risklerde içerebileceği anlaşılabılır. Eğer bir alt seviye faaliyet yetersiz bilgi ile çok erken başlatılıp çok hızlı ilerletilirse, bilgi güncellendikçe bu faaliyette yapılması gereken güncelleme işleminin de süresi çok uzayabilir. Bu durumda, proje süresinin kısaltılması için kullanılan bu üst üste çakıştırma uygulaması ters yönde bir etki yaparak sürenin uzamasına neden olabilir.

Gecikmelerden kaçınmak ve projenin zamanında bitirilmesini sağlamak amacıyla üst üste çakıştırma sürecinin çok iyi planlanması gerekir.

YÜG projelerinin çizelgelenmesinde dikkate alınması gereken diğer önemli unsurlar ise proje kaynaklarının sınırlı olması ve ürün geliştirme projelerinin diğer projelerden farklı olarak birbirleriyle bilgi alışverişi ve iletişim içersinde olan çok fazla sayıda faaliyetten oluşmasıdır.

Yukarıda belirtilen durumlar göz önüne alınarak, iyi planlandığı takdirde yeni ürün geliştirme projelerinin bitiş süresinin kısaltılmasını sağlayan üst üste çakıştırma uygulamasını da içeren bir çizelgeleme ve kaynak tahsisi modelinin geliştirilmesi amaçlanmaktadır. Analitik modelde istenilen, faaliyetler arasındaki öncelik ilişkilerini, üst üste çakıştırma durumunu ve kapasite kısıtlarını dikkate alarak proje bitiş süresinin en küçüklenmesidir. Daha sonrasında ise, ele alınan problemin çözüm karmaşıklığının olması nedeniyle, önerilen analitik modelin gerçek hayatta karşılaşılan problemler için yetersiz kalması durumu dikkate alınarak büyük çapta problemler için makul sürelerde en iyiye yakın çözüm veren bir sezgisel algoritmanın geliştirilmesi amaçlanmaktadır.

Ele alınan problemin temel özellikleri ve kapsamı aşağıdaki gibi maddeler halinde açıklanabilir;

1. Çok sayıda faaliyetlerden oluşması,
2. Faaliyetler arasında öncelik ilişkilerinin olması,
3. Faaliyetlerin gerçekleştirilmesi için gerekli olan kaynakların kısıtlı olması,
4. Çoklu yenilenebilir kaynak içermesi,
5. Belirli faaliyetlerin üst üste çakıştırılması uygulamasını içermesi.

1.6. Varsayımlar

Yapılan çalışmada aşağıdaki varsayımlar esas alınmıştır;

1. Amaç proje süresinin en küçüklenmesidir.
2. Faaliyetlerin ihtiyaç duyduğu kaynak miktarı belirli ve sabittir.
3. Faaliyetlerin süreleri belirli ve sabittir.

4. Faaliyetlerin ihtiyaç duydukları kaynaklar yenilenebilir kaynaklardır (insan, laboratuvar, bilgisayar, yazılım, vs.).
5. Çoklu yenilenebilir kaynaklar ve kaynak limitleri söz konusudur.
6. Üst üste çakıştırma yapılırken alt seviye faaliyet en erken, üst seviye faaliyetten bir zaman birimi sonra başlatılabilir.
7. Üst üste çakıştırma yapılırken alt seviye faaliyet üst seviye faaliyet bitmeden bitemez.

1.7. Çalışmanın Adımları

Çalışmanın oluşturulmasında sırasıyla aşağıdaki adımlar izlenmiştir;

1. Çalışmada ele alınan problemin tanımlanması,
2. Literatürde yapılan önceki çalışmaların incelenmesi,
3. Matematiksel modelin kurulması,
4. Modelin analitik olarak çözülebilmesi için uygun çözüm yöntemlerinin araştırılması,
5. Örnek veri setleri kullanılarak analitik modelin performansının test edilmesi,
6. Farklı büyüklükteki problemlerin makul sürelerde ve en iyiye yakın çözüm verecek şekilde çözülebilmesi için uygun ileri sezgisel metodun araştırılması ve probleme özgü algoritmanın bu sezgisele dayalı olarak geliştirilmesi,
7. Algoritmanın C# programlama dilinde kodlanması,
8. Farklı parametrelere sahip veri setleri kullanılarak (proje büyüklüğü, kaynak kısıtlılığı ve proje ağı karmaşıklığı faktörüne göre) algoritmanın test edilmesi,
9. Senaryo analizlerinin yapılması ve bir yeni ürün geliştirme projesinde uygulanması,
10. Çalışma sonucunda ortaya çıkan sonuçların özetlenmesi ve daha sonraki dönemlerde yapılacak çalışmalara yönelik önerilerde bulunulması.

1.8. Çalışmanın Orijinal Katkısı

Ele alınan problemle ilgili daha önce yapılan çalışmaların incelenmesi neticesinde “Yeni ürün geliştirme projelerinin çizelgelenmesinde, problemin yalnızca “kaynak kısıtlı proje çizelgeleme” problemi olması üzerine odaklanılmış olduğu ve problemin bu şekilde matematiksel olarak modellendiği görülmektedir. İncelenen önceki çalışmalarda, ürün geliştirme projelerine ait spesifik bir özellik olarak, yalnızca tasarım süreci olmasından kaynaklanan belirsizlikler vurgulanmakta ve modellemede bu belirsizlikler dikkate alınmaktadır. Fakat ürün geliştirme sürecinin analiz edildiği çalışmaların incelenmesi neticesinde ürün geliştirme projelerinde sürenin mümkün olduğu kadar kısaltılmasının önemli olduğu ve bu amaçla bir takım uygulamalar geliştirildiği görülmüştür. Bu uygulamalar arasında yer alan faaliyetlerin üst üste çakıştırılması, sıklıkla kullanılan bir yöntem olarak dikkati çekmektedir. Ancak, bu uygulamanın proje süresini kısaltması için oldukça iyi bir şekilde planlanması gerekir.

Bu çalışmada, proje süresinin kısaltılması için kullanılan uygulamalardan biri olan “faaliyetlerin üst üste çakıştırılması (overlapping)” süreci çizelgeleme aşamasına dahil edilmiştir. Üst üste çakıştırma uygulaması ile proje bitiş süresinin kısaltılmasını sağlayabilmek için, uygun üst üste çakıştırma miktarının belirlenmesi çok önemlidir. Aksi takdirde, alt seviye faaliyetin çok erken başlatılması yani üst üste çakıştırmanın çok fazla yapılması durumunda meydana gelecek fazla yeniden çalışma miktarı proje bitiş süresinin kısaltılmasını engelleyebilir hatta proje bitiş süresinde uzamaya bile neden olabilir. Önerilen model ile, kısıtlı kaynaklar ve faaliyetler arasındaki öncelik ilişkileri dikkate alınarak, proje süresini en küçükleyecek şekilde projenin çizelgelenmesi gerçekleştirilirken aynı zamanda önceden belirlenen faaliyetler arasındaki uygun üst üste çakıştırma miktarı da amaca uygun olarak belirlenmektedir.

Ayrıca, üst üste çakıştırma içerecek şekilde uyarlanmış yani probleme göre revize edilmiş seri çizelgeleme yaklaşımını kullanan “Parçacık Sürü Optimizasyonu” algoritması geliştirilerek büyük çapta problemlerin çözümü de makul süreler içinde

gerçekleştirilebilmektedir. Sonraki adımda, geliştirilen mevcut algoritma birden fazla faaliyet çifti arasında üst üste çakıştırmaya imkân verecek şekilde genişletilmiştir. Bu sayede, farklı üst üste çakıştırma alternatifleri birlikte değerlendirilebilmekte ve daha fazla süre kazancı elde edilebilmektedir.

Bunlara ek olarak, incelenen önceki çalışmalarda, KKPÇP'nin çözümü için önerilen PSO prosedürlerinin genellikle küçük ve belirli bir sistematığı olmayan test problemleri ile test edildiği görülmüştür. Bu çalışmada, farklı kaynak kısıtlılığı ve proje ağı karmaşıklığı derecelerine göre sınıflandırılmış ve literatürde kıyaslama verisi olarak kullanılan, 30-60-90 ve 120 faaliyetli farklı proje ağıları algoritmanın etkinliğinin gösterilebilmesi amacıyla kullanılmıştır. Bu sayede daha iyi bir analiz yapmak ve daha net bir şekilde algoritmanın performansını ortaya koymak mümkün olmuştur.

Son olarak, geliştirilen PSO temelli algoritma için yazılan program kullanıcı kolaylığı sağlayan bir ara yüze sahip olarak tasarlanmıştır. Bu sayede, program hakkında teknik olarak her hangi bir bilgiye sahip olunmasına gerek kalmadan kullanıcı tarafından proje verileri rahatlıkla girilebilmekte ve program çalıştırılabilmektedir.

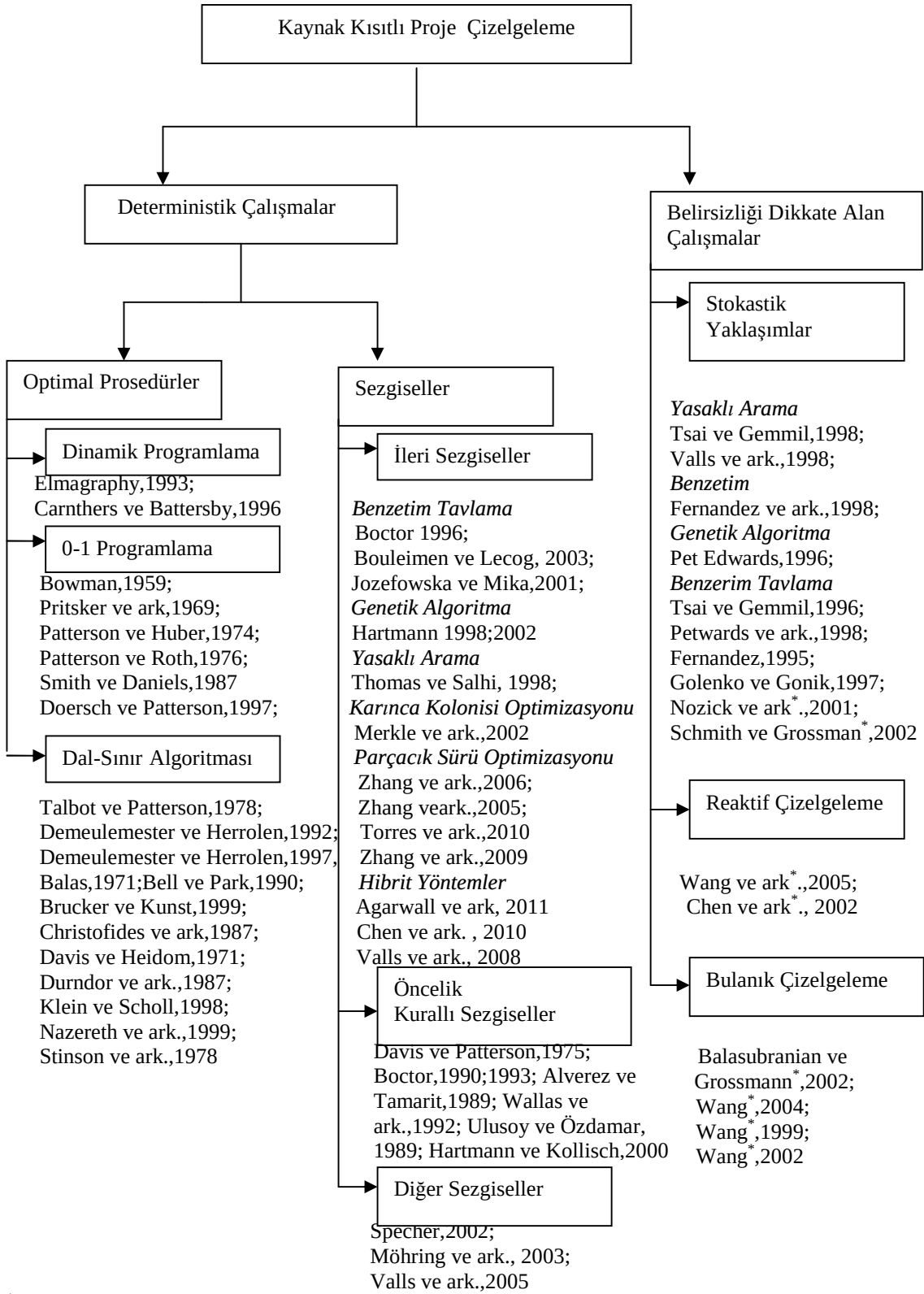
2. ÖNCEKİ ÇALIŞMALAR

Yeni ürün geliştirme projelerinin çizelgelenmesi ile ilgili çalışmaların araştırılması neticesinde, problemin genel olarak Kaynak Kısıtlı Proje Çizelgeleme Problemi (KKPÇP) olarak ele alındığı görülmüştür. KKPÇP’de, birçok farklı amaç fonksiyonu dikkate alınabilir ancak YÜG projelerinde ürünün mümkün olan en kısa süre içerisinde piyasaya sunulmasının çok önemli olması nedeniyle “proje bitiş süresinin en küçüklenmesi” yapılan çalışmalarda geçerli bir amaç olarak kabul görmektedir. Bu nedenle literatür taramasına zaman bazlı amaç fonksiyonuna sahip KKPÇP’ne ait çalışmaların araştırılmasından başlanmış ve sonrasında, yeni ürün geliştirme projelerinin çizelgelenmesi başlığı altında ele alınan çalışmalar incelenmiştir. Son olarak da, yeni ürün geliştirme projelerinde proje zamanının kısaltılmasına yönelik yapılmış bazı kavramsal ve matematiksel modellenme çalışmaları araştırılmıştır.

Bu nedenle bu bölümde öncelikle zaman bazlı amaç fonksiyonuna sahip kaynak kısıtlı proje çizelgeleme çalışmaları özetlenmeye çalışılacak ve sonrasında yeni ürün geliştire projelerinin çizelgelenmesi ile ilgili çalışmalar incelenecektir. Son kısımda ise, yeni ürün geliştirme süreçlerine özgü özelliklerin ve uygulamaların daha iyi anlaşılması amacıyla YÜG süreçlerinin modellenmesi ve analizi ile ilgili çalışmalar açıklanacaktır.

2.1. Kaynak Kısıtlı Proje Çizelgeleme Çalışmaları

Şekil 2.1’de zaman bazlı kaynak kısıtlı proje çizelgeleme problemleri çalışma alanlarına göre sınıflandırılmıştır. Yapılan bu çalışmalar ilgili alt bölümlerde detaylı olarak açıklanmaktadır.



*Yeni Ürün Geliştirme Projelerinin Çizelgelenmesi ile ilgili çalışmalar.

Şekil 2.1. Kaynak kısıtlı proje çizelgeleme problemleri ile ilgili çalışmalar.

2.1.1. KKPC ile İlgili Belirsizlik İçermeyen Çalışmalar

Bu bölümde kaynak kısıtlı proje çizelgeleme problemlerinde belirsizliğin dikkate alınmadığı çalışmalar, kullandıkları çözüm yöntemlerine göre sınıflandırılmış ve alt başlıklar halinde açıklanmıştır.

2.1.1.1. Kesin Çözüm Yöntemlerini Kullanan Çalışmalar

Proje süresinin en aza indirilmesi amacına sahip kaynak kısıtlı proje çizelgeleme problemi “çözüm karmaşıklığı olan bir problem (*NP-hard in strong sense*)” olarak tanımlanmıştır (Garey ve ark., 1976; Blazewicz ve ark., 1983; Klein 2000’den). Bu nedenle, kesin çözüm yöntemleri ancak belirli problem büyüklükleri için geçerli olabilmektedir. Önceki çalışmalarda yer alan, ilk kesin çözüm yöntemleri arasında 0-1 tamsayı programlama yöntemi yer almaktadır (Bowman, 1959; Pristker ve ark., 1969; Patterson ve Huber, 1974; Patterson ve Roth, 1976; Doersch ve Patterson, 1997; Smith-Daniels, 1987). Diğer kesin çözüm yöntemleri ise; dinamik programlama (Elmaghraby, 1993; Carruthers ve Battersby, 1966) ve dal-sınır algoritmasından (Talbot ve Patterson, 1978; Demeulemester ve Herrolen, 1992; Balas, 1971; Bell ve Park, 1990; Christofides ve ark., 1987; Davis ve Heidorn, 1971; Stinson ve ark., 1978) oluşmaktadır.

Oldukça fazla sayıda çalışmada dal-sınır (*branch and bound*) yöntemi kullanılmaktadır. Çalışmalar, dal-sınır algoritmasının “kesin sayım” (implicit enumeration) prosedürünü kullanılmaktadır. Yapılan literatür çalışmalarında kesin sayım algoritmalarındaki farklılıklar dallandırma (branching) ve budama (prune) kurallarındaki farklılıklardan kaynaklanmaktadır.

Davis ve Heidorn (1971), kaynak kısıtlı proje çizelgelemede birden fazla kaynak olması durumu için sınırlı sayım temelli optimizasyon (bounded enumeration based optimization) metodu geliştirmişlerdir. Stinson ve ark. (1978) ise, tek kaynaklı projeler için kısmi (partial) çizelgenin oluşturulmasında genişletilmiş alternatifler kullanan dal-sınır algoritması kullanmıştır. Talbot ve Patterson (1978), öncelik ilişkilerini dikkate alarak faaliyetleri bir faaliyet listesi içerisinde sıralamıştır. Daha sonra, $t=0$ anından ileri özyineleme ile ve projenin bitişi için en geç süreden geri özyineleme ile bütün faaliyetler için zaman aralığı türetmişlerdir. Bell ve Park (1990), faaliyetlerin detaylı

çizelgelemesinden ziyade kaynak çatışmasının ortadan kaldırılmasına yönelik bir yöntem önermişlerdir. Önerilen yöntemde, öncelikle kaynak çatışması yaşanan faaliyetler tespit edilmekte ve sonrasında bu faaliyetlerin sıralanmasını sağlayacak öncelik kısıtları tanımlanarak kaynak çatışması çözülmektedir. Demeulemester ve Herrolen (1992) tarafından geliştirilen algoritma ise, Christofides ve ark. (1987) tarafından geliştirilen “Derinlik Öncelikli (Depth First) Dal Sınır” algoritmasının geliştirilmiş halidir. Daha sonraki çalışmalarında, bu algoritmalarını revize etmişler ve “derinlik öncelikli” metodu, “en iyi arama” metodu ve “hibrit arama” metodu ile karşılaştırmışlardır. Sonuç olarak, “derinlik öncelikli” metodunun diğer iki metottan daha kısa sürede çözüm verdiğini göstermişlerdir (Demeulemeester ve Herroen,1997).

Bu yöntemler, özellikle kısıtlı kaynak durumlarında, küçük ölçekli test problemlerinde optimum çözümü verebilmekte, ancak karmaşık problemlerin çözümünde yetersiz kalmaktadır. Problemin yapısı gereği, faaliyet sayıları ve kaynak kısıtlılığı arttıkça çözüm süresinin çok hızlı bir şekilde artması nedeniyle, optimal prosedürler ile çözüm yaklaşımlarının yanı sıra büyük problem boyutları için sezgisel yaklaşımlar ortaya konmaya başlanmıştır.

2.1.1.2. Sezgisel Yaklaşımların Kullanıldığı Çalışmalar

Sezgisel algoritmalar, büyük olasılıkla en iyiye yakın veya önceden tespit edilmiş değerlendirme ölçütlerine göre, yeterli sayılabilecek çözümleri sağlayan yöntemlerdir. Kaynak kısıtlı proje çizelgelemede, projenin büyüklüğü ve şebekenin özellikleri problemin kesin çözüm veren yöntemler ile çözülmesini engelleyebilmektedir. Bu durumda etkin ve güvenilirliği yüksek, basit programlama kuralları kullanan sezgisel algoritmalar devreye girmiştir. Bu sebeple, matematiksel yöntemler, en iyi çözümün mutlaka sağlanması amacıyla geliştirilmesine karşılık, sezgisel yöntemlerde amaç, etkin ve güvenilirliği yüksek olan en iyi çözüm stratejilerinin geliştirilmesidir.

Genel olarak problemin çözümü için geliştirilen sezgiseller iki ana başlık altında toplanabilir; öncelik kuralı temelli X-Geçiş (X Pass) sezgiselleri, ve ileri sezgisellerdir.

İleri sezgiseller, tabu arama, genetik algoritma, benzetim tavlama, karınca kolonisi ve parçacık sürüsü optimizasyonu gibi sezgisellerden oluşmaktadır.

Öncelik kuralı temelli sezgiseller, paralel veya seri çizelgeleme planlarını kullanmaktadırlar. Bu sezgiseller, tekli geçiş (single-pass) ve çoklu geçiş (multiple-pass) metotlarını içermektedir (Hartmann ve Kolisch, 2000). Tekli geçişli öncelik kuralları, belirli tek bir değeri en büyükleyecek veya en küçükleyecek şekilde faaliyetlerin önceliklerini belirler. Literatürdeki birçok sezgisel bu şekildedir. Bu sezgiseller, öncelik kuralları kullanarak çizelge oluşturan algoritmalar ve tek bir kere uygulanarak elde edilen çözüm nihai çözüm olarak kabul edilir.

Literatürde çok sayıda sezgisel öncelik kuralı önerilmiştir ve bu kuralların performansına ilişkin olarak da çeşitli değerlendirmeler yapılmıştır. Bu kurallardan nispeten daha çok kullanılanları, literatürdeki kısaltmaları da kullanılarak şöyle sıralanabilir; en kısa işlem süresi (*shortest processing time - SPT*), en kısa bolluk süresi (*minimum slack - MINSLK*), en çok sayıda toplam ardıl (*most total successor - MTS*), en büyük kaynak talebi (*greatest resource demand - GRD*), en geç bitirme zamanı (*latest finish time - LFT*), en geç başlama zamanı (*latest start time - LST*), kaynak çizelgeleme metodu (*resource scheduling method - RSM*), en büyük konumsal ağırlık (*greatest rank positional weight - GRPW*), en kötü alternatif bolluk (*worst case slack - WCS*), yerel kısıt tabanlı analiz (*local constraint based analysis - LCBA*).

Davis ve Patterson (1975) çalışmaları neticesinde “en kısa bolluk süresi” sezgisel metodunun optimal toplam proje bitiş süresinde %5.6 sapma ile en iyi sezgisel olduğunu ve “en geç bitirme zamanı” ile “kaynak çizelgeleme” sezgisellerinin bunu takip ettiğini belirtmişlerdir. Buna benzer olarak, Patterson (1976) çalışmasında, “en kısa bolluk süresi” ve “en geç bitirme zamanı” sezgisellerinin, faaliyet başına daha yüksek aylak zaman olduğu durumlarda proje bitiş süresini en küçüklemede daha etkin olduğunu göstermiştir. Cooper (1976), öncelik kuralları kullanan iki tane sezgisel önermiştir. İlk metot, çizelge oluşturma şeması olarak paralel çizelgeleme metodunu kullanmaktadır. İkinci sezgisel ise, rasgele yöntemlerle oluşturulmuş olurlu çözümleri kullanan örnekleme metoduna dayanmaktadır.

Ulusoy ve Özdamar (1989), “Ağırlıklandırılmış kaynak kullanımı ve öncelik (WRUP)” adını verdikleri bir sezgisel geliştirmişlerdir. Bu yöntemde, öncelik ilişkileri

ve kaynak durumu dikkate alınarak faaliyetlere öncelik atanmaktadır. Yöntemlerini, en kısa bolluk süresi, en geç bitirme zamanı, kaynak çizelgeleme metodu, en büyük kaynak talebi ile karşılaştırmışlar ve önerdikleri sezgiselin 111 test probleminde bilinen en iyi sezgisel olan en kısa bolluk süresinden %2.04 daha iyi sonuç verdiğini göstermişlerdir. Drexl ve Grünewald (1993)'de yanlı örnekleme yöntemini önermişlerdir. Özdamar ve Ulusoy (1994), yerel kısıt tabanlı analiz algoritmasının çok modlu durum için uyarlamasını sunmuşlardır. En kısa bolluk süresi, en geç bitirme zamanı, ağırlıklandırılmış kaynak kullanımı ve öncelik sezgisel kuralları sezgisellerini, yerel kısıt tabanlı analiz ile 95 problem üzerinde karşılaştırmışlar ve yerel kısıt tabanlı analizin çok daha üstün bir performans sergilediği gösterilmiştir. Özdamar ve Ulusoy daha sonraki çalışmalarında da (1995; 1996a; 1996b), yerel kısıt tabanlı analiz yinelemeli uygulamasını ayrıntılı olarak irdemişlerdir. Boctor (1996), karar kümesi içinden bir baskın çizelgelenebilir faaliyetler kümesi tanımlayarak; karar kümesinden bir faaliyet yerine, tanımlanan kümenin faaliyetlerinin atandığı bir yöntem önermiştir. Sprecher ve Drexl (1998) kendi dal-sınır yöntemlerinin uygulanmasına bir zaman sınırı koyarak, sezgisel bir çözüm yöntemi olarak kullanılmasını önermişlerdir.

Çok geçişli sezgiseller ise, birden fazla öncelik kurallarını içeren yaklaşımlardır (Lova ve Tormos, 2001). Bu yaklaşımlarda, rasgele bir şekilde bir öncelik kuralı seçilerek uygulanmakta ve bu işlem birden fazla kez gerçekleştirilmektedir (Hartmann ve Kollisch, 2000). Öncelik kurallarına dayalı çizelgeleme kullanılarak ve birden çok geçiş (*multi pass*) uygulanarak, birden çok çizelge türetmek için çeşitli yollar denenmiştir. Bunlardan bir tanesi, türetilen her çizelge için ayrı bir öncelik kuralı uygulanmasıdır. Boctor (1990), yedi değişik sezgisel kural uygulamıştır. Bu yaklaşımlardan farklı olarak Ulusoy ve Özdamar (1989) tarafından sezgisel öncelik kuralı önerilmiştir. Ağırlıklandırılmış kaynak kullanım oranı ve öncelik (*weighted resource utilization ratio and precedence*) olarak tanımlanmış olan bu sezgisel kuralda, ağırlık parametresine $\{0, 1\}$ aralığında değişik değerler atanarak çeşitli çizelgeler elde edilmektedir. Hem seri, hem de paralel çizelgeleme uygulayarak çözüm yapabilen birden çok geçişli bir diğer algoritma da Kollisch ve Drexl (1996) tarafından önerilmiştir.

Bu uygulamaların yanı sıra, KKPÇP'nin çözümü için farklı ileri sezgisel metotlar da kullanılmıştır. Bu sezgiseller, Genetik Algoritma (GA), Yasaklı Arama, Benzetim Tavlama, Karınca Kolonisi, Parçacık Sürü Optimizasyonu (PSO) sezgiselleridir. Meta sezgisellerin kullanımının en önemli nedeni, çözüm karmaşıklığı olan bu problemlerin çözümlerinin makul süreler içerisinde optimuma yakın bir şekilde elde edilebilmesidir.

Bu sezgisel teknikler bazı doğal oluşum süreçlerini taklit etmektedirler. Tavlama benzetimi, termo-dinamik programlamaları taklit etmekte, tabu arama bir çeşit hafıza uygulamasını dikkate alarak “zeki” programları taklit etmekte, Genetik Algoritma ise genetik yapıya benzer şekilde problemleri formüle ederek “doğanın en iyi olanı hayatta kalır” prensibini taklit etmektedir.

Kaynak kısıtlı proje çizelgeleme uygulamalarının çözümünde, GA'nın operatörleri üzerinde değişiklikler yapılmak suretiyle farklı algoritmalar geliştirilmiştir. Lee ve Kim (1996) önerdikleri GA'da, rastlantısal anahtar gösterimi ile birlikte, tek noktadan çaprazlama (*one-point crossover*) kullanmışlar ve paralel çizelgeleme uygulamışlardır. Leon ve Ramamoorthy (1995) ise rastlantısal anahtar gösterimi ve tek noktadan çaprazlama ile birlikte, amaca uyarlanmış paralel çizelgeleme uygulamışlardır. Hartmann (1998), tek noktadan çaprazlama, çift noktadan çaprazlama (*two-point crossover*) ve düzgün sıralı çaprazlama (*uniform crossover*) yöntemlerini sınamıştır. Yine her gösterimde, seri çizelgeleme kullanılmıştır. Sivrikaya-Şerifoğlu (1997) sunduğu GA yaklaşımını; NBD (net bugünkü değer)'nin maksimize edilmesinin yanı sıra, proje süresinin en küçüklenmesi amacına yönelik olarak da uygulamıştır. Proje süresinin en küçüklenmesi için yinelemeli yerel kısıt tabanlı analiz ile karşılaştırılma yapılmış ve daha iyi sonuçlar verdiği gösterilmiştir. Sivrikaya-Şerifoğlu ve ark. (2000) bu algoritmayı yenilenemez kaynakları da içerecek şekilde genişletmişlerdir.

Mori ve Tseng (1997) sadece yenilenebilir kaynakların bulunduğu kaynak kısıtlı proje çizelgeleme problemini ele almışlardır. Çalışmada, kromozomun her geninde bir faaliyet numarası, bu faaliyet için atanmış bir mod, çizelgeleme sırası ve başlama bitiş zamanları bilgilerini taşıyan bir kodlama uygulanmıştır. Sundukları bu GA yaklaşımını Drexl ve Grünewald (1993)'ın sundukları stokastik bir yaklaşımla karşılaştırarak, GA'nın daha iyi sonuçlar verdiğini göstermişlerdir. Hartmann (1998), hem yenilenebilir

hem de yenilenemez kaynakların yer aldığı bir GA yaklaşımı sunmuştur. Özdamar (1999) ise, yenilenemez kaynakların da yer aldığı probleme, özel bilgilerin bulunduğu bir melez GA yaklaşımı ile çözüm aramıştır. Bu yaklaşımda, dolaylı bir kodlama kullanarak, ilki faaliyetler için atanmış moda ait, diğeri bir grup öncelik kuralından belirlenen bir kural numarası olan iki boyutlu bir kromozom yapısı tanımlanmıştır. Çalışmada, bu yaklaşımla, uygun çözüm süresi içinde en iyi sonuçlara ulaşıldığı öne sürülmüştür. Wall (1996), tarafından 1000 proje üzerinde GA uygulanmıştır. Bu projeler 10-300 faaliyet boyutlarında, 3-10 arasında sınırlı kaynağı içermektedir. Çalışma neticesinde GA, problemlerin çoğunda iyi görünmüş, % 2 sapma ve % 60 projede ise en iyi çözümü yakalamıştır. Hegazy (1999), 6 adet kaynakla sınırlandırdığı, 20 faaliyetli proje programlama modelini üç farklı proje programlama yazılımına yüklemiş ve projenin çözümünü yapmıştır. Güvenç (2000), Goldberg(1989) tarafından tanımlanan basit algoritmayı kullanılarak, çok modlu kaynak kısıtlı proje planlaması için Delphi 4.0 programlama dili ile bir bilgisayar programı hazırlamıştır. Alcaraz ve Maroto(2003) çok fazla sayıda proje için iyi çözüm veren iki çaprazlama tekniği önermiş ve çözümlerinin gösterimi için standart faaliyet listesi gösterimine dayalı yeni bir gösterim şekli önermişlerdir. Mutasyon operatörü olarak da yine Hartmann (1998) ve Boctor(1990) tarafından önerilen yaklaşımları kullanmışlardır.

Tavlama Benzetimi yöntemi ise, ilk defa Kirkpatrick ve Gelatt (1983) tarafından kaynak kısıtlı proje çizelgelemede kullanılmıştır. Lee ve Kim (1996) önerdikleri tavlama benzetimi yönteminde rastlantısal anahtar gösterimi ve paralel çizelgeleme kullanmışlardır. Yeni çözüm yaratmak için kısıtlı çiftli yer değiştirme (*pair wise interchange*) uygulamışlardır. Cho ve Kim (1997), Lee ve Kim (1996)'in tavlama benzetimi yönteminden hareketle bir algoritma önermişlerdir. Bu algoritmada rastlantısal anahtar gösterimi, bazı atanabilir faaliyetlerin ertelenmelerine izin verecek şekilde uyarlanmış ve paralel çizelgeleme yöntemi kullanılmıştır. Bouleimen ve Lecocq (2003), yenilenemez kaynakları da içeren bir tavlmalı benzetim yaklaşımı önermişlerdir. Jozefowska ve ark. (1999), yenilenemez kaynakları da içeren yaklaşımlarında faaliyet listesi ve mod listesi üzerinde çalışmışlar ve iki ayrı çözüm stratejisi önermişlerdir. Birinci yaklaşım, sürekli olurlu çözümlerle çalışmakta ve proje

süresini en küçüklemeyi amaçlamaktadır. Diğer yaklaşım ise, yenilenemez kaynaklar açısından olurlu olmayan çözümleri de kullanır.

Yasaklı Arama yöntemi, daha önce incelenmiş belli sayıda çözümü tabu listesi olarak adlandırılan listede tutarak, o çözümlere geri dönmeyi belirli bir zaman yasaklayan ve böylece aramayı çözüm uzayının daha iyi noktalarına yönlendiren bir arama prosedürüdür. Brucker ve ark. (1999), gerek çözüm gösterimi, gerekse komşu küme tanımı için farklı iki ayrı yöntem önermişlerdir. Başlangıç çözümleri için sezgisel karar kuralları kullanmışlardır. Bu algoritmanın 30 faaliyetli problemlerde en iyi değerlerden ortalama % 0.4 ve en fazla % 6.9 saptığını belirlemişlerdir.

Parçacık sürü optimizasyonu algoritmasının KKPÇP'nin çözümünde kullanılması ise diğer ileri sezgisel tekniklere göre daha yakın zamana tekabül etmektedir. İlk olarak, Zhang ve ark. (2005), proje bitiş süresinin en küçüklenmesi amaçlı KKPÇP için çözümlerinin çok boyutlu parçacıklar tarafından temsil edildiği parçacık sürü optimizasyonu metodolojisi geliştirmişlerdir. Bu çalışmada, KKPÇP'nin çözümlerinin temsili için, öncelik temelli gösterim (priority based) ve faaliyet temelli gösterimin (permutation based) kullanıldığı iki farklı PSO algoritması ortaya konmuştur. Çizelge oluşturma planı olarak ise, seri çizelge oluşturma planı kullanılmıştır. Çözülen sayısal örneklerin sonuçlarına dayalı olarak, PSO'nun da problemin çözümünde kullanılan diğer ileri sezgiseller kadar iyi sonuç verdiği vurgulanmıştır. Aynı araştırmacılar diğer bir çalışmalarında önerdikleri PSO algoritmasında, çözümlerin temsili için öncelik temelli gösterim kullanılmış ve parçacıklarla temsil edilen bu önceliklerin çizelgeye çevrilmesi için paralel çizelge oluşturma şemaları kullanılmıştır. Çözülen örneklerin sonuçlarına dayalı olarak, KKPÇP için PSO temelli yaklaşımın global en iyiyi (global optima) araştırma yeteneği olduğu ortaya konulmuştur. Ayrıca, bu yöntemin, PSO araştırmaları boyunca tek yönlü deneyim paylaşma mekanizmasına sahip olması ve daha az parametre ayarı gerektirmesi nedeniyle genetik algoritmaya göre daha kullanışlı olduğu belirtilmiştir. Sonuç olarak, KKPÇP için PSO algoritmasının etkin ve kullanımı kolay bir araç olduğu vurgulanmıştır (Zhang ve ark., 2006).

Çözümlerin temsiliinde faaliyet temelli gösteriminin kullanıldığı bir PSO uygulaması ise Zhang ve ark. (1996) tarafından yapılmıştır. Bu çalışmanın sonuç

kısımında da, PSO'nun kaynak kısıtlı proje çizelgeleme probleminin çözümü için kullanılabilecek iyi bir alternatif olduğu belirtilmiştir. Diğer çalışmalara benzer olarak, PSO'nun genetik algoritmaya göre üstün yanları olarak, araştırma süreci boyunca yerel ve global en iyi deneyimlerinin tek yönlü paylaşımının yapılması ve iyi sonuçları elde edebilmek için kolayca ayarlanan birkaç parametreye sahip olması üzerinde durulmuştur.

KKPÇP'de çözümlerin, yukarıda kullanılan gösterimlerden farklı olarak, faaliyetlerin başlama süreleri ile temsil edildiği bir PSO prosedürü Zhang ve ark.(2009) tarafından yapılan çalışmada ortaya konulmuştur. Bu algoritmada, çözümler direk olarak çizelge ile temsil edildiği için herhangi bir çizelge oluşturma planına ihtiyaç duyulmamaktadır. Bu yaklaşımda, her bir faaliyetin en erken ve en geç başlama sürelerine dayalı olarak faaliyetlerin başlayabileceği zaman aralıkları oluşturulmakta ve sonrasında bu aralıkta rasgele olarak üretilen sayılar ile başlangıç çözüm oluşturulmaktadır. Yalnız bu gösterimde, faaliyetlerin öncelik ilişkileri açısından olurlu olmayan çözümleri de içerebilmesi nedeniyle parçacıkların tamir edilebilmesi için bir tamir prosedürü ortaya konulmuştur.

Bu çalışmaların dışında, KKPÇP'nin çözümü için, farklı algoritmaların bazı özelliklerinin kombine edilmesi ile oluşturulmuş hibrit ileri sezgiseller de geliştirilmiştir. Valls ve ark. (2008), KKPÇP'nin çözümü için bir hibrit genetik algoritma ortaya koymuşlardır. HGA adını verdikleri bu algoritmanın bilenen GA'dan farklı özelliklerini şu şekilde sıralamışlardır; KKPÇP'ye özgü bir çaprazlama operatörü, oluşturulan bütün çizelgelere uygulanan bir lokal geliştirme operatörü, seçilen ebeveynlerin kombine edilmesi için geliştirilen yeni bir yöntem ve ilk aşamada bulunan en iyi çözümün komşu popülasyonundan aramaya devam eden iki aşamalı bir yöntem. Chen ve ark.(2010), KKPÇP'nin çözümü için karınca kolonisi optimizasyonu ve dağıtık arama (scatter search) algoritmasını kombine etmişlerdir. Önerilen proseste, ilk olarak karınca kolonisi optimizasyonu çözüm uzayını araştırmakta ve dağıtık aramanın başlangıç çözümünü oluşturmaktadır. Agarwal ve ark. (2011), lokal aramanın yapay sinir ağları yinelemeleri ile ve global aramanın ise GA yinelemeleri ile gerçekleştirildiği bir hibrit yaklaşım ortaya koymuşlardır. Bu yöntemlerin, mevcut problem için ayrı ayrı da iyi sonuç verdikleri belirtilmiş ancak hibrit yaklaşımın her birinin ayrı ayrı

kullanılmasından daha iyi sonuç verdiği vurgulanmıştır. Önerdikleri algoritmayı, literatürde KKPÇP’de kıyaslama problemleri olarak sıklıkla kullanılan PSPLIB (Project scheduling library)’de yer alan J30, J60, J90 ve J120 verileri ile test etmişlerdir.

2.1.2. KKPÇP’de Faaliyet Sürelerindeki Belirsizliği Dikkate Alan Çalışmalar

Genel olarak projelerdeki belirsizliklerle baş çıkabilmek için beş farklı çizelgeleme yaklaşımı kullanılmaktadır. Bunlar daha önce yapılan çalışmalarda; reaktif çizelgeleme, stokastik çizelgeleme, bulanık çizelgeleme, kararlı ve dayanıklı çizelgeleme (robust) çizelgeleme ve duyarlılık analizi olarak sınıflandırılmıştır.

Stokastik kaynak kısıtlı proje çizelgeleme, yenilenebilir kaynak kısıtları ve öncelik ilişkilerini dikkate alarak projenin beklenen bitiş süresini en küçüklemeyi amaçlar. Bu konuda yapılan çalışmalar, genellikle değişik sezgisel yöntemler ile çözüm önerileri getirmişlerdir (Pet-Edwards, 1996; Golenko-Ginzburg and Gonik, 1997; Tsai and Gemmil, 1996, 1998). Golenko-Ginzburg and Gonik (1997), faaliyet sürelerini belirli bir yoğunluk fonksiyonuna sahip (beta, düzgün, normal dağılım kullanılmıştır) bir rasgele değişken ile ifade etmişler ve faaliyet süreleri için daha önceden belirli olan üst ve alt sınırlar kullanmışlardır. Amaç fonksiyonu, beklenen proje süresini en küçüklemektir. Çözüm süreçlerini bir sayısal örnek üzerinde göstermişlerdir. Tsai and Gemmil (1998) ise aynı problem için “Yasaklı arama” ileri sezgisel çözüm yöntemini kullanmışlardır. Proje faaliyetlerinin süreleri için “beta” dağılımı kullanmışlar ve beta dağılımının parametrelerini hesaplayabilmek için iyimser, kötümser ve en yüksek olasılıklı tahminleri kullanmışlardır. Beklenen faaliyet sürelerini ve En Kısa Bolluk Süresi sezgiselini kullanarak başlangıç olurlu çözümleri bulmuşlardır. Pet- Edwards (1996), stokastik kaynak kısıtlı proje çizelgelemede en iyi çizelgeleme sezgiselleri ile GA’nın nasıl kombine edilebileceğini araştırmıştır. Genetik algoritmada kullanılmak üzere seçilen her bir sezgisel algoritma kombinasyonlarının performanslarını değerlendirmek amacıyla bir simülasyon modeli kullanmıştır. Kullanılan sezgisel yaklaşımlar; “en kısa ortalama faaliyet süresi (the shortest mean activity duration)” ve “en az kaynak sayısı (the least number of resources)”dır.

Belirsizliği model yansıtabilmek amacıyla yaygın olarak kullanılan diğer bir yaklaşım ise “bulanık kaynak kısıtlı proje çizelgelemedir”. Bu yaklaşımın savunucuları, geçmiş verilerin eksikliğinden dolayı faaliyet sürelerinin olasılık dağılımlarının bilinemeyeceğini belirtmişlerdir. Uygulamada, tekrarlı olmayan ve spesifik özelliklere sahip bir proje için faaliyet süreleri çoğu zaman insanların deneyimlerinden yararlanılarak tahmin edilmek zorunda olduğu için proje yöneticileri çoğu zaman “muğlak (vague)” ve kesin olmayan “imprecise) yargılara dayalı tanımlar yapabilmektedir. Klasik anlamda bilinen belirsizlikten (uncertainty) ziyade muğlak (imprecise) bilgilerin yer aldığı durumlarda faaliyet sürelerini modellemek için “stokastik değişkenlerden” ziyade “bulanık sayılar” kullanılmıştır.

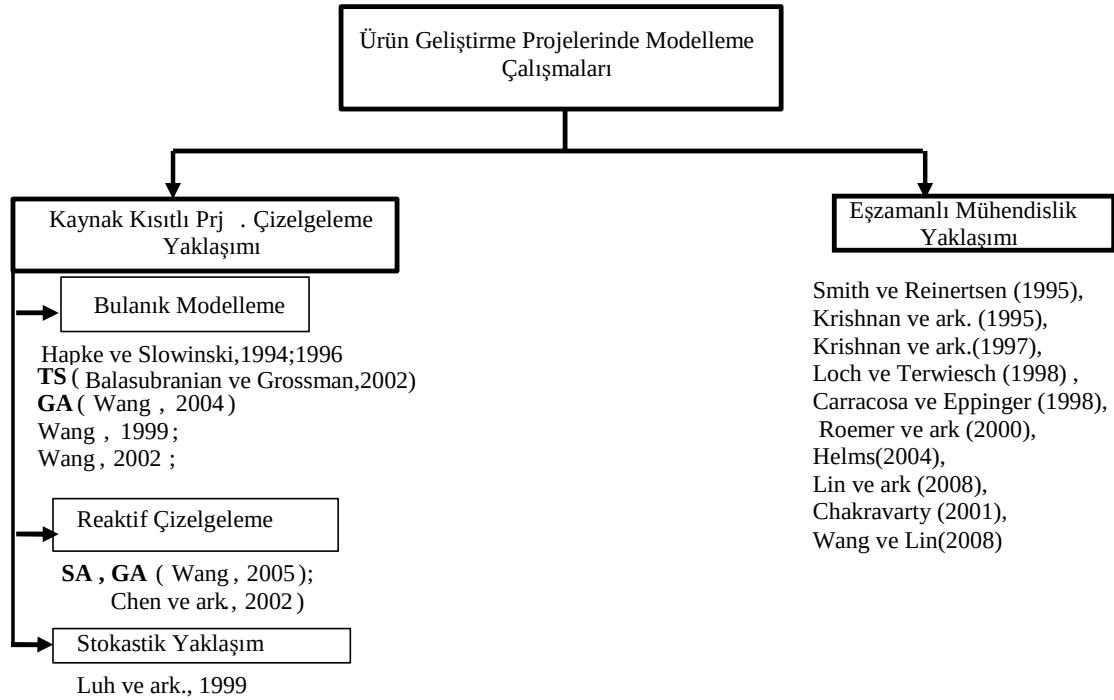
Kaynak kısıtlı proje çizelgeleme için ilk bulanık modelleme yaklaşımı Hapke ve ark.(1994) tarafından gerçekleştirilmiştir. Çalışmalarında, seri ve paralel çizelgelemeye dayalı öncelik kurallarını bulanık sayıların kullanımı ile genişletmişlerdir. Hapke and Slowinski (2000) ise çok amaçlı bulanık kaynak kısıtlı proje çizelgeleme probleminin çözümü için “Benzetim Tavlamasının” kullanımını göstermişlerdir. Önerdikleri algoritma Czyzak ve Jaskiewicz (1996) tarafından önerilen ve “Pareto Benzetim Tavlaması” algoritmasının belirsizlik içeren proje süreleri için değiştirilmiş halidir. Özdamar and Alanya (2000), proje sürelerinde belirsizlik içeren yazılım geliştirme projelerinde çizelgeleme problemi üzerinde çalışmışlar ve problem için doğrusal olmayan karmaşık tamsayılı matematiksel model geliştirmişlerdir. Önerdikleri modeli çözebilmek için bir sezgisel geliştirmişlerdir. Faaliyetlerin bulanık faaliyet süreleri ile temsil edilebilen birden fazla yolla (multi-mode) gerçekleştirilebileceğini kabul etmişlerdir. Amaçları, proje süresini en küçükmektir .

Belirsizlik içeren kaynak kısıtlı proje çizelgeleme yaklaşımı ürün geliştirme projelerinin çizelgelenmesinde de kullanılmıştır. Bu alanda yapılan çalışmalara Bölüm2.2’de yer verilmektedir.

2.2. Yeni Ürün Geliştirme Projeleri İle İlgili Modelleme Çalışmaları

Bu bölümde yeni ürün geliştirme projelerinin çizelgelenmesi başlığı altında yapılan çalışmaların yanı sıra yeni ürün geliştirme sürecine özgü özelliklerinin

anlaşılmasına yönelik bazı süreç modelleme ve analiz çalışmalarına da yer verilmiştir. İncelenen çalışmalar Şekil 2.2’de görüldüğü gibi sınıflandırılmıştır.



Şekil 2.2. Ürün geliştirme projelerinde modelleme çalışmaları

2.2.1. Yeni Ürün Geliştirme Projelerinde Çizelgeleme Çalışmaları

Ürün geliştirme projelerinin çizelgelenmesi ile ilgili çalışmaların ortak noktası problemin “kaynak kısıtlı proje çizelgeleme problemi” olduğudur. Literatürde yapılan çalışmalar, diğer proje çizelgeleme çalışmalarından ayrı olarak YÜG sürecinin yapısal özellikleri nedeniyle bazı belirsizlikleri barındırdığını kabul etmişlerdir. Bu nedenle, stokastik, reaktif ve bulanık modelleme yaklaşımları kullanılmıştır. Literatürde “kaynak kısıtlı proje çizelgeleme problemi” ile ilgili çok fazla çalışma vardır. Ancak, YÜG sürecinin bazı spesifik ve yapısal özelliklerini barındıran modeller azdır ve son yıllarda daha fazla gündeme gelmiştir.

Mevcut yaklaşımlar arasından YÜG projelerinin çizelgelenmesinde en sıklıkla karşımıza çıkan yöntem “bulanık çizelgelemedir”. Bu alanda yapılan çalışmalar, yeni ürün geliştirme projelerinin yapısal olarak birbirinde farklı olduğunu öne sürerek, belirsizliğin modellenmesinde istatistiksel verilere dayanan stokastik modellemenin kullanılmasındansa bulanık modellemenin daha uygun olduğunu öne sürmüşlerdir.

Wang (1999), kesin olmayan bilgiye sahip ürün geliştirme projelerinde çizelgeleme için bir yöntem geliştirmiştir. Problem, “bulanık kısıt tatmini” problemi olarak modellenmiş ve bulanık kısıtların tatmin derecesinin belirlenebilmesi için yeni bir metot önerilmiştir. Önerilen metot kullanılarak bütün bulanık kısıtların tatmin olasılığını en büyükleyen ve proje gecikmesini en küçükleyen bulanık çizelgeleme algoritması geliştirilmiştir. Sayısal örnekler ile yaklaşımın çözüm süresi ve çözüm kalitesi incelenmiştir.

Wang, (2002), ürün geliştirme projelerindeki belirsizlik ve esnek bilgi ile baş edebilmek için “bulanık çizelgeleme” yaklaşımını kullanmış ve faaliyetlerin sürelerini “üçgen bulanık sayı” ile ifade etmiştir. Projenin hazırlık ve bitiş süresi ise esnek bilgi olarak ele alınıp yamuk bulanık sayılar ile ifade edilmiştir. Bitiş süresindeki esneklik ile proje yöneticisinin projenin bitirilmesi için en uygun süre, en geç süre gibi öngörülerinin modele aktarılmasını sağlamak amaçlanmıştır. Çizelgenin performansını değerlendirebilmek için ise “çizelge riski” ölçütü kullanılmıştır. En düşük risk ile çizelgenin oluşturulabilmesi için “bulanık demet arama algoritması (fuzzy beam search)” geliştirilmiştir. Aynı problem için, bir sonraki çalışmada ise projede yer alan mevcut belirsizliklerle baş edebilmek amacıyla “bulanık küme” teorisine dayalı “dayanıklı ve kalıcı çizelgeleme (robust scheduling) metodolojisinin” geliştirilmesini amaçlamıştır. Projede yer alan belirsizlik içeren parametreler bulanık küme teorisi kullanılarak temsil edilmiştir. Aynı zamanda, sayısal olasılık teorisine dayalı “çizelgenin dayanıklılığı” ölçütü tanımlanmıştır. Problemin, çizelgenin dayanıklılığı açısından kabul edilebilir sınırlar içerisinde çözülebilmesi için “Genetik Algoritma” kullanılmıştır. Modelin amacı, en fazla çizelge dayanıklılığını sağlayacak çizelgeyi oluşturmaktır (Wang, 2004).

Balasubramanian, Grossman (2003)’da çalışmalarda faaliyet sürelerinin belirsiz olduğu çizelgeleme problemleri için bulanık küme teorisine dayalı yaklaşımlar

geliştirilmiştir. İlk olarak, bulanık küme teorisi hakkında bilgi verilmiş ve birisi yeni ürün geliştirme projesi olan iki problem dikkate alınarak karmaşık tamsayılı model geliştirilmiştir. Önerilen modelin, en yüksek olasılıklı, iyimser ve kötümser sürenin hesaplanmasında nasıl kullanılacağı örnekler üzerinde gösterilmiştir. Sonuç kısmında, olasılık teorisine dayalı çözümlerle kıyaslandığında bulanık küme teorisinin avantajlar şu şekilde sıralanmıştır; geçmişe ait verilerin mevcut olmadığı durumlarda belirsizlik altında modelleme imkanı sağlaması, belirsizliğin modellenmesinde oldukça fazla esneklik sağlaması, sezgisel arama algoritmaları ile makul süreler içerisinde çözüm verebilmesidir. Bu durum, amaç fonksiyonu değerlerinin hesaplanmasındaki kolaylıktan kaynaklanmaktadır.

Belirsizliğin modele dahil edilmesinde kullanılan diğer bir yaklaşım ise “reaktif çizelgeleme” olmuştur. Wang (2005), ürün geliştirme projelerinin çizelgelenmesini “Dinamik kısıt tatmini problemi (DCSP: Dynamic Constraint Satisfaction Problem)” olarak ele almıştır. Çalışmada, beklenmeyen durumlar nedeniyle proje planlama safhasında oluşturulan orijinal çizelgenin genellikle bozulduğu öne sürülmüş ve gerçek durumu yansıtacak şekilde çizelgenin ayarlanması gerektiği vurgulanmıştır. Bu nedenle, belirsizliğin olduğu ürün geliştirme ortamında beklenmeyen olaylar neticesinde bozulan çizelgenin ileri sezgisel teknikler yardımıyla tamir edilmesi için “reaktif çizelge” geliştirilmiştir. Problemi, çizelge hatalarını yakalayabilecek ve tamir edebilecek yeteneğe sahip esnek bir yapısı olan “dinamik kısıt tatmini problemi (DCSP)” olarak ele almıştır. Tavlama Benzetimi ve Genetik Algoritmanın önerilen reaktif çizelgenin çözümü için uygun olduğu bulunmuştur. Önerilen çözüm yöntemleri birçok problem üzerinde test edilmiş ve tatmin edici sonuçlar bulunmuştur.

Zhuang ve Yasinse(2004) ise yeni ürün geliştirme projelerinin çoğunlukla çoklu proje ortamında gerçekleştirildiğini öne sürmüşlerdir. Çalışmalarında, ürün geliştirme ortamında genellikle aynı kaynakları kullanan birden fazla projenin olduğu ve bunların eş zamanlı olarak gerçekleştirildiği durum dikkate alınmıştır. Ancak projeler arasında herhangi bir öncelik ilişkisi olmadığını kabul etmişlerdir. Projelerin faaliyetleri arasındaki ilişkilerinin gösterilmesin Tasarım Yapısı Matrisi (DSM -Design Structure Matrix) kullanmışlardır. Kaynak kısıtlı bu proje çizelgeleme probleminde projenin

bitirilme süresini en küçükleme için tek noktalı çaprazlama operatörü kullanan bir genetik algoritma geliştirmişlerdir.

2.2.2. Yeni Ürün Geliştirme Projelerinde Faaliyetlerin Üst Üste Çakıştırılması ile İlgili Çalışmalar

Her ne kadar yukarıda bahsedilen çalışmalar yeni ürün geliştirme projelerinde çizelgeleme problemini ele almış olsa da YÜG projelerinde faaliyet sürelerini kısaltmak için sıklıkla kullanılan üst üste çakıştırma uygulamasını içermemektedir (Lawson ve Karandikar,1994; Roemer ve Ahmadi, 2004).

Üst üste çakıştırma, üst seviye faaliyet tamamlanmadan temel bilgi ile alt seviye faaliyetin başlanması durumudur. Bazı deneysel çalışmalar, üst üste çakıştırmanın yeniden çalışmaya neden olmasına rağmen iyi planlandığı takdirde proje bitiş süresini kısaltabileceğini göstermiştir (Smith ve Reinertsen, 1995; Helms, 2004). Ancak faaliyetlerin üst üste çakıştırılması başlangıç bilgisine dayalı olarak gerçekleştirildiği için bazı dezavantajları da ortaya çıkabilmektedir. Örneğin, alt seviye faaliyetin çok erken başlatılması durumunda üst seviye faaliyetin bilgisinde ortaya çıkan büyük değişikliklerin alt seviye faaliyet tarafından güncellenmesi için çok fazla yeniden çalışma yapılması gerekebilir. Bu durum proje bitiş süresini kısaltmaktan ziyade tam tersi etki yaparak uzamasına neden olabilir. Bu nedenle birtakım araştırmacılar, üst üste çakıştırma içeren ürün geliştirme projelerini analiz etmek için bazı modeller geliştirilmişlerdir (Krishnan ve ark.,1997; Loch ve Terwiesch,1998; Roemer ve ark, 2000; Carracosa ve Eppinger, 1998; Lin ve ark, 2008). Krishnan (1996), en uygun üst üste çakıştırma oranının belirlenmesinde iki önemli parametre belirtmiştir. Bunlar; üst seviye faaliyet bilgi gelişimi ve alt seviye faaliyet duyarlılığıdır. Gelişim, bir üst seviye tasarım faaliyetinin son değerine yaklaşma hızı olarak tanımlanmıştır. Duyarlılık ise, verilen üst seviye parametre değişikliklerini tolere edebilmek için alt seviye faaliyette gerekli olan yeniden çalışma miktarını temsil etmektedir. Bu durumda, daha fazla üst üste geçirmenin ancak, üst seviye faaliyetin hızla geliştiği ve alt seviye faaliyetin duyarlılığının daha az olduğu durumda daha faydalı olduğunu belirtmiştir. Krishan ve ark. (1997) ise, üst üste çakıştırılmış faaliyetler arasındaki en uygun bilgi transferi sayısı ve zamanı için temel bir çerçeve geliştirmişlerdir. Üst üste çakıştırma stratejilerini üst

seviye bilgi değişimi ve alt seviye duyarlılığına dayalı olarak açıklamışlardır. Lock ve Terwiesch (1998), Krishnan(1996) tarafından geliştirilen bilgi iletimi (information exchange) modelini genişletmişlerdir. Krishnan (1996)'ın modeline benzer olarak, üst seviye faaliyetteki değişimin alt seviye faaliyeti etkilediği iki faaliyetli eşzamanlı mühendislik yaklaşımı tanımlamışlardır. Daha sonra modeli bilgi iletişim sıklığını ve en uygun üst üste çakıştırma miktarını hesaplamak için kullanmışlardır. Roemer ve ark. (2000), üst üste çakıştırmadan elde edilen zaman kazancı ve bu uygulamanın maliyetini dikkate alarak bir zaman-maliyet analizi gerçekleştirmiş ve analitik modelini kurmuşlardır. Bu modelde dikkate alınan maliyetten kasıt, üst üste çakıştırmadan oluşabilecek yeniden çalışma miktarıdır. Aynı şekilde, Chakravarty (2001), üst üste çakıştırmamanın riski ve kazanılan zaman arasındaki denge üzerine çalışmıştır. Bahsedilen çalışmalardan farklı olarak Lin ve ark.(2008), alt seviye faaliyetin, üst seviye faaliyetin hatalarını yakalayıp üst seviye faaliyete geri bildirim verdiği durumu göz önüne alarak üst üste çakıştırılmış yinelemeli ürün geliştirme için bir kesikli simülasyon modeli geliştirmişler ve farklı üst üste çakıştırma durumlarını analiz etmişlerdir. Geliştirdikleri simülasyon modeli ile farklı üst üste çakıştırma politikalarının proje performansı üzerindeki etkisini tahmin edebilmek için bir araç sunmuşlardır. Benzer olarak, Wang ve Lin (2009), proses yapısının yani faaliyetlerin sırasının ve farklı üst üste çakıştırma derecelerinin proje bitiş süresi üzerindeki etkisi inceleyebilmek amacıyla bir kesikli zaman simülasyon modeli geliştirmişlerdir. Yeniden çalışma miktarının belirlenmesinde Krishnan(1996)'nın ortaya koyduğu üst seviye faaliyet bilgi gelişimi ve alt seviye faaliyet duyarlılığını dikkate almışlardır.

2.3. Literatürün Değerlendirilmesi

Yapılan literatür taramaları neticesinde, “yeni ürün geliştirme projelerinin çizelgelenmesi” alanında yapılan çizelgeleme çalışmalarının hepsinde mevcut problemin, yenilenebilir çoklu kaynak kısıtlarını ve proje faaliyetleri arasındaki öncelik ilişkilerini dikkate alan, proje bitiş süresinin en küçüklenmesinin amaçlandığı bir KKPC problemi olarak tanımlandığı görülmüştür. Bu alanda yapılan çalışmalar, yeni ürün geliştirme projelerinde belirsizlik durumunu sürece özgü bir özellik olarak vurgulamış ve yeni

ürün geliştirme projelerinde kullanılabilecek uygulamaları dikkate almamışlardır. Oysa, yeni ürün geliştirme projeleri ile ilgili yapılan çalışmaların incelenmesi neticesinde, rekabet avantajının sağlanabilmesi açısından proje bitiş süresinin kısaltılmasının oldukça önemli olduğunun sıklıkla vurgulandığı görülmüştür. Luh ve ark. (1999) bildirdiklerine göre, ürün geliştirme süresinin önemi Nichols (1990) tarafından yapılan çalışma ile oldukça net bir şekilde gösterilmiştir. Bu çalışmaya göre ürün pazara 6 ay daha geç çıkartılırsa karın yaklaşık üçte biri kadar kayıp olmaktadır. Bu nedenle, YÜG projelerinde sürenin kısaltılması amacıyla birtakım araçların kullanıldığı yapılan literatür araştırmaları sonucunda görülmüştür. Bunlar arasında literatürde en çok bahsedilen uygulamalardan birisi de “faaliyetlerin üst üste” çakıştırılması uygulamasıdır. Yapılan deneysel çalışmalar bu uygulamanın iyi planlandığı takdirde proje süresini kısaltabileceğini göstermiştir. Literatürde bu alanda yapılan çalışmalar genellikle; uygulama ile ilgili kavramsal modellerin ortaya konması, alt seviye ve üst seviye faaliyet arasındaki bilgi iletimi sıklığının belirlenmesi, üst üste çakıştırmadan doğan risk elde edilen kazancın dengelenmesine yönelik modelleme çalışmalarından oluşmaktadır.

Bu tez çalışmasında ele alınan YÜG projelerinin çizelgelenmesi problemi, problemin yalnızca bir KKPÇP olarak tanımlanmasından ziyade YÜG projelerinde sıklıkla kullanıldığı bilenen faaliyetlerin üst üste çakıştırılması uygulamasını da modele dahil etmektedir. Yapılan çalışmada, kaynak kısıtlılığı ve faaliyetler arasındaki öncelik ilişkileri dikkate alınarak, proje süresini en küçükleyecek şekilde projenin çizelgelenmesi gerçekleştirilirken aynı zamanda önceden belirlenen faaliyetler arasındaki uygun üst üste çakıştırma miktarı da amaca uygun olarak belirlenmektedir. Üst üste çakıştırma miktarının belirlenmesi kararı, üst üste çakıştırmadan doğan yeniden çalışma miktarı ve bu uygulamadan elde edilen süre kazancının dengelenmesi esasına göre gerçekleştirilmektedir. Bu sayede bu uygulamanın doğru bir şekilde planlanması sağlanabilmektedir. Ayrıca, önerilen sezgisel algoritma ile farklı çoklu üst üste çakıştırma alternatifleri kısa süreler içinde değerlendirilebilmekte ve en iyi kazancı sağlayan farklı üst üste çakıştırma alternatifleri orta konabilmektedir.

3. MATERYAL VE METOD

3.1. Materyal

Daha öncede belirtildiği gibi, ele alınan problem “yeni ürün geliştirme projelerinin çizelgelenmesi ve kaynak tahsisi” problemidir ve yeni ürün geliştirme projelerinde bitiş süresinin kısaltılması için sıklıkla kullanılan üst üste çakıştırma uygulamasını da içermektedir. Bu şekilde, üst üste çakıştırma uygulamasının iyi bir şekilde planlanmasını sağlayarak projenin bitiş süresinin kısaltılması yönündeki alternatiflerin değerlendirilebilmesini amaçlamaktadır. Bu doğrultuda geliştirilen gerek matematiksel modelin gerekse sezgisel algoritmanın performansının farklı proje koşulları altında test edilmesi için örnek verilere ihtiyaç vardır. Örnek verilerin araştırılmasına geçilmeden önce, ele alınan problem için çözüm kalitesini ve süresini etkileyebilecek faktörler, literatürde kaynak kısıtlı proje çizelgeleme alanındaki çalışmaların incelenmesi neticesinde belirlenmiştir. Bu faktörler, KKPC alanında daha önce yapılan çalışmalarda dikkate alınarak, proje büyüklüğü (projedeki faaliyet sayısı), proje ağının karmaşıklığı, kaynak gücü(kaynak kısıtlılığı) olarak belirlenmiştir. Bu faktörlere göre önerilen modelin ve test edilebilmesi için yeterli sayıda yeni ürün geliştirme projesi bulunmasının mümkün olmaması nedeniyle, KKPC alanında kıyaslama problemleri olarak sıklıkla kullanılan j30-j60-j90-j120 olarak bilinen örnek proje ağları içinden seçilen problemler kullanılmıştır.

Test amaçlı kullanılan problemler ve bu problemlere ait parametre değerleri ile şimdiye kadar bulunan en iyi çözümler, “<http://129.187.231/psplib/>” adresinden elde edilmiştir. J30, J60, J90, J120, örnek proje ağları, sırasıyla 30, 60, 90, ve 120 faaliyet ile 4 yenilenebilir kaynak içermektedir. Bu proje ağlarına ait örnekler, farklı parametreler dikkate alınarak sınıflandırılmıştır. Yapılan çalışmada, kaynak gücü ve proje karmaşıklığı faktörleri 3 farklı seviyede dikkate alınarak her bir proje ağı grubu için dokuz farklı örnek proje ağı test amaçlı olarak kullanılmıştır.

Algoritmaların test edilmesi için seçilen problemlerin özellikleri; karmaşıklığı, büyüklüğü, kaynakların kıtlığı/bolluğu gibi faktörler çoğu zaman KKPC problemlerinin çözüm süresi ve kalitesini oldukça fazla etkileyen faktörlerdir.

Ağ karmaşıklığı faktörü, faaliyet başına düşen ortalama öncelik ilişkileri anlamına gelmektedir. Bir ağdaki öncelik sonralık ilişkilerinin miktarını belirlemektedir. Kaynak gücü ise faaliyetler için gereken kaynak miktarının, kullanılabilir kaynak ile ilişkisini göstermektedir. Kaynakların bol ya da kıt olduğunu belirlemekte kullanılan bu faktör; bir kaynağın kullanılabilir üst sınırının, bu kaynağın ortalama kullanım miktarına bölünmesi ile elde edilmektedir. Bu nedenle, düşük kaynak gücü, çizelgelenen problemin zor bir problem olduğunu göstermektedir.

Bu açıklamalardan da anlaşılacağı gibi bazen az faaliyete sahip çizelgeleme problemleri bahsedilen faktörlerin etkisi ile çok faaliyetli bazı problemlerden daha zor bir problem olabilir. Dolayısıyla, geliştirilen algoritmanın PSPLIB'deki problemler üzerinden test edilmesi algoritmanın güvenilirliğinin daha net bir şekilde ortaya konmasını sağlamaktadır. Test amacıyla kullanılan proje verilerinin özellikleri Tablo 3.1'de verilmiştir.

Tablo 3.1. Test amaçlı kullanılan proje verilerinin özellikleri

Problem No	Faaliyet Sayısı	Kaynak Gücü	Ağ Karmaşıklığı
1	30	Düşük	Düşük
2	30	Orta	Düşük
3	30	Yüksek	Düşük
4	30	Düşük	Orta
5	30	Orta	Orta
6	30	Yüksek	Orta
7	30	Düşük	Yüksek
8	30	Orta	Yüksek
9	30	Yüksek	Yüksek
10	60	Düşük	Düşük
11	60	Orta	Düşük
12	60	Yüksek	Düşük
13	60	Düşük	Orta
14	60	Orta	Orta
15	60	Yüksek	Orta
16	60	Düşük	Yüksek
17	60	Orta	Yüksek
18	60	Yüksek	Yüksek
19	90	Düşük	Düşük
20	90	Orta	Düşük
21	90	Yüksek	Düşük
22	90	Düşük	Orta
23	90	Orta	Orta
24	90	Yüksek	Orta
25	90	Düşük	Yüksek
26	90	Orta	Yüksek
27	90	Yüksek	Yüksek
28	120	Düşük	Düşük
29	120	Orta	Düşük
30	120	Yüksek	Düşük
31	120	Düşük	Orta
32	120	Orta	Orta
33	120	Yüksek	Orta
34	120	Düşük	Yüksek
35	120	Orta	Yüksek
36	120	Yüksek	Yüksek

Son olarak ise, Wang (2002) çalışmasında yer alan bir yeni ürün geliştirme projesi verileri kullanılarak, senaryo analizleri yapılmıştır. Bu proje verisi, 52 faaliyet ve 4 yenilenebilir kaynaktan oluşmaktadır. Senaryo durumları, üst üste çakıştırmadan elde edilebilecek kazancı etkileyebilen parametrelerin farklı değerleri dikkate alınarak geliştirilmiştir.

3.2. Metod

Bu başlık altında, öncelikle problemin formülasyonu açıklanmaktadır. İkinci kısımda, önerilen matematiksel modelde kullanılan gösterimle birlikte modelin kısıtları ve amaç fonksiyonu verilmektedir. Üçüncü kısımda, problemin çözümünde kullanılan ileri sezgisel metodun seçimi ve bu metotla ilgili genel bilgiler verilmektedir. Dördüncü kısımda ise, seçilen ileri sezgisel metot temel alınarak geliştirilen algoritma açıklanmaktadır. Son bölümde ise, algoritmanın uygun bir programlama dili ile programlanması hakkında bilgi verilmekte ve geliştirilen programın ara yüzü tanıtılmaktadır.

3.2.1. Problemin Formülasyonu

Yeni ürün geliştirme projelerinde çizelgeleme ve kaynak tahsisi probleminin net olarak ortaya konulabilmesi için, probleme özgü özelliklerin detaylı olarak incelenmesi gerekir. Bu nedenle, bu bölümde öncelikle sistemin özellikleri hakkında bilgi verilmekte daha sonraki bölümlerde ise sırasıyla, problemin varsayımları, girdi değişkenleri ve parametreler, ve performans değişkenleri açıklanmaktadır.

3.2.1.1. Sistemin Özellikleri

Konu ile ilgili daha önce yapılan çalışmaların incelenmesi sonucunda, genel olarak ürün geliştirme projelerinin çizelgelenmesi probleminin “kaynak kısıtlı proje çizelgeleme problemi” şeklinde modellenmenin gerçek hayat problemlerine daha uygun olacağı görülmüştür. Diğer bazı projelerde olduğu gibi yeni ürün geliştirme

projelerinde de kaynak teminin zor veya çok yüksek maliyetli olması bu modelleme yaklaşımının geçerliliğini sağlamaktadır. Bu nedenle sistemin temel özelliği olan kaynak kısıtları önerdiğimiz modelleme yaklaşımda da dikkate alınmıştır. Dikkate alınan kaynak kısıtları, literatürde yenilenebilir kaynak kısıtları olarak isimlendirilen ve kullanım ile tükenmeyen her zaman diliminden sonra tekrar kullanılabilir olan kaynaklardır. Bu kaynaklara örnek olarak, proje çalışan insan gücü, kullanılan bilgisayar, yazılım, laboratuvar, test cihazları örnek olarak verilebilir.

Bunun yanı sıra, incelenen çalışmalarda, rekabetin oldukça yüksek oranda artması ile yeni ürün geliştirme projelerinin hızlı bir şekilde bitirilmesi ve bir an önce piyasaya sunulması çok önemli bir rekabet avantajı haline geldiği özellikle vurgulanmaktadır. Bu nedenle, ürün geliştirme süreçlerinde bir takım yeni araçlar kullanılmaya başlanmıştır. Bunların içlerinde en yaygın olarak kullanılanı, eşzamanlı mühendislik yaklaşımına dayalı “faaliyetlerin üst üste çakıştırılması” uygulamasıdır. Üst üste çakıştırma, iyi planlandığı takdirde yeni ürün geliştirme projelerinin sürelerini önemli ölçüde azalttığı bazı deneysel çalışmalar ile gösterilmiştir (Smith ve Reinersten,1995; Helms,2004). Bu amaçla, bir kısım araştırmacılar faaliyetlerin üst üste çakıştırılması uygulamasının fayda-maliyet analizleri (Chakravarty, 2001), ve en uygun üst üste çakıştırma miktarı ve bilgi transferi zamanlarını belirleyen modelleme yaklaşımları üzerinde çalışmışlardır (Krishan ve ark.,1997; Roemer ve ark, 2000; Lin ve ark.,2008).

Üst üste çakıştırma yönteminde, bir faaliyet üst seviye faaliyetten elde edilen ön ve eksik bilgi ile başlatılabilir. Üst seviye faaliyet gerçekleştirildikçe, başlangıçtaki bilginin yenilendiği ve alt seviye faaliyete iletildiği göz önüne alındığında alt seviye faaliyette zaman içerisinde bir takım yeniden çalışmalar gerçekleştirilmek durumunda kalınabilir (Eastman, 1980). Bu nedenle, önerilen modelde üst üste çakıştırma miktarının belirlenmesinde, bu uygulamanın neticesinde elde edilen kazanç yani proje süresindeki kısalma miktarı ile yöntemin dezavantajı yani oluşacak yeniden çalışma miktarının dengelenmesi dikkate alınmıştır. Yeniden çalışma miktarının belirlenmesinde, Krishan(1997) çalışmasında belirtilen alt seviye faaliyetin duyarlılığı (downstream sensivity) ve üst seviye faaliyetin gelişimi (upstream evolution) kavramları dikkate alınmıştır. Üst seviye faaliyetin gelişimi, bir

üst seviye tasarım faaliyetinin son değerine yaklaşma hızı olarak tanımlanmıştır. Duyarlılık ise, verilen üst seviye parametre değişikliklerini modifiye edebilmek için alt seviye faaliyette gerekli olan yeniden çalışma miktarını temsil etmektedir. Matematiksel olarak üst seviye faaliyetin gelişim fonksiyonu ($e(t)$) aşağıdaki gibi tanımlanmıştır (Wang ve Lin,2009);

$$e(t) = \left[\frac{t - t_0}{d} \right]^a \quad (3.1)$$

Burada, t , mevcut zaman dilimini; t_0 , faaliyetin başlama zamanını; d , faaliyetin süresini; a , üst seviye faaliyetin gelişim parametresidir.

Bir faaliyetin gelişim fonksiyonu, son haline yakın tasarım bilgisi yüzdesini temsil etmektedir. Bu nedenle gelişim fonksiyonu bir faaliyette belirli bir zamanda tasarım bilgisi değişimi olasılığını tahmin etmek için kullanılabilir. Eğer p_i tanımlanmış gelişim fonksiyonuna göre t zamanında faaliyet i'nin son haline ulaşmış bilgi yüzdesi ise, $(1 - p_i)$ değişebilecek tasarım bilgisi yüzdesini verir (Wang ve Lin, 2009). Aynı çalışmada, yeniden çalışma miktarına etki eden diğer bir faktörün ise alt seviye faaliyetin duyarlılığı olduğunu belirtmiştir. Bu durumda, yeniden çalışma miktarını aşağıdaki gibi yazmıştır;

$$rw_{ik} = (1 - p_i) * ovr_{imp}_k \quad (3.2)$$

Yaptığımız çalışmada üst üste çakıştırmadan kaynaklanan yeniden çalışma miktarının bulunmasında yukarıda açıklanan çalışma temel alınmıştır. $(1 - p_i)$ olasılığı eşitlik (3.1)'in yardımıyla hesaplanmıştır ve bu olasılık üst seviye faaliyetin bilgi gelişimi etki faktörü olarak yeniden çalışma miktarını etkilemektedir. Burada t zamanı, üst üste çakıştırmanın yapıldığı zamanı diğer bir ifade ile alt seviye faaliyetin başlama zamanını ifade etmektedir. t_0 zamanı ise üst seviye faaliyetin başlama zamanını ifade eder. Matematiksel modelde, $a = 1$ kabul edilmiştir. Bunun anlamı, üst seviye faaliyetin zaman içerisinde doğrusal olarak bilgi gelişimi hızına sahip olmasıdır. Bu durumda yeniden çalışma miktarı aşağıdaki gibi hesaplanmaktadır;

$$rw_{ik} = (1 - \left[\frac{St_k - St_i}{d_i} \right]) * ovrimp_k \quad (3.3)$$

Denklemden yer alan ifadeler aşağıda açıklandığı gibidir;

rw_{ik} : alt seviye “k faaliyetinin”, üst seviye “i faaliyeti” ile üst üste çakıştırılmasından kaynaklanan yeniden çalışma miktarı

d_i : üst seviye faaliyetin süresi

St_k : alt seviye faaliyetin başlama zamanı

St_i : üst seviye faaliyetin başlama zamanı

$ovrimp_k$: alt seviye faaliyetin çakıştırma etkisi faktörü

Yapılan çalışmada alt seviye faaliyetin çakıştırma etkisi faktörü, Cho ve Eppinger(2001) çalışmasına paralel olarak alt seviye faaliyetin süresinin belirli bir yüzdesi olarak alınmıştır. Analizlerde, bu yüzde iki farklı seviyede alınmış ve bu durumda etki faktörünün düşük ve yüksek olduğu durumlar için analizler yapılmıştır.

Geliştirilen modelde, proje bitiş süresini en küçükleyecek şekilde çizelge oluşturulurken, aynı zamanda, denklem (3.3) kullanılarak, önceden belirlenen faaliyetler arasında amaca uygun şekilde en uygun üst üste çakıştırma zamanı da belirlenmektedir.

Yapılan çalışmada, YÜG projelerinde ürünün mümkün olan en kısa süre içerisinde piyasa sunulmasının önemi dikkate alınarak, faaliyetlerin üst üste çakıştırılması proje çizelgeleme problemi ile birlikte modellenmiştir. Geliştirilen model ile proje süresini en küçükleyecek şekilde faaliyetlerin çizelgelenmesi ve kaynakların faaliyetlere aktarılması kararının verilmesini sağlamanın yanı sıra proje bitiş süresinin kısaltılmasını sağlayacak şekilde, belirlenen faaliyetler arasında en uygun üst üste çakıştırma miktarının hesaplanmaktadır.

Burada çok önemli olan hususlardan biri de üst üste çakıştırma yapılacak faaliyetlerin seçimidir. Özellikle proje ağı büyüdükçe bu faaliyetlerin seçimi güçleşebilmektedir. Eğer, rasgele seçim yapılırsa projenin bitiş süresinin kısaltılması yönünde olumlu bir katkı sağlanamayabilir. Bu nedenle yapılan çalışmada, hem matematiksel model hem de önerilen sezgisel algoritma ilk olarak üst üste çakıştırma

durumunu içermeyecek şekilde oluşturulmuştur. Bu model veya algorithmadan elde edilen sonuç dikkate alınarak, kaynak kısıtlı çizelgenin kritik yolu tespit edilmekte, daha sonraki aşamada ise bu kritik yol üzerindeki uygun faaliyetler üst üste çakıştırma için aday olarak belirlenmektedir. Üst üste çakıştırma için kritik yol üzerinde bulunan faaliyetlerin tercih edilmesi Cho ve Eppinger (2001)'in çalışmalarında da önerilen bir yaklaşımdır. Daha sonraki aşamada, üst üste çakıştırma uygulamasını içeren analitik model veya sezgisel algoritma kullanılarak uygun üst üste çakıştırma miktarını da içeren yeni çizelge elde edilmektedir.

3.2.1.2. Varsayımlar

Yapılan çalışmada aşağıdaki varsayımlar esas alınmıştır;

1. Amaç proje süresinin en küçüklenmesidir.
2. Faaliyetlerin ihtiyaç duyduğu kaynak miktarı belirli ve sabittir.
3. Faaliyetlerin süreleri belirli ve sabittir.
4. Faaliyetlerin ihtiyaç duydukları kaynaklar yenilenebilir kaynaklardır (insan, laboratuvar, bilgisayar, yazılım, vs.).
5. Çoklu yenilenebilir kaynaklar ve kaynak limitleri söz konusudur.
6. Üst Üste çakıştırma yapılırken alt seviye faaliyet en erken, üst seviye faaliyetten bir zaman birimi sonra başlatılabilir.
7. Üst üste çakıştırma yapılırken alt seviye faaliyet üst seviye faaliyet bitmeden bitemez.

Önceki bölümlerde bahsedildiği gibi, dünya ticaretinin artması, bölgesel ya da ulusal pazar özelliklerinin dışına taşması ile rekabet yoğunlaşmış ve uluslararası rakipler ortaya çıkmıştır. Bu durum mevcut veya yeni pazarlarda her zaman rakipler olmasına neden olmaktadır. Bu nedenle rekabette avantajlı olmak zorunludur. Bunu sağlamanın yolu fark yaratan ürünler geliştirmek ve bunu olabildiğince hızlı yapmaktır. Ürün tasarım süresinin önemi Nichols (1990) tarafından yapılan çalışma ile oldukça net bir şekilde gösterilmiştir. Bu çalışmaya göre ürün geliştirmede yaklaşık %50 daha fazla harcama yapılması durumunda kardaki kayıp %4 iken eğer ürün pazara 6 ay daha geç çıkartılırsa bu kayıp karın yaklaşık üçte biri kadar

olmaktadır. Bu nedenle, oluşturulacak modelin ürün geliştirme projelerinin bitiş süresini en küçüklenmesi amacına dayalı olarak yapılması oldukça geçerli bir varsayımdır.

Bahsedilen ilk üç varsayım genel anlamada kaynak kısıtlı proje çizelgeleme probleminin (RCPSP-Resource Constrained Project Scheduling Problem) temelini oluşturan varsayımlardır. Buna ek olarak, kaynakların yenilenebilir kaynaklar olarak sınırlandırılması yeni ürün geliştirme projeleri için geçerli ve diğer yayınlarda da sıklıkla kullanılan bir varsayımdır (Wang, 1999; Wang, 2002; Wang, 2005, Cho ve Eppinger,2005). Kaynakları çeşitli şekillerde sınıflandırmak mümkündür. Zaman bazına göre sınıflandırıldığında, yenilenebilir kaynaklar, bir zaman birimi içerisindeki toplam kullanılabilir miktarı kısıtlı olmakla birlikte kullanım ile tükenmeyen kaynaklardır. Yeni ürün geliştirme projelerinde sıklıkla kullanılan işgücü, test laboratuvarları ya da cihazları, kullanılan yazılım ve donanım cihazları bunlara örnek olarak verilebilir. Kaynak kısıtları ile ilgili bu varsayımların, yeni ürün geliştirme projelerinin çizelgelenmesi ile ilgili yapılan çalışmalarda varsayımlarla uyumlu olduğu görülmüştür. Bu çalışmalarda, yeni ürün geliştirme projelerinde kullanılan kaynakların genellikle, mühendis, laboratuvar, bilgisayar, yazılım gibi yenilenebilir kaynaklar olduğu özellikle vurgulanmıştır.

Üst üste çakıştırma yapılırken, alt seviye faaliyetin başlayabilmesi için üst seviye faaliyetten ön bilginin alt seviye faaliyet ulaşması gerekir bu nedenle alt seviye faaliyet üst seviye faaliyet başlamadan başlayamaz. Aynı şekilde alt seviye faaliyet ancak üst seviye faaliyetten son bilgiyi alıp düzeltmeleri yaptıktan sonra bitirilebilir. (Cho ve Eppinger,2005; Roemer ve ark.,2000).

Yapılan çalışmada, yeniden çalışma miktarının belirlenmesinde, Krishan(1997) çalışmasında tanımladığı her iki faktörde dikkate alınmıştır. Bu faktörlerden biri olan üst seviye faaliyetin gelişim fonksiyonu, Wang ve Lin (2009)'in çalışmalarında ortaya koydukları üst seviye faaliyetin gelişim fonksiyonu denkleminde faydalanılarak üst üste çakıştırma miktarının hesaplanmasına dahil edilmiştir. Alt seviye faaliyetin duyarlılığı için ise Cho ve Eppinger (2001)'in çalışmalarında da kabul ettiği üzere alt seviye faaliyetin süresinin belirli bir yüzdesi

olarak yeniden çalışma miktarına etki ettirilmiştir. Bu sayede Cho ve Eppinger (2001) çalışmasındaki gibi yalnızca alt seviye faktörün duyarlılığını dikkate alarak yeniden çalışma miktarının hesaplanmasına göre etki eden faktörlerin her ikisini de dikkate alan daha gerçekçi bir denklem ortaya konmuştur. Ancak, modelin analitik çözümünde, α katsayısı 1 olarak alınmıştır.

3.2.1.3. Girdi Değişkenleri ve Parametreler

Ele aldığımız problem, genel anlamda kaynak kısıtlı proje çizelgeleme problemi sınıfına girmesi nedeniyle karar değişkenleri iki grup altında ele alınabilir. İlk grup karar değişkenleri, faaliyetlerin başlangıç ve bitiş zamanlarını belirleyen karar değişkenleridir. Başlangıç zamanlarının belirlenmesi ardından belirli ve sabit olan faaliyet sürelerinin eklenmesi ile bitiş zamanlarına ait değişkenler kolaylıkla bulunabilmektedir. Başlangıç değişkeni, ikili değişken şeklinde modele dahil edilmiştir. Bunlara ilave olarak, bu çalışmada üst üste çakıştırma olması nedeniyle bu uygulamadan doğan alt seviye faaliyetin yeniden çalışma miktarı bu faaliyetin bitiş süresinin belirlenmesi gerekir. Bu nedenle, çalışmamıza özgü olarak yeniden çalışma miktarını belirleyen değişken de modele dâhil olmaktadır. Yeniden çalışma miktarı ise tamsayı bir değişken olarak modele eklenmiştir.

İkinci grup karar değişkeni ise proje süresince faaliyetlerin ihtiyaç duyacağı kaynakların nasıl dağıtılacağını belirleyen kaynak atama değişkenleridir. Belirli bir dönem içerisinde kullanım miktarı sınırlandırılmış olan yenilenebilir kaynakların faaliyetlere atanabilmesi gerekli olan atama değişkenleri tamsayı değişkenler olarak modele dâhil edilmiştir. Kaynakların hangi süreler içerisinde atandığı faaliyetlerde kullanılacağı ise, ikili değişken olarak tanımlanmış olan devam değişkenleri ile belirlenmektedir.

Model de kullanılan parametreler ise genel olarak; faaliyet süresi parametresi, mevcut kaynak miktarı parametresi, faaliyetlerin kaynak ihtiyacı parametresi, öncelik ilişkileri parametreleri ve üst üste çakıştırma etki faktörü olarak açıklanabilir.

3.2.1.4. Performans Değişkenleri

Öncelikle modelde yer alacak performans değişkenlerinin ve amaç fonksiyonun belirlenmesi amacıyla literatürde yer alan çalışmalar ve sürece özgü özellikler incelenmiştir. Yeni ürün geliştirme projelerinin çizelgelenmesi ile ilgili literatürde yer alan çalışmalar (Wang, 1999; Wang, 2002;2004;2005; Balasubramanian ve Grossman, 2003; Zhuang ve Yasinse, 2007) incelendiğinde zaman tabanlı amaç fonksiyonu olan proje bitiş süresinin en küçüklenmesi hedefinin seçildiği görülmektedir. Bu hedef sürecin spesifik özellikleri açısından oldukça uygundur. Bunun en önemli nedeni, dünya ticaretinin artması, bölgesel ya da ulusal pazar özelliklerinin dışına tasma ile rekabet yoğunlaşması ve uluslararası rakiplerin ortaya çıkması mevcut veya yeni pazarlarda her zaman rakipler olmasına neden olmaktadır. Sonuç olarak, rekabet avantajı açısından fark yaratan ürünleri mümkün olan en kısa sürede piyasaya sunmak çok önemlidir. Bu nedenle, geliştirilen modelin amaç fonksiyonu proje bitiş süresinin en küçüklenmesi olarak belirlenmiştir.

3.2.2. Matematik Programlama Modeli

3.2.2.1. Notasyon

Modelde kullanılan notasyon aşağıdaki şekildedir;

İndisler

i: Faaliyet indisi ($i=1, \dots, n$)

k: Faaliyet indisi ($i=1, \dots, n$)

t: Zaman indisi ($t=1, \dots, T$)

u: Zaman indisi ($u=1, \dots, T$)

j: Kaynak indisi ($j=1, \dots, m$)

dstrm: alt seviye faaliyetin indisinin yer aldığı liste

upstrm: üst seviye faaliyetin indisinin yer aldığı liste

Karar Değişkenleri:*Faaliyetin başlangıç zamanı değişkeni;*

$$s_{it} = \begin{cases} 1 & \text{faaliyet i hafta t'de basliyorsa} \\ 0 & \text{aksi halde} \end{cases}$$

Faaliyetin devam değişkeni;

$$dvm_{it} = \begin{cases} 1 & \text{faaliyet i hafta t'de devam ediyorsa,} \\ 0 & \text{aksi halde} \end{cases}$$

Faaliyetin başlama zamanı değişkeni; St_i :faaliyet i'nin başladığı zaman*Faaliyetin bitiş zamanı değişkeni;* Ft_i :faaliyet i'nin tamamlandığı zaman*Faaliyetin kullandığı kaynak miktarı değişkeni;* kkm_{ijt} : faaliyet i'nin t. haftada j. kaynaktan kullandığı miktar*Alt seviye faaliyette üst üste çakıştırılmadan dolayı oluşan yeniden çalışma miktarı değişkeni;* rw_{ik} : faaliyet “i” ile üst üste çakıştırılan alt seviye faaliyet “k” da oluşan yeniden çalışma miktarı irw_{ik} : faaliyet “i” ile üst üste çakıştırılan alt seviye faaliyet “k” da oluşan yeniden çalışma miktarının tamsayıli değeriGirdi Parametreleri: d_i : faaliyet i'nin süresi iht_{ij} : faaliyet i'nin devam edebilmesi için j kaynaktan ihtiyaç duyduğu miktar EM_j :kaynak j'nin bir dönemde kullanılabilecek miktarı onc_{ik} : öncelik parametresi; $onc_{ik} = \begin{cases} 1 & \text{faaliyet i faaliyet k'dan hemen önceyse} \\ 0 & \text{aksi halde} \end{cases}$

ovr_{ik} : üst üste çıkıştırma parametresi

$$ovr_{ik} = \begin{cases} 1 & \text{faaliyet i ve k arasında üst üste çıkıştırma varsa} \\ 0 & \text{aksi halde} \end{cases}$$

$ovrimp_k$: alt seviye faaliyet k'nın çıkıştırma etkisi faktörü

3.2.2.2. Matematiksel Model

Amaç Fonksiyonu:

$$\min z = \sum_{t=1}^T t.s_{nt} \quad (3.4)$$

Denklem (3.4) kullanılarak, proje ağının en sonuna eklenen kukla faaliyetin başlama süresi yani projenin bitiş süresi hesaplanmaktadır.

Başlama ve Bitiş Sürelerinin Hesaplanması:

$$St_i = \sum_{t=1}^T s_{it}.t \quad \forall i, t \quad (3.5)$$

$$Ft_i = St_i + d_i \quad \text{Eğer } i \notin \text{dstm} \text{ ise} \quad (3.6)$$

$$Ft_i = St_i + d_i + irw_{ik} \quad \text{Eğer } ovr_{ik}=1 \text{ ve } i \in \text{dstm} \text{ ise} \quad (3.7)$$

Denklem (3.5) bütün faaliyetler için başlama zamanını hesaplamaktadır. Denklem (3.6), alt seviye faaliyet haricindeki bütün faaliyetlerin bitiş süreleri hesaplarken, denklem (3.7) sadece alt seviye faaliyetin bitiş süresini hesaplamakta kullanılmaktadır.

Kaynak Kapasitesi Kısıtları:

$$kkm_{ijt} = iht_{ij}.dvm_{it} \quad \forall i, j, t \quad (3.8)$$

Denklem (3.8), i. faaliyetin, t. haftada, j. kaynaktan kullandığı miktarı hesaplamaktadır. Bu miktar, ihtiyaç duyduğu miktar ile faaliyetin t. zamanda devam edip etmediği gösteren ikili değişkenin çarpımı ile elde edilir. Daha sonraki aşamada ise bütün faaliyetlerin t. haftada j. kaynaktan kullandığı toplam miktar denklem (3.9) yardımıyla hesaplanır.

$$KM_{jt} = \sum_{i=1}^n kkm_{ijt} \quad \forall j, t \quad (3.9)$$

$$KM_{jt} \leq EM_j \quad \forall j, t \quad (3.10)$$

Son olarak, denklem (3.10), t. haftada j kaynaktan kullanılan miktarın o kaynağın mevcut miktarını geçmemesini sağlar.

Öncelik Kısıtları:

$$\sum_{u=t+d_i}^T s_{ku} \geq s_{it} \cdot onc_{ik} \quad \begin{matrix} \forall i, k \\ i \neq k \end{matrix} \quad (3.11)$$

Faaliyet i ve k arasında öncelik ilişkisi varsa, Denklem(3.11) kullanılarak, faaliyet k'nın faaliyet i'nin tamamlanmasından sonra başlaması garanti altına alınır.

Üst üste çıkıştırma olan faaliyetler için öncelik kısıtları:

$$St_k > St_i \quad \begin{matrix} \text{Eger } ovr_{ik} = 1 \text{ ise} \\ i \in \text{upstrm ve } k \in \text{dstrm} \end{matrix} \quad (3.12)$$

$$St_k \geq St_i + d_i - d_k \quad \begin{matrix} \text{Eger } ovr_{ik} = 1 \text{ ise} \\ i \in \text{upstrm ve } k \in \text{dstrm} \end{matrix} \quad (3.13)$$

Faaliyet i ve faaliyet k arasında üst üste çıkıştırma yapılması düşünülüyorsa, bu durumda faaliyet i (üst seviye faaliyet) başlamadan faaliyet k (alt seviye faaliyet) başlayamaz. Bu durum, denklem (3.12) ile sağlanmaktadır. Denklem (3.13) ise, üst seviye faaliyet bitmeden alt seviye faaliyetin bitmeyeceğini garanti altına alır.

Faaliyet Devam Kısıtları:

$$dvm_{iu} \geq s_{it} \quad i \notin \text{dstrm}, \forall t, u \geq t \text{ ve } u \leq t + d_i - 1 \quad (3.14)$$

$$\sum_{t=1}^T dvm_{it} = d_i \quad i \notin \text{dstrm}, \forall t \quad (3.15)$$

Denklem (3.14), alt seviye faaliyet dışındaki diğer bütün faaliyetlerin işlem gördüğü süre boyunca devam değişkeninin “1” değerini almasını sağlar. Denklem

(3.15), toplam devam değişkeni sayısını belirler ve bu sayının faaliyetin süresine eşit olmasını garantiler.

Alt Seviye Faaliyet için Devam Kısıtları:

$$dvm_{ku} \geq s_{kt} \quad k \in dstrm, \forall t, u \geq t \text{ ve } u \leq t + d_i + irw_{ik} - 1 \quad (3.16)$$

$$\sum_{t=1}^T dvm_{kt} = d_k + irw_{ik} \quad (3.17)$$

Eğer üst seviye faaliyet “ i ” ve alt seviye faaliyet “ k ” arasında üst üste çıkıştırma yapılıyor ise, alt seviye faaliyetin devam değişkenin hesaplanması için (3.16) ve (3.17) numaralı denklemler kullanılmalıdır. Bu denklemler, alt seviye faaliyetin yeniden çalışılmasından kaynaklanacak sürenin de devam değişkenine ilave edilebilmesini sağlar.

Yeniden çalışma miktarının hesaplanması:

Yeniden çalışma miktarının hesaplanması için sırasıyla Denklem(3.18) ve Denklem(3.19) kullanılmaktadır.

$$rw_{ik} = \left(1 - \left[\frac{St_k - St_i}{d_i}\right]\right).ovrimp_k \quad \text{eğer } i \in upstrm \text{ ve } k \in dstrm \quad \text{ve } St_k \leq Ft_i \text{ ise} \quad (3.18)$$

$$rw_{ik} = 0 \quad \text{eğer } i \in upstrm, k \in dstrm \text{ ve } St_k > Ft_i \text{ ise} \quad (3.19)$$

$$irw_{ik} \geq rw_{ik} \quad ovrip_{ik} = 1 \quad \text{ise} \quad (3.20)$$

$$irw_{ik} \leq rw_{ik} + 1 \quad ovrip_{ik} = 1 \quad \text{ise} \quad (3.21)$$

irw_{ik} değişkeni, modelde tamsayıli değişken olarak tanımlanmıştır. Bu şekilde, denklem (3.20) ve denklem (3.21) kullanılarak, hesaplanan yeniden çalışma miktarı tamsayıli hale getirilmektedir

$$\sum_{t=1}^T s_{it} = 1 \quad \forall i, t \quad (3.22)$$

Denklem (3.22), bir faaliyetin ancak belirli bir t zamanında başlayabileceğini garanti altına alır. İkili değişken kısıtları ve değişkenlerin pozitif değer alma kısıtları Denklem (3.23) ve Denklem (3.24) ile matematiksel olarak ifade edilmiştir.

$$s_{it}, dvm_{it} = \begin{cases} 1 \\ 0 \end{cases} \quad \forall i, t \quad (3.23)$$

$$kkm_{ijt}, s_{it}, dvm_{it}, rw_{ik}, St_i, Ft_i, irw_{ik} \geq 0 \quad \forall i, j, t \quad (3.24)$$

3.2.3. Kullanılan İleri Sezgisel Karar Verilmesi

Ele alınan problem için, kesin çözüm yöntemleri ile çözüm sağlanabilen problem büyüklüklerinin sınırlı olması, büyük ve karmaşık projeler için en iyi çözüm olmasa da en iyiye yakın çözüme hızlı bir şekilde ulaşan sezgisel algoritmaların geliştirilmesine neden olmuştur.

Sezgisel yöntemlerin sonuçları aynı zamanda kesin çözüm yöntemleri için bir alt veya bir üst sınır oluşturmak amacıyla da kullanılmaktadır. Ele alınan problemin genel anlamda kaynak kısıtlı proje çizelgeleme başlığı altında olması nedeniyle ilk olarak kaynak kısıtlı proje çizelgeleme problemlerinin çözümü için kullanılan sezgisel yöntemler ve ileri sezgisel yöntemler araştırılmıştır. Bu aşamadan sonra kullanılacak ileri sezgisel yöntem belirlenmiş ve probleme uyarlaması gerçekleştirilmiştir.

Kaynak kısıtlı proje çizelgeleme problemlerinin (KKPÇP) çözümü için kullanılan sezgisel yöntemler genel olarak literatürde dört ana başlık altında incelenmektedir. Birinci grup sezgisel algoritmalar, öncelik kurallarına dayalı tek çözüm üreten sezgisel yöntemlerdir. İkinci grupta ise, öncelik kuralları kullanarak veya örnekleme yolu ile birden çok uygulama yapılır ve birden çok çözüm elde edilir. Yaygın bir şekilde kullanılmakta olan ileri sezgisel yöntemler ise diğer bir grubu oluşturmaktadır.

Kısıtlı kaynak kullanımlı projelerde proje süresinin en küçüklenmesi problemini çözmek amacıyla son yıllarda kullanılan ve oldukça başarılı sonuçlar

alınan yöntemlerden arasında ileri sezgisel (*Meta-heuristic*) yöntemler yer almaktadır.

İleri sezgisel yöntemler, Genetik Algoritmalar (*Genetic Algorithms*), Tavlama Benzetimi Yöntemi (*Simulated Annealing Method*), Tabu Tanımlayarak Arama Yöntemi (*Tabu Search Method*), Karınca Kolonisi yöntemi (*Ant Colony Optimization*), ve Parçacık Sürü Optimizasyonu (*Particle Swarm Optimization*) başlıkları altında toplanabilir. Bunlara ek olarak, son yıllarda, KKPÇP’ni için farklı sezgisel yöntemlerin birlikte kullanıldığı hibrit yöntemlerde geliştirilmiştir (Agarwal ve ark, 2011; Chen ve ark., 2010; Valls ve ark, 2008).

Bir evrimsel hesaplama yöntemi olan Genetik Algoritma (GA), Kaynak Kısıtlı Proje Çizelgeleme problemlerinin çözümü için yaygın bir şekilde kullanılmıştır. Problem çözümü genlerden oluşan bir kromozom ile ifade edilir ve bir dizi çözümün oluşturduğu bir nesil ile (generation) algoritma başlatılır. Mevcut nesil üzerindeki kromozomlara kopyalama (reproduction), çaprazlama (crossover) ve mutasyon (mutation) operatörleri uygulanarak yeni bir nesil oluşturulur. GA’nın yapısındaki temel güncelleme mekanizması, algoritmanın yerel en iyiye takılmasını engelleyerek global en iyinin araştırılmasını sağlar. Fakat, GA yaklaşımındaki temel eksiklikler çözüme yavaş ilerleme (premature convergence) yani çözüme ulaşmak için çok fazla nesiller gerektirmesidir. (Fogel, 1995; Zhang ve ark. 2006’dan).

Parçacık sürü optimizasyonu (PSO), kuş sürülerinin davranışlarından esinlenerek ortaya çıkarılmış, topluluk tabanlı bir tekniktir. PSO, GA gibi evrimsel hesaplama teknikleri ile birçok benzerlikler gösterir. Sistem rasgele çözümler içeren bir topluluk ile başlatılır ve nesilleri güncelleyerek en iyi çözümü araştırır. Fakat PSO herhangi bir evrim operatörü kullanmaz. PSO’da parçacık olarak adlandırılan olası muhtemel çözümler, kendi en iyi deneyimlerini ve bütün sürünün en iyi deneyimini izleyerek problem uzayında dolaşırlar. Diğer analitik ve genel sezgisel metotlardan farklı olarak, PSO hesaplama açısından oldukça etkin ve yerel en iyiden sıçrayabilme açısından da oldukça başarılıdır. (Eberhard ve Shi,1998; Zhang ve ark. 2006’dan). Bunlara ek olarak, PSO, uygulama kolaylığı, daha hızlı araştırma prosesi ve etkili performans açısından GA’dan üstündür. (Eberhard ve Shi,1998, Robinson

ve ark., 2002: Zhang ve ark. 2006'dan). PSO algoritmasında, ayarlanması gereken parametre sayısının az olması nedeniyle de uygulaması oldukça basittir.

Bahsedilen bu avantajları nedeniyle ele alınan problemin çözümünde PSO'nun kullanılmasına karar verilmiştir. Bu nedenle, daha sonraki bölümlerde sırasıyla, genel PSO algoritması, ele alınan problem için kullanılan PSO gösterimleri, ve son olarak problemin çözümü için önerilen farklı algoritmalara yer verilecektir.

3.2.4. Parçacık Sürü Optimizasyonu Algoritması

Daha öncede belirtildiği gibi yeni ürün geliştirme projelerinin çizelgelenmesi problemi sınırlı sayıda yenilenebilir kaynakların yer aldığı bir proje olarak tanımlanmaktadır ve bu sebeple genel anlamda kaynak kısıtlı proje çizelgeleme problemi olarak ele alınmaktadır. kaynak kısıtlı proje çizelgeleme problemi literatürde “*NP-hard*” olarak tanımlanmaktadır. Proje ağının büyüklüğü ve özellikle kaynak kısıtlılığı arttıkça, problemin çözümü için kesin çözüm yöntemleri yetersiz kalmakta veya çözüm süresi çok uzun olmaktadır. Bu nedenle, eniyi çözüme yakın çözümler üreten ileri sezgisel metotlar kullanılması uygun görülmektedir.

Kullanılacak olan ileri sezgisel metodun seçilmesi için literatürde kaynak kısıtlı proje çizelgeleme probleminin çözümünde kullanılan ileri sezgisel metotlar incelenmiştir. Kullanılan başlıca sezgisel metotlar; GA, Tabu Arama Algoritması, Karınca Kolonisi Algoritması, Parçacık Sürü Optimizasyonu algoritmasıdır. PSO'nun ayarlanacak parametre sayısının az olması, uygulama kolaylığı sağlaması, daha hızlı araştırma prosesine sahip olması ve etkili performans sunması açısından GA'dan üstün olduğu belirtilmiştir (Eberhard ve Shi,1998, Robinson ve ark., 2002: Zhang ve ark. 2006'dan). Ayrıca PSO, hesaplama açısından oldukça etkin ve yerel eniyiden sıçrayabilme açısından da oldukça başarılıdır. Bu nedenle ele alınan problemin çözümünde PSO algoritmasının kullanılmasına karar verilmiştir. Bu karardan sonra PSO algoritmasının KKPÇ'ye adapte edilmesi için kullanılan gösterim yöntemleri ve kullanılan PSO prosedürleri detaylı olarak incelenmiştir.

Ele alınan problemin çözümü için, öncelik temelli gösterim ve probleme uyarlanmış seri çizelgeleme prosedürüne dayalı bir PSO algoritması geliştirilmiştir.

Daha sonrasında geliştirilen algoritma birden fazla faaliyet çiftleri arasında çakıştırmaya imkan verecek şekilde genişletilmiştir.

Bu bölümde öncelikle, genel olarak PSO algoritması hakkında bilgi verildikten sonra, KKPÇ’de PSO algoritmasının nasıl kullanıldığına değinilecektir. Sonrasında ise, yeni ürün geliştirme projelerinde süreyi kısaltmak amacıyla sıklıkla kullanılan bir uygulama olan üst üste çakıştırma uygulamasının da çizelgeleme prosesine dahil edildiği problemimizin çözümü için geliştirilen algoritmaya yer verilecektir.

3.2.4.1. KKPÇP’de Parçacıkların Temsili

Kennedy ve Eberhart (1995) tarafından geliştirilmiş olan PSO, kuş sürülerinin davranışlarının bir benzetimidir. Kuşların yerlerini bilmedikleri yiyecekleri aramaları, bir probleme çözüm aramaya benzetilir. Kuşlar yiyecek ararken yiyeceğe en yakın olan kuşu takip ederler. Parçacık olarak adlandırılan her tekil çözüm, arama uzayındaki bir kuştur. Parçacık hareket ettiğinde kendi koordinatlarını bir fonksiyona gönderir ve böylece parçacığın uygunluk değeri ölçülmüş olur. Bir parçacık, konumunu (position), hızını (velocity), şimdiye kadar elde ettiği en iyi uygunluk değerini, ve bu değeri elde ettiği konumu hatırlamalıdır. Çözüm uzayındaki her boyuttaki hızının ve yönünün her seferde nasıl değişeceği, bütün topluluk içindeki en iyi değeri sağlayan konum ve kendi en iyi konumunun bir birleşimi olacaktır.

PSO, bir grup rasgele çözüm ile (parçacık sürüsü) başlatılır ve güncellemelerle en iyi çözüm bulunmaya çalışılır. Her yinelemede (iterasyonda) parçacık konumları, iki en iyi değere göre güncellenir. İlki; o ana kadar parçacığın elde ettiği en iyi çözümü sağlayan konumdur. Bu değer “pbest” olarak adlandırılır ve hafızada saklanmalıdır. Diğer en iyi değer ise, toplulukta o ana kadar tüm parçacıklar tarafından elde edilen en iyi çözümü sağlayan konumdur. Bu değer global en iyidir ve “gbest” olarak adlandırılır. Örneğin, N boyutlu uzayda nokta olarak tanımlanan m adet parçacık sürüsü olduğunu kabul edelim. Bu durumda i. parçacık $x_i = [x_{i1}, x_{i2}, \dots, x_{iN}]$ ile ifade edilir. Önceki en iyi uygunluk değerini veren i’inci parçacığın pozisyonu $pbest_i = [p_{i1}, p_{i2}, \dots, p_{iN}]$ olarak ifade edilir. *Gbest* ise her

yinelemede tüm parçacıklar için tektir ve $gbest = [p_1, p_2, \dots, p_N]$ şeklinde gösterilir. Parçacıkların hızı yani her boyuttaki konumunun değişim miktarı $v_i = [v_{i1}, v_{i2}, \dots, v_{iN}]$ olarak ifade edilir. İki en iyi değer bulunmasından sonra parçacık hızları ve konumları aşağıda verilen (3.25) ve (3.26) nolu denklemler ile güncellenir (Kennedy ve Eberhard, 1995; Zhang ve ark. 2006'dan);

$$v_i(t) = w(t)v_i(t-1) + c_1 r_1 (pbest_i - x_i(t-1)) + c_2 r_2 (gbest - x_i(t-1)) \quad (3.25)$$

$$x_i(t) = v_i(t) + x_i(t-1) \quad (3.26)$$

Denklem (3.25)'de c_1 ve c_2 öğrenme faktörleridir. Bu değerler, her parçacığı $pbest$ ve $gbest$ pozisyonlarına doğru çeken hızlanma terimlerini ifade eden sabitlerdir. c_1 , parçacığın kendi tecrübelerine göre hareket etmesini, c_2 ise sürüdeki diğer parçacıkların tecrübelerine göre hareket etmesini sağlar. Yapılan çalışmalar neticesinde $c_1 = c_2 = 2$ olarak almanın iyi sonuçlar verdiği görülmüştür. r_1 ve r_2 ise $[0,1]$ arasında rasgele sayılardır. $w(t)$ ise önceki hız değerinin mevcut hız üzerindeki etkisini kontrol etmek amacı ile kullanılan süredurum ağırlık (inertia weight) parametresidir. Denklemlerde t , yineleme sayısını ifade etmektedir. Denklem (3.25), parçacığın bir önceki hız değeri ve mevcut pozisyonun $pbest$ ve $gbest$ den olan uzaklığına göre parçacığın yeni hızını hesaplamaktadır. Denklem (3.26) ise parçacığın kendi deneyimi ($pbest$) ve bütün parçacıkların en iyi deneyimini ($gbest$) dikkate alarak parçacığın yeni konumunu hesaplamaktadır.

PSO, özellikle GA ile bir çok benzer özellik taşımaktadır. Örneğin, başlangıç topluluğunun rasgele üretilmesi, en iyi değere her yinelemede güncelleme ile ulaşmaya çalışması, uygunluk değerinin mümkün çözümler için değerlendirilmesi bu benzerlikler arasında sayılabilir. Ancak, her iki algoritmanın bilgi paylaşım mekanizmaları birbirinden oldukça farklıdır. GA'daki bütün kromozomlar sınıflandırılmamış bütün araştırma tecrübelerini birbirleri ile paylaşırlarken, PSO parçacıkları yalnızca kendilerinin en iyi deneyimlerini (lokal en iyi) ve bütün parçacıklar arasındaki en iyi değeri (global en iyi) paylaşırlar.

PSO algoritmasının bir probleme uygulanmasına karar verildikten sonra parçacıklar için uygun gösterimin belirlenmesi gerekir. Genel olarak, kaynak kısıtlı proje çizelgeleme probleminin çözümünde PSO algoritması kullanılacak ise iki tip

gösterim kabul görmüştür. Bunlar, faaliyet listesi gösterimi (permutation based) ve öncelik temelli yada diğer bir adıyla rasgele anahtar (priority based/random key based) gösterimidir (Zhang ve ark., 2005).

Faaliyet listesi gösteriminde bir faaliyetin gösterimdeki yeri o faaliyetin çizelgelenme sırasını temsil ederken, öncelik temelli gösterimde parçacığın her bir elementi sabit olarak bir faaliyeti temsil etmekte ve bu element içindeki değerler o faaliyetin çizelgelenme önceliğine karşılık gelmektedir.

Faaliyet listesi gösteriminde parçacıklar faaliyet listesi şeklinde temsil edilirler. Faaliyetlerin gösterimdeki sıraları çizelgelenme esnasındaki önceliğini ifade etmektedir. Bu nedenle faaliyet listesi gösterimi aslında faaliyetlerin başlama sıralarını göstermektedir. Öncelik kısıtlarının sağlanabilmesi için, gösterimdeki bir faaliyet ancak hemen kendinden önceki faaliyetlerden sonra yer alabilir. Daha öncede belirtildiği gibi, PSO'daki her bir parçacık, çok boyutlu parametre uzayındaki mümkün çözümleri temsil eden N boyutlu veya elamanlı bir dizi ile kodlanır. Bu durumda N adet faaliyet içeren bir projenin çözümlerinin temsil edilmesi için faaliyetlere 1 den N a kadar indisler verilir. Bu durumda, N boyutlu parçacık, faaliyet listesini temsil eder. Parçacığın her bir elementi bir faaliyetin indeks numarasını içerir ve bu elementin yeri faaliyetin çizelgelenme sırasını gösterir. Bu durumda, faaliyet listesi gösteriminde parçacıkların bütün elementleri tamsayı bir değerdir. Buna ek olarak, bir elementin değeri $[1, N]$ arasında sınırlandırılmıştır. Yani $X^{\min} = 1$ ve $X^{\max} = N$ dir.

Örneğin, 20 faaliyeti içeren bir proje ağı için faaliyet listesi gösterimli iki parçacık Şekil 3.1' de verilmiştir (Zhang, 2005);

Parçacık 1	1	3	5	4	6		18	20	19
Parçacık 2	3	1	5	6	4		20	19	18

Şekil 3.1. Faaliyet listesi gösterim örneği

Öncelik temelli gösterim, GA'da kromozomların rasgele anahtar gösterimi ile temsil edilmesine benzemektedir (Chan ve Chua, 1996: Zhang ve ark. 1996'dan). Öncelik temelli gösterimde, PSO parçacıklarının N elementi projedeki N faaliyeti

temsil etmektedir. Böylece, i. parçacığın N boyutlu pozisyonunu tanımlayan N parametre, projedeki N adet faaliyetin önceliğidir. N adet parametrenin çok boyutlu parçacıktaki yeri ise sırasıyla faaliyetlerin indeks numaralarıdır. Bu durumda $X^{\min} = 0$ ve $X^{\max} = 1$ dir. Bu gösterime, 19 faaliyetli bir proje ağı için örnek Şekil 3.2' de verilmiştir(Zhang ve ark.,2005).

Parçacık 1	0.8	0.5	0.2	0.34	0.61		0.118	0.21	0.39
Parçacık 2	0.77	0.39	0.66	0.7	0.41		0.45	0.18	0.23

Şekil 3.2. Öncelik temelli gösterim örneği

Kendinden önce gelen faaliyetler bitmiş ve eldeki miktardan daha fazla kaynağa ihtiyaç duymayan birden fazla faaliyet olması durumunda öncelik değerlerini kullanarak sınırlı kaynağa göre faaliyetlerin çizelgelenme sırasını belirler. Böyle bir durumda en yüksek öncelik değerine sahip olan faaliyet ilk sırada çizelgelenir. Bir faaliyetin öncelik değerinin belirlenmesi için bir çok yöntem kullanılabilir. Öncelik değerlerinin belirlenmesi için kullanılan sezgisel kuralların birbirlerine göre herhangi bir üstünlükleri olmadığı yapılan çalışmalar neticesinde görülmüştür (Zhang ve ark. 2006: Davis ve Patterson,1975). Bu nedenle literatürde genellikle çoklu geçiş sezgiselleri veya önceliklerin rasgele üretilmesi yöntemlerine başvurulmuştur.

Kaynak Kısıtlı Proje Çizelgeleme çözümlerinin temsil edilmesi için hangi gösterim kullanılırsa kullanılsın bu gösterimleri çizelgeye dönüştürebilmek için Seri Çizelge Oluşturma Planları veya Paralel Çizelge Oluşturma Planlarından biri seçilmelidir. Öncelik temelli gösterim için her ikisi de uygunken, faaliyet listesi gösterimi için sadece Seri Çizelge Oluşturma Planları uygundur.

Öncelik temelli gösterimde, bazı farklı öncelik setleri aynı çizelgeye karşılık gelebilmektedir. Çünkü öncelik değerlerindeki bazı küçük değişiklikler bu değerlerin büyükten küçüğe doğru sıralamasını değiştirmeyebilmektedir. Fakat aynı durum faaliyet listesi gösterimi içinde geçerlidir. Bu gösterimde de bazen farklı parçacıklar

aslında aynı listeye karşılık gelebilir ve bu durumda çözüm uzayı aslında gerçek boyutundan daha büyük olabilir.

Öncelik temelli parçacık temsili, faaliyet listesi gösteriminden farklı olarak olurlu olmayan bir çizelge içermez (Chan ve ark.,1996; Hartmann, 1998; Bean, 1994; Zhang ve ark. 2005'den). Faaliyet listesi gösterimi seçildiğinde, parçacıkların güncellenme mekanizması olurlu olmayan çözümleri elimine edecek şekilde değiştirilmelidir. Zhang ve ark.(2005), faaliyet listesi gösterimini kullandıkları PSO algoritmasında parçacıkları olurlu hale getirecek bir mekanizma önermişlerdir.

Ele alınan problemin çözümü için, parçacıkların öncelik temelli gösterim ile temsil edildiği ve çizelge oluşturma planı olarak üst üste çakıştırmaya olanak verecek şekilde değiştirilmiş seri çizelgelemenin kullanıldığı bir algoritma geliştirilmiştir. Sonrasında bu algoritma, birden fazla faaliyet çiftleri arasın arasında çakıştırmaya imkan verecek şekilde genişletilmiştir.

3.2.4.2. Çizelge Oluşturma Planları

Çizelge oluşturma planları, kısıtlı kaynaklarla proje çizelgelenmesi problemlerine ilişkin sezgisel çözüm yöntemlerinin temelini oluşturmaktadır. Çizelge geliştirme planları, projenin başlangıç konumundan başlayarak, kısmi bir çizelgenin parçalarının aşamalı olarak arttırılması yoluyla çizelge üretmektedir. Çizelge parçası ise, $(n+2)$ faaliyetten oluşan projenin faaliyetler alt kümesidir. Kısmi çizelge, $(n+2)$ faaliyetin alt kümesinin çizelgelendiği bir çizelgedir.

Çizelge oluşturma planları, faaliyet ve zaman artışına göre birbirinden farklılık göstermektedir. Seri çizelge oluşturma planı olarak adlandırılan tür, faaliyet artışına dayanırken, paralel çizelge oluşturma planı zaman artırımını gerçekleştirir. Bu bölümde, algoritmada temel alınan seri çizelge oluşturma planı açıklanmaktadır.

Kaynak kısıtlı proje çizelgeleme için seri çizelge oluşturma planları Kelley (1963) tarafından geliştirilmiştir. Bu yöntem tam olarak n adımda bir olurlu çözüm oluşturur. Her bir yinelemede mevcut kısmi çizelgeye bir faaliyetin eklenmesi ile çizelge genişletilir. Faaliyet seçimi, öncülleri tamamlanmış olan seçilebilir faaliyetler kümesinden belirli bir öncelik kuralına dayalı olarak seçilir ve kaynak açısından

uygulanabilir en erken tamamlanma süresi içerisinde çizelgelenir. Her bir g adımımda üç ayrı küme bilgisi güncellenir. Bunlar; çizelgelenen faaliyetler kümesi (Sec_g), çizelgelenebilir faaliyetler kümesi (D_g) ve çizelgelenen faaliyetlerin bitiş sürelerinin tutulduğu F_g kümesidir(Torres ve ark., 2010).

$\bar{R}_m(t)$, t zamanında m kaynağından mevcut miktarı göstermektedir ve aşağıdaki gibi hesaplanmaktadır;

$$\bar{R}_m(t) = RA_m - \sum_{\substack{j/S_j \leq t \\ f_j > t}} r_{jm} \quad \forall m \in M \quad (3.27)$$

Bu bilgilere dayalı olarak seri çizelge oluşturma planının algoritması aşağıdaki gibidir (Torres ve ark,2010);

Adım 1. Başlama

$$t_s(i) = t_f(i) = 0 \quad IP(i) = f \text{ ise}$$

$$Sec_1 = \{1\}$$

Adım 2 g=2 den n-1' e kadar t

Adım 2.1. D_g, F_g ve $\bar{R}_m(t)$ ($t \in F_g, m \in M$) hesaplanır

Adım 2.2. $i \in D_g$ 'den bir faaliyet seçilir.

$$\textbf{Adım 2.3.} t_{ES}(i) = \max_{k \in IP(i)} \{f_k\}$$

$$\textbf{Adım 2.4.} t_s(i) = \min \{t/t \geq t_{ES}(i), t \in F_g, r_{im} \leq R_m(t), m \in M, t \in [t, t + t_d(i)] \cap F_g\}$$

$$\textbf{Adım 2.5.} t_f(i) = t_s(i) + t_d(i)$$

$$\textbf{Adım 2.6} Sec_g = Sec_{g-1} \cup \{i\}$$

$$\textbf{Adım 3.} t_f(n) = \max_{k \in IP(i)} \{t_f(k)\}$$

3.2.5. Önerilen PSO Algoritması

Daha önceki bölümlerde de belirtildiği gibi öncelik temelli gösterimin seçilmesi durumunda, parçacığın çizelgeye dönüştürülmesi için uygun bir çizelge geliştirme planının kullanılması gerekir. Öncelik temelli gösterimin seçilmesi durumunda her iki çizelge oluşturma planı da kullanılabilir. Ancak ele alınan

problem, klasik KKPC'den farklı olarak faaliyetlerin üst üste çıkıştırma uygulamasını da içerdiğinden seçilen çizelge oluşturma planının literatürde yer aldığı şekilde kullanılması mümkün değildir. Seçilen çizelge oluşturma planının eğer üst üste çıkıştırma uygulaması avantajlı ise bu uygulamaya imkan verecek şekilde düzenlenmesi gerekmektedir. Her iki çizelge oluşturma planları üzerinde yapılan çalışmalar neticesinde, seri çizelge oluşturma planının, üst üste çıkıştırma uygulaması daha uygun olduğu öngörülmüştür. Bunun en önemli nedeni, paralel çizelgelemenin zaman ilerletme esasında göre işlemesi nedeniyle, üst üste çıkıştırmaya imkan verecek zaman aralığının kontrol edilmemesidir. Oysa, seri çizelge oluşturma planları faaliyet ekleme esasına uygun olarak çalışmaktadır ve seçilen faaliyet en erken başlama zamanından itibaren kaynak olurluluğu açısından denenerek uygun bir başlama zamanına atanmaktadır. Probleme uygun şekilde algoritma güncellenirken, alt seviye faaliyet için en erken başlama zamanının üst üste çıkıştırmaya imkan verecek şekilde hesaplanması mümkündür. Revize edilmiş seri çizelge oluşturma planı, belirli iki faaliyet arasındaki bütün üst üste çıkıştırmaların başlama zamanı senaryolarını ve bunlara ek olarak üst üste çıkıştırma olmaması alternatifini de dikkate alarak projenin bitişi açısından en uygun olan alternatifi seçme imkanı sağlayarak çizelgeyi oluşturmaktadır.

Bu bölümde öncelikle, kullanılan PSO prosedürü adım adım açıklanacak ve sonraki aşamada ise probleme uyarlanan seri çizelge oluşturma planı detaylı olarak verilecektir. Burada;

i: parçacık indisi

j: faaliyet indisi

p: faaliyet indisi

dstrm: alt seviye faaliyet indisinin yer aldığı liste

upstrm: üst seviye faaliyet indisinin yer aldığı liste

1. **Başlangıç:** M adet parçacığın başlangıç pozisyonu $X_i(0) = \{x_{i1}(0), x_{i2}(0), \dots, x_{iN}(0)\}$ ve başlangıç hızı $V_i(0) = \{v_{i1}(0), v_{i2}(0), \dots, v_{iN}(0)\}$ aşağıdaki denklemler kullanılarak rasgele olarak üretilir;

$$\begin{aligned}
x_{ij}(t) &= \text{random } \{0,1\} & \text{eger } j \notin \text{dstrm} & \text{ise} \\
x_{ij}(t) &= 1 & \text{eger } j \in \text{dstrm} & \text{ise} \\
v_{ij}(t) &= \text{random } \{-V_{\max}, V_{\max}\}
\end{aligned}$$

2. Başlangıç Parçacıklarının Değerlendirilmesi: Parçacıkların uygunluk değerlerinin, $pbest$ değerlerinin ve $gbest$ 'in belirlenmesi için, başlangıç parçacıkları revize edilmiş seri çizelgeye gönderilmelidir. Revize edilmiş seri çizelgelemeden gelen uygunluk değeri bilgilerine göre her bir parçacığın $pbest$ değerleri, ve bütün parçacıklar için en iyi uygunluk değerini veren $gbest$ belirlenir.
3. Parçacıkların Hız Değerlerinin Güncellenmesi ve Tamiri: Her bir parçacığın hız değeri Denklem (3.25) kullanılarak hesaplanır ve gerekiyorsa aşağıdaki gibi tamir edilir;

$$\text{Eğer } v_{ij}(t) > V^{\max} \text{ ise}$$

$$v_{ij}(t) = V^{\max}$$

$$\text{Eğer } v_{ij}(t) < -V^{\max} \text{ ise}$$

$$v_{ij}(t) = -V^{\max}$$

4. Parçacıkların Pozisyonlarının Güncellenmesi ve Tamiri: Parçacık pozisyonlarının güncellenmesi Denklem (3.26) kullanılarak gerçekleştirilir. Güncelleme işleminden sonra gerekiyorsa tamir işlemi gerçekleştirilir;

Eğer $j \notin \text{dstrm}$ ise (faaliyet alt seviye faaliyet değilse)

$$\text{Eğer } x_{ij}(t) > X^{\max} \text{ ise}$$

$$x_{ij}(t) = X^{\max}$$

$$\text{Eğer } x_{ij}(t) < -X^{\max} \text{ ise}$$

$$x_{ij}(t) = -X^{\max}$$

Aksi takdirde,

$$\text{Eğer } x_{ij}(t) \neq 1 \text{ ise}$$

$$x_{ij}(t) = 1$$

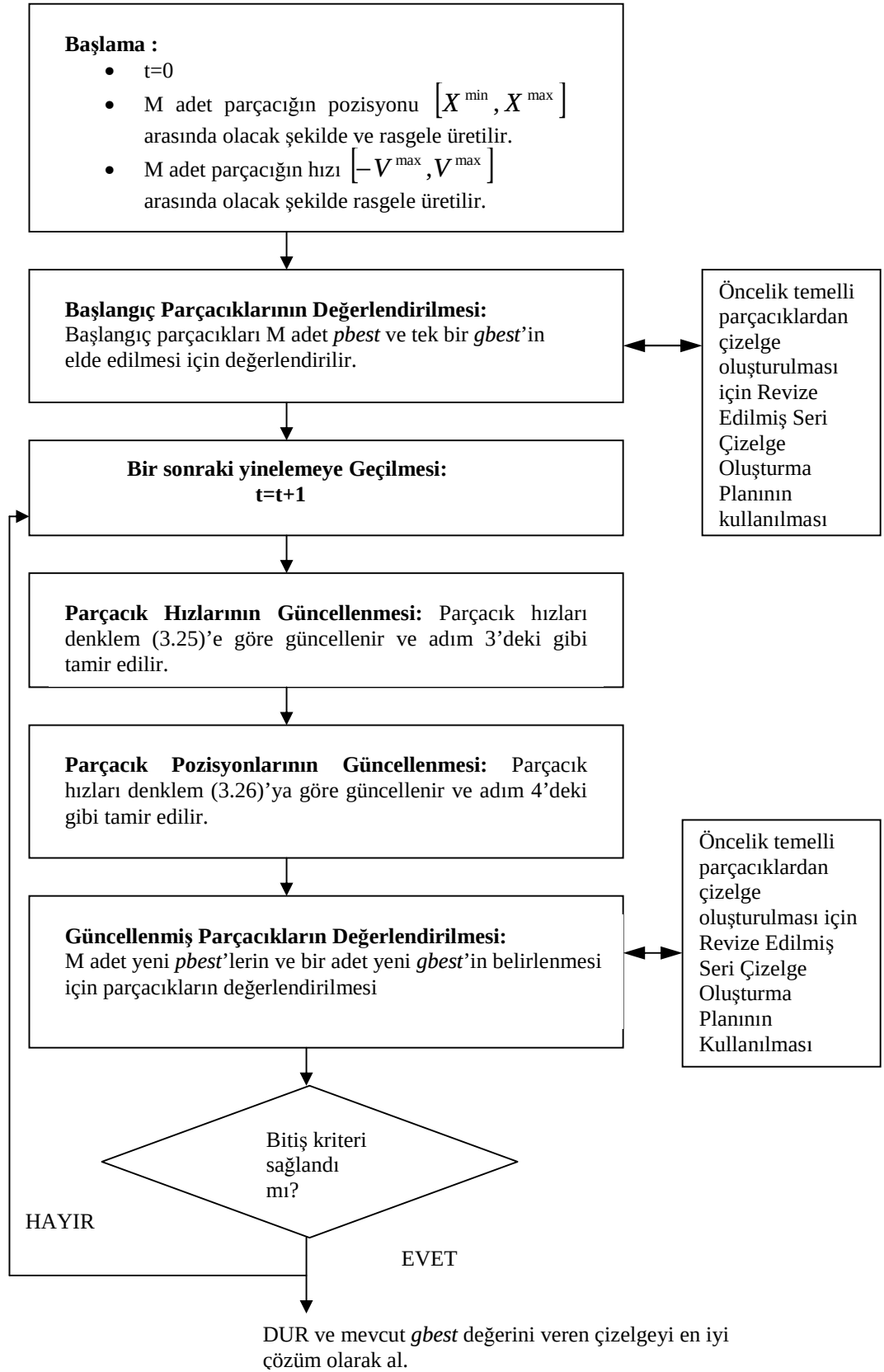
5. Güncellenmiş Parçacıkların Değerlendirilmesi: Güncellenmiş parçacıkların uygunluk değerlerinin, *pbest* değerlerinin ve *gbest*'in belirlenmesi için revize edilmiş seri çizelgeme algoritması kullanılarak önce parçacıklar çizelgeye çevrilir, sonrasında uygunluk değerleri hesaplanır. Hesaplanan uygunluk değerine göre *pbest* ve *gbest* bulunur.

6. Durma:

Eğer Maksimum yineleme sayısı kriteri sağlanmışsa durulur ve mevcut en iyi global değeri ve bunu sağlayan parçacığı en iyi olarak alınır.

Aksi Halde Adım 3'e gidilir ve devam edilir.

Algoritmanın görsel olarak ifadesi Şekil 3.3 ile verilmiştir.



Şekil 3.3.Öncelik temelli gösterimli ve seri çizelgeleme kullanan PSO Algoritması

Ele alınan problem için algoritmanın esas noktası, öncelik temelli gösterimlerin çizelgeye dönüştürüldüğü probleme uyarlanmış seri çizelge oluşturma planıdır. Literatürde yer alan ve KKPC problemi için geliştirilmiş olan seri çizelge oluşturma planları, belirli faaliyetler arasında üst üste çakıştırma durumunun değerlendirilmesine imkan vermemektedir. Bu nedenle seri çizelge oluşturma planları aşağı açıklandığı gibi revize edilmiştir. Burada;

j: faaliyet indisi

p: faaliyet indisi

s: faaliyet indisi

k: kaynak indisi

$t_{ESovr}(j)$: alt seviye faaliyet j'nin üst üste çakıştırma yapılabileceği en erken süre

$t_{ES}(j)$: faaliyet j'nin en erken başlama süresi

$t_d(j)$: faaliyet j'nin süresi

$t_s(j)$: faaliyet j'nin başlama süresi

IP(j): j faaliyetinden hemen önce gelen faaliyetlerin listesi

Adım 1. Başlama

$$t_s(j) = t_f(j) = 0 \quad IP(j) = f \text{ ise}$$

$$Sec_1 = \{1\}$$

Adım 2 g=2 den n-1' e kadar

Adım 2.1. D_g , F_g ve $\bar{R}_k(t)$ ($t \in F_g$, $m \in M$) hesaplanır

Adım 2.2. $j \in D_g$ 'den bir faaliyet seçilir.

Adım 2.3. En erken başlama sürelerinin hesaplanması: Seçilen faaliyet alt seviye faaliyetse Adım 2.3.1. ve 2.3.2. yapılarak Adım 2.4.'e geçilir. Aksi halde Adım 2.3.3.yapılarak Adım 2.7.'ye geçilir.

Eğer $j \in dstrm$ ise (seçilen faaliyet alt seviye faaliyet ise) en erken başlama süresi aşağıdaki gibi hesaplanır;

Adım 2.3.1. $t_{ESovr}(j)$ hesaplanır

Eğer $t_d(p) > t_d(j)$ ise ($p \in upstrm$ iken)

$$t_{ESovr}(j) = t_s(p) + t_d(p) - t_d(j)$$

Aksi Halde

$$t_{ESovr}(j) = t_{ES}(p) + 1$$

Adım 2.3.2. $t_{ES}(j)$ hesaplanır

$$t_{ES}(j) = \max_{\substack{s \in IP(j) \\ s \neq p}} \{t_{ES}(s) + t_d(s)\}, t_{ESovr}(j)\}$$

Sonrasında, Adım 2.4., 2.5., 2.6. sırayla yapılır.

Aksi Halde (seçilen faaliyet alt seviye faaliyet değilse)

Adım 2.3.3.

$$t_{ES}(i) = \max_{s \in IP(i)} \{t_{ES}(s) + t_D(s)\} \text{ hesaplanır ve Adım 2.7.'ye geçilir.}$$

Adım 2.4. Bu aşamada, üst üste çakıştırma mümkün ise üst üste çakıştırmanın bütün başlama süresi alternatifleri için alt seviye faaliyetin alternatif başlama ve bitiş süreleri bulunur ve sırası ile ALS (alternatif başlama süresi listesi) ve AFL(Alternatif bitiş süresi listesi) listelerinde tutulur. Bu işlemler adım adım aşağıda açıklanmıştır.

Eğer $j \in dstrm$ ve $p \in upstrm$ iken $t_{ES}(j) < t_f(p)$ ise (alt seviye faaliyetin en erken başlama süresi üst seviye faaliyetin bitiş süresinden küçük ise yani üst üste çakıştırma uygulaması mümkün ise Adım 2.4.1.'den 2.6.3'e kadar bütün adımlar gerçekleştirilir.)

Aksi Halde Adım 2.7.'ye geçilir.

Adım 2.4.1. Üst üste çakıştırmanın yapılabileceği zaman aralığı belirlenir;

$$l = [t_{ES}(j), t_f(p) - 1]$$

Adım 2.4.2. Alt seviye faaliyetin bu aralıktaki her bir t değerinde başladığı durumlar için yeniden çalışma miktarları (denklem 3.28) ve buna bağlı olarak alternatif bitiş zamanları (denklem 3.29) hesaplanır.

$$rw(j, t) = \left\lceil \left[1 - \frac{t - t_s(p)}{t_d(p)} \right] \cdot ovrimp(j) \right\rceil \quad (3.28)$$

$$t_{AF}(j, t) = t + rw(j, t) + t_d(j) \quad (3.29)$$

Adım 2.4.3. Her bir $t \in l$ için alt seviye faaliyetin başlama bitiş aralıkları yazılır.

$t = [t, t_{AF}(j, t)]$ ve her bir aralık için adım 2.4.4. gerçekleştirilir.

Adım 2.4.4. Her bir t aralığı için, bu aralıktaki her bir t değerinde kaynak miktarının faaliyetin ihtiyacını karşılayıp karşılayamadığı kontrol edilir;

Eğer her bir $t \in t$ için $r_{jm} \leq \bar{R}_m(t)$ ise mevcut t aralığının ilk elemanı ASL listesine, son elemanı ise AFL listesine sıra ile eklenir.

Aksi Halde yazılmadan geçilir.

Adım 2.5. Üst üste çakıştırma alternatiflerinin dikkate alınmadığı durumun değerlendirilmesi yapılır.

Adım 2.5.1. Üst üste çakıştırmanın yapılmadığı durum için alt seviye faaliyetin en erken başlama süresi hesaplanır.

$$t_{ES}(j) = \max_{k \in IP(j)} \{t_{ES}(k) + t_D(k)\}$$

Adım 2.5.2. Adım 2.5.1’de hesaplanan en erken başlama süresi dikkate alarak, kaynak ihtiyacını faaliyetin devam ettiği süre boyunca karşılayan en erken süre faaliyetin diğer bir alternatif başlama süresi olarak belirlenir. Bunun için öncelikle, faaliyetin en erken başlama süresinde başladığı düşünülerek, faaliyetin kaynak kullanacağı zaman aralığı içerisinde kaynağın mevcut olup olmadığı kontrol edilir. Eğer mevcut değilse, aynı işlemler F_g listesindeki $t_{ES}(j)$ ’den büyük en küçük eleman için yapılır. Bu işlem faaliyetin başlama süresinin atanabileceği bir zaman bulunana kadar yapılır.

Adım 2.5.3. Bu başlama süresine karşılık gelen bitiş süresi hesaplanır;

$$t_{AF}(j) = t(j) + t_d(j)$$

Adım 2.5.4. Hesaplanan başlama süresi ASL listesine, değere karşılık gelen bitiş süresi ise AFL listesine eklenir.

Adım 2.6. Bütün durumlar dikkate alınarak alt seviye faaliyetin başlama ve bitiş süreleri aşağıdaki gibi atanır.

Adım 2.6.1. $t_f(j) = \min\{t\} \quad t \in AFL$ (alternatif bitişlerin tutulduğu listede en küçük süre, alt seviye faaliyetin bitiş süresi olarak atanır).

Adım 2.6.2. $t_s(j) = \{ t \mid t \in \text{ASL} \Rightarrow \min\{t\}, t \in \text{AFL} \}$ (atanan bitiş süresine karşılık gelen ASL listesindeki başlama süresi, alt seviye faaliyetin başlama süresi olarak atanır).

Adım 2.6.3. Adım 2.9'a gidilir.

Adım 2.7. En erken başlama süresi hesaplanan faaliyetin başlama süresi, faaliyetin gerçekleştirildiği süre boyunca kaynak kısıtlarını da sağlayan en erken süreye atanır.

Adım 2.8. $t_f(j) = t_s(j) + t_d(j)$

Adım 2.9. $\text{Sec}_g = \text{Sec}_{g-1} \cup \{j\}$

Adım 3 $t_f(n) = \max_{p \in IP(n)} \{t_f(p)\}$

Bu algoritmanın şekilsel açıklaması EK 1'de verilmiştir.

3.2.6. Geliştirilen Algoritmaların Programlanması

Geliştirilen algoritmaların programlanması için C# programlama dili kullanılmıştır. C# programlama dili, Microsoft'un geliştirmiş olduğu yeni nesil bir dildir ve Microsoft tarafından .NET Teknolojisi için geliştirilmiş dillerden biridir. C#, güçlü, modern, nesne tabanlı ve aynı zaman tip-güvenli(type-safe) bir programlama dilidir. C#'ın en önemli avantajı, C++ dilinin profesyonelliğini ve Visual Basic' in ise kolaylığını sağlar. Visual Basic, kolay kullanılır bir dil olmasına rağmen C++'a göre yetenekleri azdır. Fakat ve C++ ile kodladığınız programlarınızın yazımı çok uzun sürebilir. C++'ın gücünü ve Visual Basic'in kolaylığını ve hızını birleştiren C#'dır.

Bu aşamada ilk olarak, üst üste çıkıştırma uygulamasını içermeyen algoritma programlanmıştır. Herhangi bir test problemi, ilk olarak bu program kullanılarak çalıştırılmakta ve üst üste çıkıştırma yapılacak faaliyetlerin belirlenmesinde bu programların sonuçları da dikkate alınmaktadır. Sonrasında ise, mevcut algoritmanın üst üste çıkıştırma uygulamasını içeren hali programlanmıştır.

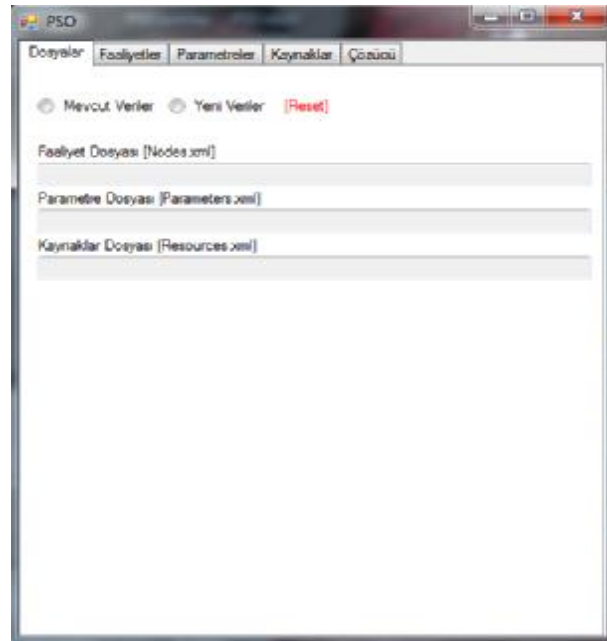
Yazılan programların kullanımında, kullanıcı kolaylığı sağlaması açısından proje verilerinin ve algoritma parametrelerinin girişlerinin yapılabilmesi ve programın

çalıştırılabildiği bir ara yüz oluşturulmuştur. Bu sayede programda kullanılması gereken veriler ve parametreler rahatlıkla girilebilmekte ve kullanıcı açısından kolaylık sağlamaktadır. Aşağıda sırasıyla geliştirilen programların ara yüzleri verilmiş ve nasıl kullanılacağı açıklanmıştır.

3.2.6.1. Algoritmaların Ara Yüzlerin Tanıtılması

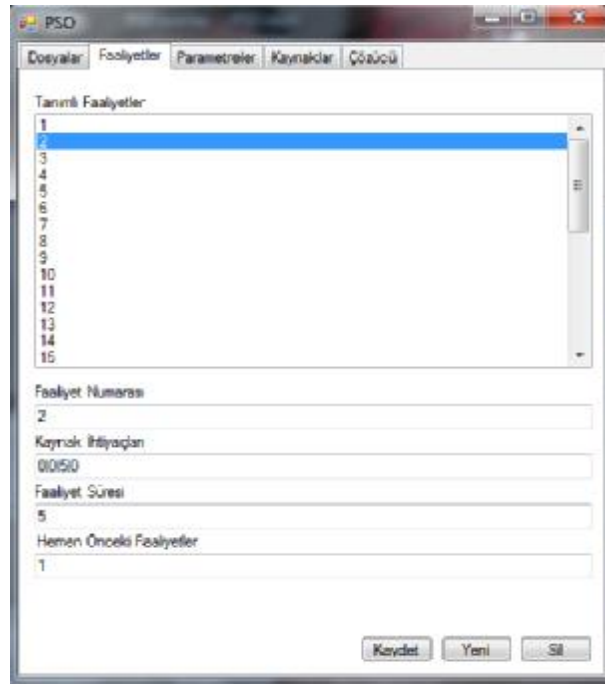
Programlara ait bazı farklılıklar olması nedeniyle programların ara yüzlerinde bazı değişiklikler olabilmektedir. Bu nedenle, bu bölümde üst üste çakıştırma uygulamasını içeren ve içermeyen programların ara yüzleri ayrı ayrı tanıtılmaktadır.

Şekil 3.4’ de, üst üste çakıştırma içermeyen programın ara yüzü verilmiştir. Görüldüğü üzere, programın ara yüzü; mevcut verilerin girilmesi veya yeni verilerin dosyalarının oluşturulabilmesi için “Dosyalar” bölümü, faaliyetlere ilişkin bilgilerin girilebilmesi veya değişikliklerin yapılabilmesi için “Faaliyetler” bölümü, parametrelerin girilebilmesi veya değişikliklerin yapılabilmesi için “Parametreler” bölümü, kaynaklara ilişkin kaynak sayısı ve mevcut miktarların girilebilmesi için “Kaynaklar” bölümü, ve son olarak programın çalıştırılması, sonuçların kaydedilebilmesi için “Çözücü” bölümlerinden oluşmaktadır.



Şekil 3.4. Üst üste çakıştırma içermeyen programın ara yüzü-Dosyalar bölümü-

Eğer, daha önceden hazırlanmış olan veriler, “Dosyalar” bölümündeki “Mevcut Veriler” butonu ile programa aktarılmış ise, her bir faaliyetler ilgili “Faaliyet Numarası”, Kaynak İhtiyaçları”, “Faaliyet Süresi”, “Hemen Önceki Faaliyetler” ve “Hemen Sonraki Faaliyetler” bilgilerine “Faaliyetler” bölümden ulaşılabilir (Şekil 3.5) İstenilirse bu bilgilerin herhangi birinde değişiklik yapılabilir ve “kaydet” butonuna basılarak kaydedilebilir. Ayrıca “Yeni” butonu kullanılarak yeni bir faaliyet eklenebilir veya herhangi bir faaliyet silinmek istendiğinde ilgili faaliyetin üstüne tıklandıktan sonra “Sil” butonu ile silinebilir. Eğer veriler yeni giriliyorsa faaliyetlere ilişkin belirtilen bilgilerin hepsinin girişi bu bölümde gerçekleştirilmektedir. Yeni dosyaların nasıl oluşturulduğu veri girişi bir sonraki bölümde açıklanmaktadır.



Şekil 3.5.Üst üste çıkıştırma içermeyen programın ara yüzü-Faaliyetler bölümü-

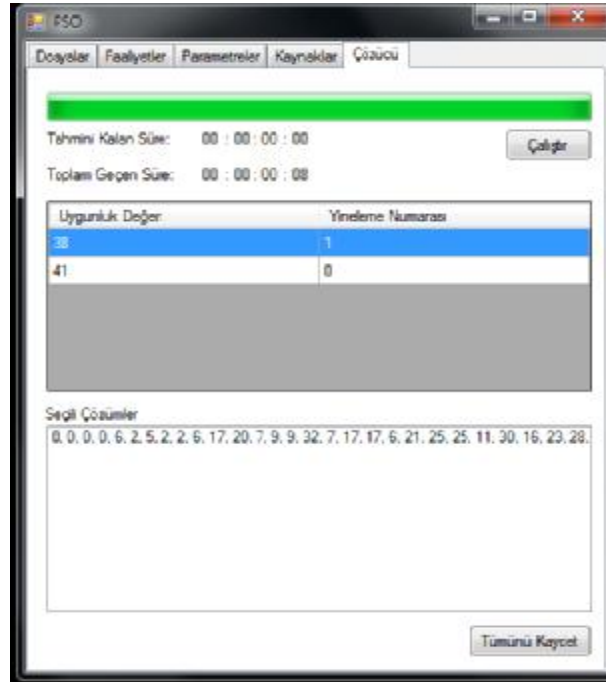
Dosya daha önceden hazırlanmış ise, Parametreler” bölümünde sırasıyla;, “Parçacık sayısı”, “Yineleme sayısı”, “ c_1 ”, “ c_2 ”, “ w ” katsayıları, “ V_{max} ”, “ $-V_{max}$ ” değerleri görülebilmektedir. Eğer yeni oluşturulan bir dosya ise bu bilgiler mevcut bölümde girilip kaydedilebilmektedir (Şekil 3.6).

Şekil 3.6. Üst üste çıkıştırma içermeyen programın ara yüzü-Parametreler bölümü-

Kaynaklar” bölümünde ise, her bir kaynağa ait “Kaynak Numarası” ve “Mevcut Kaynak Miktarı” bilgileri görülebilmektedir. Mevcut verilere ait kaynak bilgileri üzerinde de değişiklik yapıp kaydedilebilmektedir. Mevcut proje dosyası yeni oluşturuluyorsa, kaynak sayısı kadar yeni kaynak oluşturulup mevcut kaynak miktarı bu ekrandan girilmektedir(Şekil 3.7).

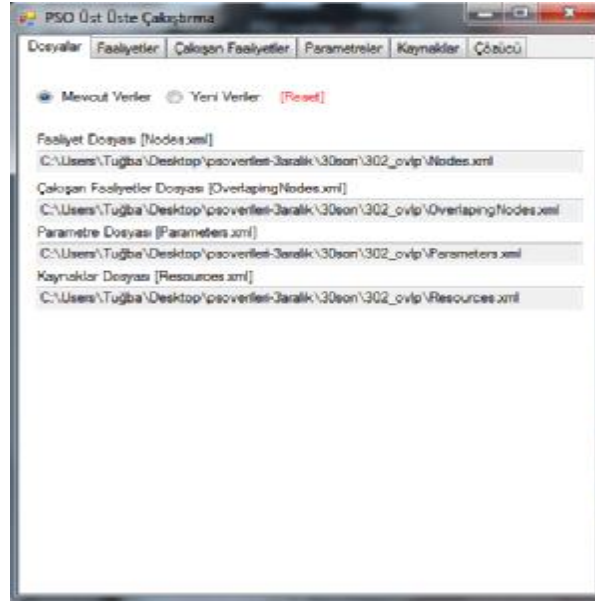
Şekil 3.7. Üst üste çıkıştırma içermeyen programın ara yüzü -Kaynaklar bölümü-

Son kısım olan “Çözücü” bölümünde ise, programın çalıştırılmasını sağlayan “Çalıştır” butonu ve “Tahmini Kalan Süre” ile “Toplam Geçen Süre” göstergeleri yer almaktadır. En altta ise “Seçili Çözümler” altında bırakılmış boş alana programın çalışması sırasında sürekli olarak güncellenen en iyi çözümler yazılmaktadır. “Tümünü Kaydet” butonu ise, programın çalışması bittikten sonra seçili bütün çözümlerin istenilen klasöre kaydedilmesi amacıyla kullanılmaktadır(Şekil 3.8).



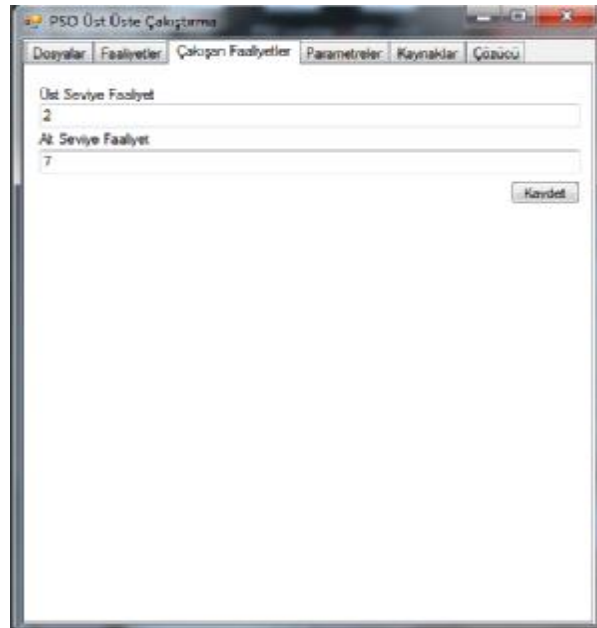
Şekil 3.8. Üst üste çakıştırma içermeyen programın ara yüzü-Çözücü bölümü-

Üst üste çakıştırma uygulamasını içeren programının ara yüzünde ise bahsedilen diğer programın ara yüzünden farklı olarak, “Dosyalar” bölümünde, “Çakışan Faaliyetler Dosyası” da yer almaktadır. Bu dosya içerisinde, hangi faaliyet çifti arasında üst üste çakıştırma yapılacağı bilgiler yer almaktadır(Şekil 3.9).



Şekil 3.9. Üst üste çakıştırma uygulamasını içeren programın ara yüzü-Dosyalar bölümü-

Ayrıca, diğer programın ara yüzünden farklı olarak, “Çakışan Faaliyetler” adı altında bir bölüm yer almaktadır. Bu bölümünde, üst üste çakıştırma yapılması istenilen üst ve alt seviye faaliyetlerin numaralarının girilmesi için sırasıyla, “Üst Seviye Faaliyet” ve “Alt Seviye Faaliyet” olarak isimlendirilmiş alanlar altında boşluklar bulunmaktadır (Şekil 3.10).



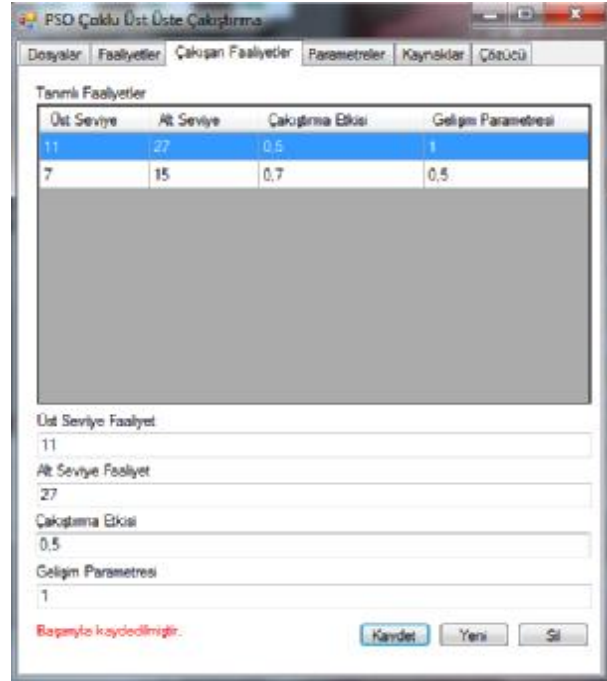
Şekil 3.10. Üst üste çakıştırma uygulamasını içeren programın ara yüz-Çakışan faaliyetler bölümü

Diğer programdan farklı olarak üst üste çıkıştırmanın neden olduğu yeniden çalışma miktarının hesaplanması için girilmesi gereken çıkıştırma etkisi ve gelişim parametresi ise, “Parametreler” bölümünde girilmekte veya istenilen değişiklikler burada yapılarak kaydedilebilmektedir (Şekil 3.11)

Şekil 3.11. Üst üste çıkıştırma uygulamasını içeren programın ara yüzü-Parametreler bölümü-

Mevcut algoritmanın sonrasında birden fazla faaliyet çifti arasında üst üste çıkıştırmaya imkan verecek şekilde genişletildiği daha önceki bölümde bahsedilmişti. Bu programın ara yüzünde ise çoklu üst üste çıkıştırma yapmak için gerekli olan verilerin girilebilmesi için bazı farklılıklar söz konusudur.

Şekil 3.12’den de görülebileceği gibi, çoklu üst üste çıkıştırma içeren programın ara yüzü için “Çakışan Faaliyetler” bölümünde değişiklik yapılmıştır. Bu bölümde, birden fazla faaliyet çiftleri arasında çıkıştırmaya imkan verecek şekilde “Üst Seviye”, “Alt Seviye” ve “Çakışma Etkisi” ve “Gelişim parametresi” için çoklu alan ayrılmıştır. Bu bölümün alt kısmında yer alan alanların doldurulması ve kaydedilmesinden sonra “Yeni” kutusuna tıklanarak yeni bir faaliyet çifti için gerekli bilgiler doldurulabilmektedir. Kaydedilen bütün bilgiler “Tanımlı Aktiviteler” başlığı altında sırayla görülebilmektedir.

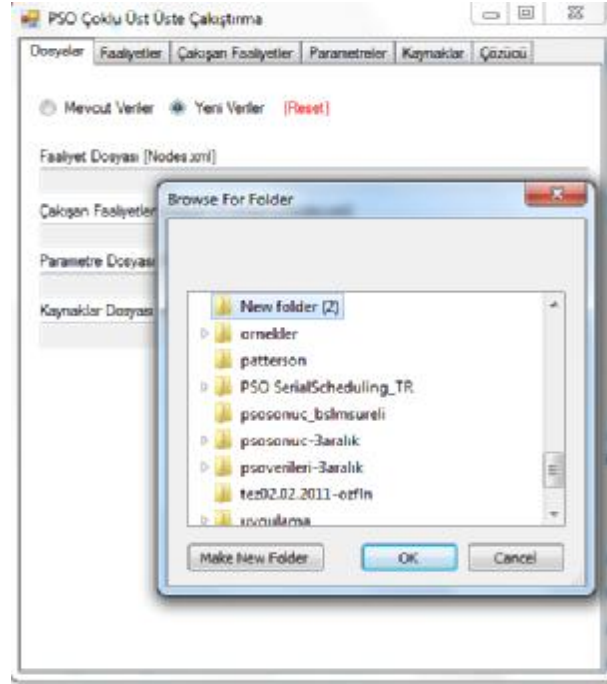


Şekil 3.12. Çoklu üst üste çakıştırma içeren programın ara yüzü-Çakışan faaliyetler bölümü-

3.2.6.2. Verilerin Programa Aktarılması ve Programın Çalıştırılması

Bir projeye ait veri dosyalarının oluşturulması ve yeni verilerin girilmesi işlemi bütün programların ara yüzlerinde aynı şekilde gerçekleştirilmektedir. Geliştirilen ara yüz kullanılarak, veri girişleri rahatlıkla yapılabilir. Bu amaçla, ilk olarak programa ait veri dosyalarının oluşturulması daha sonra ise sırası ile ilgili verilerin girilmesi gerekmektedir.

Veri dosyalarının oluşturulabilmesi için öncelikle “Dosyalar” bölümündeki “Yeni Veriler” kısmı işaretlenmelidir. Bu işlemin ardından açılan pencere kullanılarak yeni bir klasör oluşturulmalı ve klasörün ismini de verdikten sonra “OK” butonuna basarak işlem tamamlanmalıdır. Bu şekilde, açılan yeni klasör içerisinde ilgili veri dosyalarının oluşturulması sağlanmış olmaktadır (Şekil 3.13).



Şekil 3.13. Veri dosyalarının oluşturulması

Bu işlemin ardından programa ait verilerin girilmesine başlanabilir. İlk olarak proje ağında yer alan faaliyetlere ait bilgilerin girilmesi gerekmektedir. “Faaliyetler” kutusuna tıklanması sonucunda kukla faaliyete olan “1” numaralı faaliyete ilişkin bilgilerin girilmesine başlanabilir(Şekil 3.14). Bu faaliyete ait bütün bilgilerin girilmesinin ardından “kaydet” kutusuna basılarak bilgiler kaydedilmelidir. Yeni faaliyetin oluşturulması için “Yeni” butonu kullanılmakta ve oluşturulan yeni faaliyetin bilgileri de aynı şekilde girilebilmektedir.

PSO Çoklu Üst Üste Çıkartma

Dosyalar Faaliyetler Çıkartılan Faaliyetler Parametreler Kaynaklar Çözümü

Tanımlı Faaliyetler

1
2

Faaliyet Numarası
2

Kaynak İhtiyaçları
1000000

Faaliyet Süresi
5

Hemen Önceki Faaliyetler
1

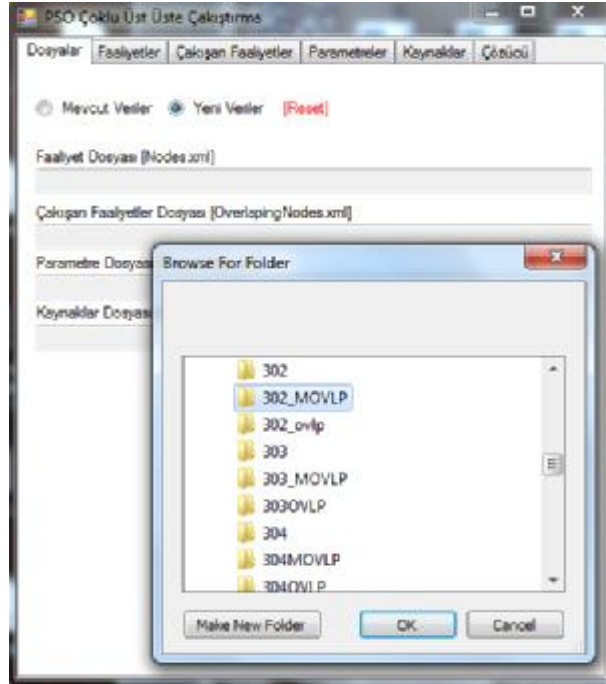
Başarıyla kaydedilmiştir.

Kaydet Yeri Sil

Şekil 3.14. Verilerin girilmesi

Benzer şekilde parametreler ve kaynaklar ile ilgili bilgilerde sırasıyla, “Parametreler” ve “Kaynaklar” kutularına tıklanarak girilmekte ve kaydedilmektedir.

Bu şekilde girilen veriler oluşturulan klasör içerisinde istenildiği zaman tekrar kullanılabilirler üzere saklanmaktadır. Bu verilerin tekrar programa aktarılabilmesi için “Dosyalar” bölümündeki “Mevcut Veriler” kısmı işaretlenmelidir. Bu işlemden sonra açılan pencereden ilgili probleme ait dosya bulunarak programa aktarılmaktadır(Şekil 3.15)



Şekil 3.15. Mevcut verilerin programa aktarılması

Mevcut verilerin programa aktarılmasının ardından istenirse, projeye ait girilen bilgilerde değişiklikler yapıp kaydedilebilir veya bazı veriler silinebilir.

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

Bu ana başlık altında öncelikle, Bölüm 3.2.2’de verilmiş olan analitik model kullanılarak çözülebilecek problem büyüklükleri ve bu problemlerin çözüm süreleri araştırılmıştır. Sonraki aşamada ise, çözüm kompleksliği olduğu bilinen mevcut problem için önerilen PSO Algoritması, literatürde yer alan test problemleri kullanılarak test edilmiş ve değerlendirilmiştir. Üçüncü bölümde, üst üste çakıştırma içermeyen temel çizelge kullanılarak, bitiş süresini kısaltabilecek üst üste çakıştırma alternatiflerinin nasıl belirlendiği açıklanmıştır. Dördüncü bölümde ise, üst çakıştırma içeren PSO algoritması, mevcut test problemleri kullanılarak değerlendirilmiştir. Son bölümde, gerçek bir elektronik yeni ürünün geliştirme projesi verileri kullanarak, üst üste çakıştırmadan elde edilebilecek süre kazancını etkileyen “alt seviye faaliyet duyarlılığı” ve üst seviye faaliyet gelişim hızı” faktörlerine dayalı senaryo analizleri yapılmıştır.

4.1. Analitik Modelin Çözüm Süresinin İncelenmesi

Bölüm 3.2.2.’de verilmiş olan matematiksel modelin bilgisayar ortamında modellenmesi için GAMS (The General Algebraic Modeling System) paket programı kullanılmıştır. Bu çalışmada kullanılan GAMS modelleme ortamı doğrusal, doğrusal olmayan ve karışık tamsayılı eniyileme problemlerinin çözümü için geliştirilmiştir. GAMS yazılımı, modellemede kullanılan yazılım dilinde standartlaşma sağlamakta ve matematiksel modelde yer alan denklemler matematiksel ifade biçimi korunarak modelleme ortamına aktarılmaktadır. GAMS modelleme ortamında program derleyicisi ve farklı çözümler bulunmaktadır. Bu çalışmada, OSL ve OSL2 çözümleri kullanılmıştır. OSL (IBM Subroutine Library) çözümleri, doğrusal programlama, karışık tamsayılı programlama problemlerinin çözümünde kullanılmaktadır. Yapılan denemeler, Intel Core 2 Duo, 2.67 GHz işlemcili ve 4 GB RAM’li bilgisayarda gerçekleştirilmiştir.

Önerilen analitik modelin çözüm etkinliğinin değerlendirilebilmesi için Tablo 3.1.’de “kaynak gücü” ve “ağ karmaşıklığı” faktörlerine göre sınıflandırılmış olan

30 faaliyetli proje ağıları kullanılmıştır. Daha önceki bölümlerler de belirtildiği gibi ele alınan problem çözüm karmaşıklığı olan bir problemidir. Burada çözüm süresini kaynak gücü ve ağ karmaşıklığı faktörleri dikkate alınarak, kesin çözüm yöntemi ile çözülebilecek problemlerin sınırları belirlenmeye çalışılmaktadır. Bu sayede geliştirilen PSO algoritmasının önemini daha net bir şekilde ortaya koymak mümkün olacaktır.

30 faaliyetli proje ağıları için dikkate alınan ve Tablo 4.19’da verilmiş olan tekli üst üste çakıştırma alternatifleri, yazılan GAMS kodları kullanılarak çözülmüştür. Üst üste çakıştırma alternatiflerinin nasıl belirlendiği Bölüm 4.3’de açıklanmaktadır. Yineleme sayısı her bir problem için 3500000 alınmıştır. Bu süre içerisinde hiçbir çözüm bulamayan problemler “çözüm bulunamadı”, olurlu çözüm bulunan problemler ise “tamsayılı çözüm” şeklinde verilmiştir. Bu durumlarda 3500000 yinelemenin tamamlanma süresi yani programın durdurulma süreleri de ayrıca belirtilmiştir. Optimum çözümün bulunduğu durumlar ise “optimum çözüm” şeklinde belirtilmiş ve optimum çözümün bulunduğu süre girilmiştir. Sonuçlar Tablo 4.1’de özetlenmiştir. Hazırlanan mevcut tabloda, problem numarası, dikkate alınan üst üste çakıştırma alternatifi, amaç fonksiyonu değeri, çözüm durumu, çözüm süresi verilmektedir.

Tablo 4.1. 30 faaliyetli proje ağlarının alternatif tekli üst üste çıkıştırma seçeneklerine göre kesin çözüm durumları

Problem Numarası	Üst Üste Çakıştırma Alternatifi	Amaç Fonk.Değeri	Çözüm Durumu	Çözüm Süresi(sn)
1	2 ve 7	-	Çözüm bulunamadı.	6180
	11 ve 27	-	Çözüm bulunamadı.	4380
2	18 ve 24	70	Tamsayılı çözüm.	5400
	24 ve 26	70	Optimum çözüm.	24,3
	26 ve 27	71	Optimum çözüm.	16,7
	27 ve 31	66	Optimum çözüm.	20,01
3	5 ve 21	47	Optimum çözüm.	20,3
	21 ve 23	48	Optimum çözüm.	10,6
	23 ve 25	42	Optimum çözüm.	10,1
	25 ve 26	45	Optimum çözüm.	10,2
	26 ve 31	46	Optimum çözüm.	7,1
4	14 ve 25	-	Çözüm bulunamadı.	6240
	25 ve 27	-	Çözüm bulunamadı.	6300
5	4 ve 8	-	Çözüm bulunamadı.	5400
	8 ve 10	40	Tamsayılı çözüm.	5460
	10 ve 18	40	Tamsayılı çözüm.	6390
	18 ve 23	40	Tamsayılı çözüm.	6180
6	3 ve 10	-	Çözüm bulunamadı.	6390
	10 ve 16	-	Çözüm bulunamadı.	6300
	16 ve 17	-	Çözüm bulunamadı.	6180
	17 ve 18	53	Optimum çözüm.	12,8
	18 ve 21	54	Optimum çözüm.	20,5
	21 ve 22	53	Optimum çözüm.	8,5
	22 ve 29	55	Tamsayılı çözüm.	6300
7	7 ve 10	-	Çözüm bulunamadı.	7200
	10 ve 15	-	Çözüm bulunamadı.	7500
	15 ve 17	-	Çözüm bulunamadı.	6600
	17 ve 20	-	Çözüm bulunamadı.	6240
	20 ve 28	-	Çözüm bulunamadı.	6300
8	2 ve 12	-	Çözüm bulunamadı.	6300
	12 ve 13	-	Çözüm bulunamadı.	6240
	13 ve 22	56	Tamsayılı çözüm.	6240
	22 ve 27	-	Çözüm bulunamadı.	5460
	27 ve 28	-	Çözüm bulunamadı.	5400
	28 ve 30	-	Çözüm bulunamadı.	5400
9	3 ve 6	65	Optimum çözüm.	12,2
	6 ve 10	61	Optimum çözüm.	11,5
	10 ve 14	65	Optimum çözüm.	11,9
	14 ve 21	65	Optimum çözüm.	9,6
	21 ve 24	66	Optimum çözüm.	12,1
	24 ve 25	61	Optimum çözüm.	19,9
	25 ve 28	62	Optimum çözüm.	20,2
	28 ve 31	59	Optimum çözüm.	13,4

Ele alınan problem, kaynak kısıtlı proje çizelgeleme problemleri içerisinde yer alması nedeniyle çözüm karmaşıklığı olan bir problemdir. Çözüm süresini etkileyen faktörler; projedeki faaliyet sayısı, kaynak kısıtlılığı derecesi, ve ağ karmaşıklığıdır.

Önerilen modelin 30 faaliyetli proje grubu için çözüm durumunu gösteren Tablo 4.1. incelendiğinde, üst üste karşılaştırma alternatiflerinin hiçbirisi için çözüm bulunamayan problemlerin 1, 4 ve 7 numaralı problemler olduğu görülmektedir. Bu problemler, farklı ağ karmaşıklığına sahip olmalarına rağmen, ortak noktaları kaynak güçlerinin düşük seviye olmasıdır. Bu bilgi neticesinde, analitik olarak çözümde en fazla zorlanılan problemlerin kaynak kısıtlılığı en yüksek olan problemler olduğu ortaya çıkmaktadır.

Optimum çözümlerin bulunabildiği problemler ise, 2, 3, 6 ve 9 numaralı kaynak gücü yüksek ve ortaya seviyede olan problemlerdir. Diğer bir önemli nokta ise, aynı proje ağına ait bazı üst üste karşılaştırma alternatifleri için optimum veya tamsayılı çözüm bulunabilirken bir kısmı için çözüm bulunamamasıdır. Bu durumda, seçilen üst üste karşılaştırma alternatifine göre de problemin analitik olarak çözülebilmek zorluğu değişebilmektedir. Örneğin 6 numaralı problem için, denen yedi üst üste karşılaştırma alternatifinin üç tanesi için çözüm bulunamamış, üç tanesi için kısa süre içerisinde optimum çözüme ulaşılmış, bir tanesinde ise mevcut yineleme sayısında ancak tamsayılı çözüm elde edilmiştir. Sonuç olarak, üst üste karşılaştırma içeren modelin çözümünün, karşılaştırma yapılan faaliyetlere bağlı olarak, analitik olarak çözümü daha da güçleştirebileceği düşünülebilir.

Tablodan da görüldüğü gibi, 30 faaliyetli proje ağları için bile bir kısım üst üste karşılaştırma alternatifleri belirlenen yineleme süresinde çözülememiştir. Bu nedenle, daha fazla faaliyet içerebilecek, farklı ağ karmaşıklığı ve kaynak gücüne sahip yeni ürün geliştirme projelerinde makul süreler içerisinde en iyi veya en iyiye yakın sonuçlar verebilecek bir sezgisel algoritma geliştirilmesi amaçlanmıştır.

4.2. PSO Algoritmasının Performansının Test Edilmesi

Bu aşamada, j30-j60-j90 ve j120 olarak adlandırılmış ve kaynak kısıtlı proje çizelgeleme problemlerinde kıyaslama problemleri olarak sıklıkla kullanılan proje verileri arasından seçilen veriler kullanılarak geliştirilen PSO performansı test edilmiştir. Kullanılan test verileri, adlarından da anlaşılacağı üzere sıra ile 30, 60, 90 ve 120 faaliyetten oluşmaktadır. Ayrıca her bir proje dört adet yenilenebilir kaynak içermektedir. Projeler, faaliyet sayısının yanı sıra, problemin çözüm süresi ve/veya problemin çözüm kalitesini etkileyebilecek, “kaynak gücü” ve “proje ağı karmaşıklığı” parametrelerine göre sınıflandırılmıştır. Tablo 3.1’de test amacıyla seçilen problemlerin bu parametrelere ait değerlerine göre sınıflandırılmış halleri verilmiştir (bkz. Bölüm 3.1). Geliştirilen algoritmanın, farklı parametre değerlerine göre sınıflandırılmış bu problemler üzerinden test edilmesi algoritmanın güvenilirliğinin daha net bir şekilde ortaya konmasını sağlamaktadır.

Elde edilen çözümlerin, çözüm kalitesi açısından değerlendirilmesi öncelikle üst üste çakıştırma içermeyen algoritma üzerinde gerçekleştirilmiştir. Bunun iki önemli nedeni vardır. İlki, üst çakıştırma uygulamasını içermeyen algoritmanın en iyi çözümü veya en iyi çözüme yakın çözümü vermesi, üst üste çakıştırma uygulanacak olan faaliyetlerin doğru bir şekilde seçilebilmesi için büyük önem taşımaktadır. Bu faaliyetlerin oluşturulan temel çizelgeye göre nasıl seçildiği Bölüm 4.3.’de verilmiştir. İkinci olarak, literatürde daha önce üst üste çakıştırma uygulamasını içeren çizelgeleme problemi ele alınmamış olduğu için çözüm kalitesi açısından kıyaslama yapılacak bir veri bulunmamaktadır. Bu nedenle, algoritmanın üst üste çakıştırma içermeyen halinin verdiği sonuçların incelenmesi mevcut algoritmanın etkinliği hakkında fikir sahibi olmamızı sağlayabilmektedir. Ayrıca, üst üste çakıştırma içermeyen projenin bitiş süresi, denen üst üste çakıştırma alternatifinin ne kadar süre kazancı sağladığının anlaşılması açısından da önemlidir. Üst üste çakıştırma içermeyen algoritma için performans ölçütü olarak aşağıdaki ölçütler dikkate alınmıştır;

- Optimum çözümü bilinen 30 faaliyetli problemler için optimum çözümden sapma miktarı,

- 60, 90, 120 faaliyetli projeler için bilinen en iyi sezgisel çözümden sapma derecesi
- En iyi değere kaçınıcı yinelemede ulaşıldığı.

Daha öncede belirtildiği gibi, üst üste çakıştırma uygulamasını içeren algoritma için kıyaslama yapabilmek amacıyla, Bölüm 3.2.2.'de verilen matematiksel modelin GAMS kodu yazılmıştır. Ancak, bu program ile de sadece bazı 30 faaliyetli problemlerin optimum çözümü elde edilebildiği için kıyaslama sınırlı kalmıştır. Üst üste çakıştırma uygulamasını içeren algoritmalarda performans ölçütü,

- Çözüme kaçınıcı yinelemede ulaşıldığı,
- GAMS programı ile çözülebilen 30 faaliyetli problemler için optimuma yakınlık derecesi alınmıştır.

Ayrıca, denen üst üste çakıştırma alternatiflerinin sağladığı süre kazançları da analiz edilmiştir. Herhangi bir alternatif ile elde edilen süre kazancı her ne kadar tek başına programın performansını değerlendirmek için yeterli olmasa da, kazançlı bir alternatifin gözden kaçırılmamış olması ve üst üste çakıştırma miktarının doğru bir şekilde belirlenerek süre kazancı sağlanmış olması algoritmanın doğru şekilde çalıştığının bir göstergesidir.

Algoritmaya ait parametre değerleri, literatürde kaynak kısıtlı proje çizelgeleme problemi için genel kabul görmüş ve iyi sonuç verdiği bilinen değerlerle aynı alınmıştır. Mevcut algortmada PSO parçacıkları 0 ve 1 arasında değişen faaliyet öncelikleri ile temsil edildiği için, hem başlangıç parçacıkları hem de güncellenmiş parçacıklar için parçacık pozisyonları $[0,1]$ aralığında sınırlandırılmalıdır. Bu durumda, $X^{\min} = 0$ ve $X^{\max} = 1$ olarak alınmıştır. Aynı zamanda, parçacık hızı (mevcut pozisyonuna bağlı olarak konum değişimi), parçacığın olurlu arama uzayının dışına çıkmasını engellemek için sınırlandırılmalıdır. Parçacık pozisyonlarının arama uzayına bağlı olarak, maksimum hızı ($V^{\max}$), $X^{\max}$ 'a eşit olacak şekilde seçilebilir. Bu durumda PSO'nun arama prosedürü boyunca, $V^{\min} = -1$ ve $V^{\max} = 1$ olarak alınmıştır (Zhang, 2006). Yapılan çalışmalar, içsel ağırlık parametresi olan w 'nın 1 alınmasının, maksimum hız

değerinin 2'den küçük olduğu ($V^{\max} < 2$) durumlar için iyi çözüm verdiğini göstermiştir (Shi ve Eberhart,1998). Bu nedenle, mevcut çalışmada da $w=1$ olarak alınmıştır. Öğrenme faktörleri olan c_1 ve c_2 katsayıları da genel kabul gördüğü gibi “2” alınmıştır (Kennedy ve Eberhart,1995). Topluluk içerisinde yer alan parçacık sayısının, projedeki kukla faaliyetler haricindeki toplam faaliyet sayısına eşit alınmasının en iyi çözümü aramada başarıyı arttıracığı yine Kennedy ve Eberhart(1998)'ın çalışmalarında gösterilmiştir. Ancak, parçacık sayısının artması ile, programın çalışma süresinin de artacağı göz önüne alınmalıdır. Yapılan çalışmada da, parçacık sayıları, ele alınan projede yer alan kukla faaliyetler haricindeki diğer faaliyetlerin toplam sayısı olarak alınmıştır.

Tablo 4.2'de 30 faaliyetli projeler için mevcut algoritmanın bulunduğu değerler ve problemlerin PSPLIB'de yayınlanan optimum çözümleri verilmiştir.

Tablo 4.2. 30 faaliyetli problemlerin üst üste çıkıştırma içermeyen PSO algoritması ile elde edilen sonuçları

Problem Numarası	PSO Sonuç	Optimum Sonuç	Geçen Süre (sn)	Yineleme Numarası
1	38	38	8	1
2	72	72	9	0
3	49	49	8	0
4	53	53	8	0
5	40	40	8	2
6	57	57	8	0
7	68	68	8	1
8	57	57	8	0
9	66	66	8	0

Geliştirilen PSO algoritması, 30 faaliyetli proje grubuna ait test problemlerinin hepsi için optimum çözümü vermiştir. Bu durumda, bir sonraki aşamada, üst üste çıkıştırma yapılacak faaliyetler, elde edilen bu optimum çizelgenin

analiz edilmesi ile doğru bir şekilde belirlenebilecektir. 30 faaliyetli proje grubuna ait problemler için çözüme kaç yinelemede ulaşıldığı yine Tablo 4.2’de verilmiştir. Kaynak gücü ve proje ağının karmaşıklığı faktörleri hangi seviyede olursa olsun algoritmanın ya başlangıç çözümle yada ilk yinelemelerde, çok kısa süre içerisinde optimum çözüme ulaştığı görülmektedir. Bütün problemler için yineleme sayısı parametresi 100 olarak alınmış ancak çözüm ekranından problemlerin genellikle 0 yada 1. yinelemede en iyi çözüme ulaşıldığı görülmüştür. Tabloda yer alan “geçen süre” sütunu, 100 yinelemenin tamamlandığı zamanı ifade etmektedir. Çözümlere genellikle ilk yinelemelerde ulaşıldığı dikkate alınırsa, çözümün bulunduğu sürenin çok daha düşük olduğu anlaşılabılır.

Şekil 4.1.’de, 1 numaralı problemin çözüm ekranı örnek olarak, verilmiştir.



Şekil 4.1. “1 numaralı problem” için çözüm ekranı

Şekil 4.1.’den görüldüğü gibi, problem için en iyi çözüme 1. yinelemede ulaşılmıştır. Aynı şekil üzerinde 100 yinelemenin 8sn gibi çok kısa bir süre içerisinde gerçekleştirildiği görülmektedir.

Tablo 4.3’de ise, 60 faaliyetli projeler için mevcut algoritmanın bulduğu proje bitiş süreleri ve PSPLIB’de yayınlanmış olan bitiş süreleri verilmiştir.

Tablo 4.3. 60 faaliyetli problemlerin üst üste çakıştırma içermeyen PSO algoritması ile elde edilen sonuçları

Problem Numarası	PSO Sonuç	PSPLIB Sonuç	Geçen Süre (sn)	Yineleme No
10	65	65	57	2
11	60	60	64	1
12	84	84	60	0
13	81	81	59	1
14	62	62	59	0
15	60	60	57	0
16	72	72	60	6
17	78	78	53	1
18	61	61	47	0

Bütün problemler için PSPLIB’de verilen sezgisel çözümlerle aynı çözüm elde edilmiştir. Mevcut verilerden bütün problemler için çözüme ilk birkaç yinelemede ve oldukça kısa süreler içerisinde ulaşıldığı görülmektedir.

90 faaliyetli proje grubuna ait problemlerin mevcut algoritma ile çözümleri ve PSPLIB’de yer alan çözümler Tablo 4.4’da verilmiştir.

Tablo 4.4. 90 faaliyetli problemlerin üst üste çıkıştırma içermeyen PSO algoritması ile elde edilen sonuçları

Problem Numarası	PSO Sonuç	PSPLIB Sonuç	Geçen Süre (sn)	Yineleme Numarası
19	96	96	199	0
20	81	81	193	0
21	93	93	155	0
22	101	101	196	0
23	98	98	174	0
24	85	85	168	0
25	83	83	198	4
26	98	98	155	2
27	97	97	136	0

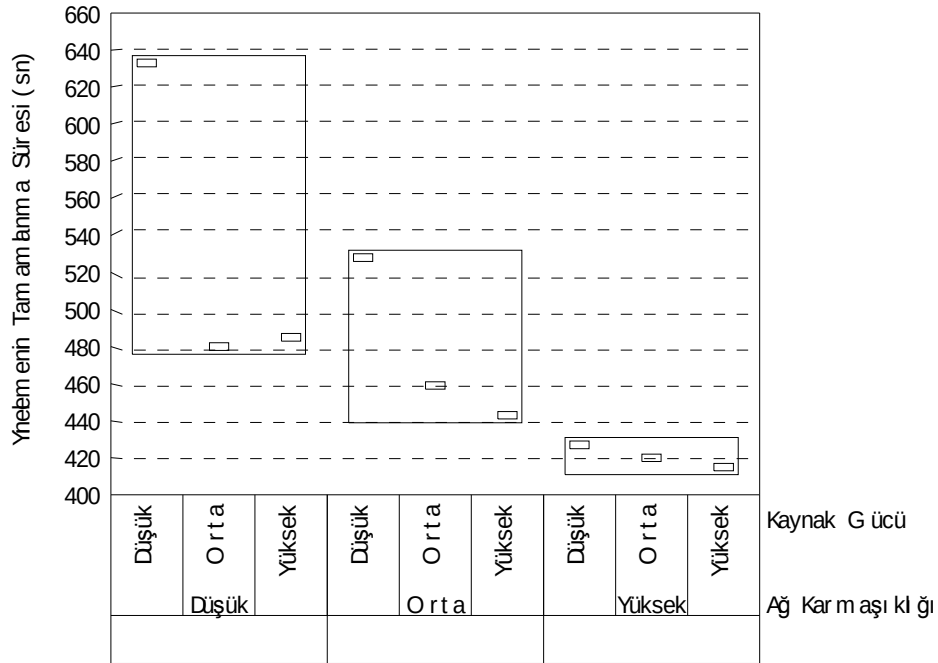
Bütün problemler için mevcut programın bulduğu sonuçlar ve PSPLIB’de yayınlanan sonuçlar aynıdır. Mevcut problemler için 100 yinelemenin bitişinin genellikle birbirine yakın olduğu görülmektedir. Aynı ağ karmaşıklığına sahip problemler için, kaynak gücü arttıkça yinelemenin tamamlanma süresinin azaldığı görülmektedir. Yine bu proje grubu içinde en iyi çözümlere en geç 4 yineleme gibi kısa süreler içerisinde ulaşılabilirdiği görülmektedir. Sonuç olarak, 90 faaliyetli proje grubu içinde yineleme sayısı çok küçük alınarak (örneğin 50) program çalıştırılabilir ve çözüme çok daha hızlı bir şekilde ulaşılabilir.

Son olarak, 120 faaliyetli proje grubuna ait problemlerin mevcut algoritma ile çözümleri ve PSPLIB’de yer alan çözümler Tablo 4.5’de verilmiştir.

Tablo 4.5. 120 faaliyetli problemlerin üst üste çakıştırma içermeyen PSO algoritması ile elde edilen sonuçları

Problem Numarası	PSO Sonuç	PSPLIB Sonuç	Geçen Süre (sn)		Yineleme Numarası	
28	71	71	633	690	85	39
29	74	74	480	502	9	12
30	92	92	485	499	0	0
31	107	107	528	530	2	2
32	93	93	459	450	14	6
33	82	82	443	444	71	35
34	105	105	427	424	39	62
35	100	100	420	402	21	29
36	91	91	415	419	18	11

Bütün problemler için PSPLIB’de yayınlanan sonuçlarla aynı sonuçlar elde edilmiştir. 120 faaliyetli proje grubunu içeren problemlerin çözülmesinin daha zor olması nedeniyle çözüm etkinliğini daha iyi gösterebilmek için iki defa çalıştırılmıştır. Her iki çalıştırmada da en iyi çözüme kaç yineleme sonucunda ulaşıldığı ve 100 yinelemenin bitiş süresi Tablo 4.5’de ilgili sütunlarda verilmiştir. 120 faaliyetli proje grubundaki problemlerin 100 yinelemeyi tamamlama süresinin diğer proje gruplarına göre daha fazla olmasının en önemli nedeni, parçacık sayısının 120 alınmasından kaynaklanmaktadır. Ancak, 120 faaliyetli projelerin faaliyet sayısı ve problemi zorlaştıran kaynak kısıtlılığı ve ağ karmaşıklığı faktörleri dikkate alındığında çözüme hızlı bir şekilde ulaşıldığı anlaşılabılır. Parçacık sayısının azaltılması ile bu süreler daha da azaltılabilir. Fakat çözüm kalitesine etkisi de dikkate alınmalıdır.



Şekil 4.2. 120 faaliyetli projeler için yinelemenin tamamlanma süreleri

Şekil 4.2.'den de görülebileceği gibi, diğer proje gruplarından daha belirgin olarak, 120 faaliyetli proje grubuna ait problemlerin 100 yinelemeyi tamamlama süreleri için bazı genellemeler yapılabilmektedir. Yinelemenin tamamlanma süresinin en uzun olduğu problem, kaynak kısıtlılığı ve ağ karmaşıklığının her ikisinin de düşük olduğu 28 numaralı problemdir. Hangi ağ karmaşıklığı seviyesinde olursa olsun, kaynak gücü arttıkça yani kaynak kısıtlılığı azaldıkça, yinelemenin tamamlanma süresinin düşüş eğilimi gösterdiği görülebilir. Aynı şekilde ağ karmaşıklığı seviyesi arttıkça yinelemenin tamamlanma süresinin düştüğü net bir şekilde görülmektedir. Genel olarak bakıldığında, hem kaynak gücü hem de ağ karmaşıklığı arttıkça sürenin azaldığı görülmektedir.

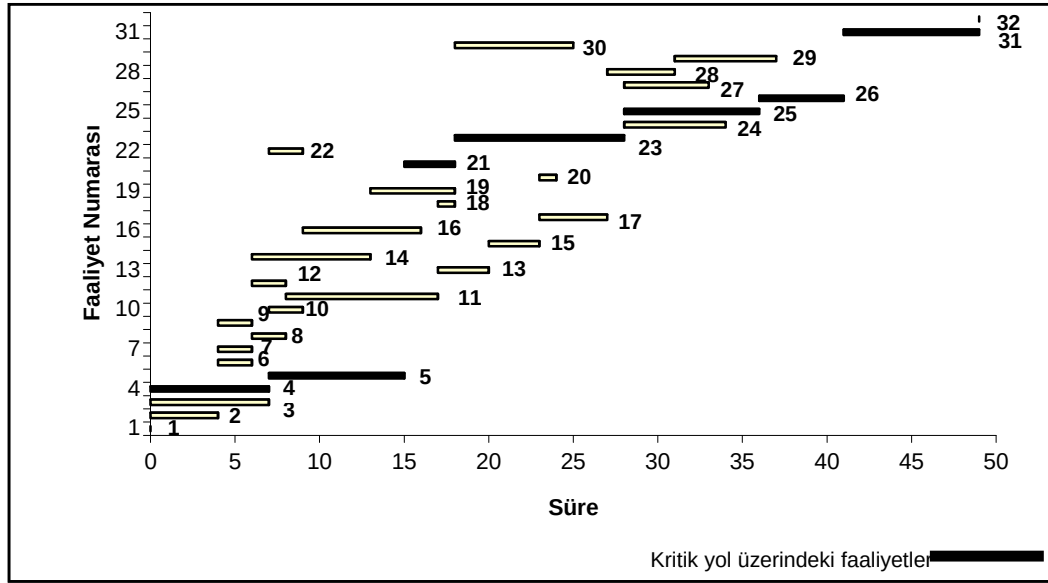
4.3. Üst Üste Çakıştırma Yapılacak Alternatif Faaliyetlerin Belirlenmesi

Yapılan çalışmada, üst üste çakıştırma yapılacak faaliyet çiftlerinin rasgele belirlenmesi yerine temel çizelge üzerindeki kritik yol analizlerine dayalı olarak belirlenmiştir. Bu yaklaşımda, proje bitiş süresi üzerinde etkisi olan faaliyetler araştırılarak, üst üste çakıştırma ile elde edilebilecek olan süre kazancı arttırılmaya çalışılmıştır.

Mevcut çalışmada da kullanılan test problemleri (j30, j60, j90, j120) öncelikle üst üste çakıştırmayı içermeyen PSO algoritması kullanılarak çalıştırılmıştır. Bu sayede hem algoritmanın performansı test edilmiş hem de çakıştırma yapılacak faaliyetlerin belirlenmesi için gerekli olan temel çizelgeler elde edilmiştir.

Elde edilen temel çizelgelerin kritik yol analizlerinin nasıl yapıldığı ve üst üste çakıştırma yapılacak faaliyetlerin nasıl belirlendiğinin anlaşılabilmesi için bu bölümde test problemleri arasından birkaç örnek sunulmaktadır. Kaynak kısıtlılığı nedeniyle kritik yol, kaynak kısıtsız proje çizelgelemede olduğu gibi düzenli bir hat şeklinde karşımıza çıkmayabilir. Bu durumları da içerecek şekilde farklı örneklerle yer vermeye çalışılmıştır.

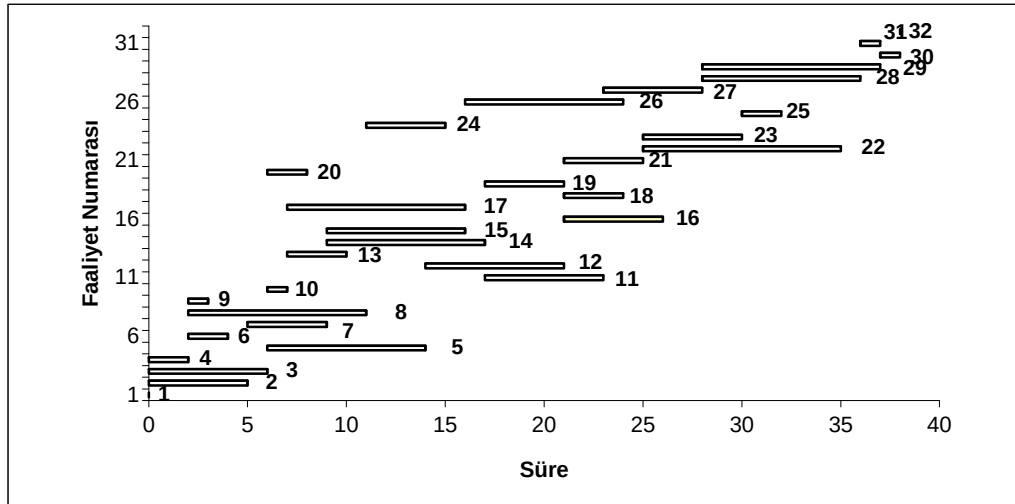
Ele alınan ilk örnek, düzenli kritik yol içeren 3 numaralı problemin GANNT şemasıdır(Şekil 4.3). Üst üste çakıştırma yapılması durumunda projenin bitiş süresini kısaltabilecek faaliyetlerin belirlenmesi için kritik yol incelemesi geriye doğru bir yöntem izlenerek gerçekleştirilmektedir. Ele alınan örnek için, Şekil 4.3'den de görüldüğü gibi, proje bitiş süresini belirleyen faaliyet 31 numaralı faaliyettir. 31 numaralı faaliyetin başlama süresi ile kendisinden hemen önce gelen 26 numaralı faaliyetin bitiş süresi aynıdır. Aynı incelemeye devam edildiğinde, 26 numaralı faaliyet ile 25 numaralı faaliyet arasında, 25 numaralı faaliyet ile 23 numaralı faaliyet arasında, 21 numaralı faaliyet ile 5 numaralı faaliyet arasında ve son olarak 5 numaralı faaliyet ile 4 numaralı faaliyet arasında da aynı şekilde başlama süresi ve bitiş süresi ilişkisi olduğu görülür. Bu analiz sonucuna göre, proje bitiş süresini öncelikli olarak etkileyebilecek faaliyetler sırası ile, 4-5-21-23-25-26-31'dir. Bu faaliyetler Şekil 4.3'de siyah renkle vurgulanmıştır.



Şekil 4.3. Düzenli kritik yol örneği- 3 numaralı problemin GANNT şeması

Mevcut kritik yolun belirlenmesinin ardından, üst üste çakıştırma yapılacak faaliyetler bu yol üzerinde yer alan faaliyetler arasından seçilebilir. Yapılan çalışmada, öncelikle belirlenen yol üzerinde yer alan faaliyet çiftleri için tekli üst üste çakıştırma alternatifleri değerlendirilmiş ve sonrasında uygun çoklu üst üste çakıştırma alternatifleri içinde problem çözülerek elde edilen süre kazancı açısından hangi alternatiflerin daha avantajlı olduğu ortaya konulmuştur. Ele alınan her bir test problemi için bu şekilde elde edilen çözümler Bölüm 4.3’de verilmiştir.

Ancak ele alınan bütün proje ağlarında bu şekilde düzenli bir yol olmayabilir. Bunun en önemli nedeni, projelerin kaynak kısıtı içermesidir. Karşılaşılabilecek özel durumlar; parçalı kritik yol, bir noktada kesişen paralel kritik yol, ve paralel kritik yol olarak isimlendirilmiştir. Bazen bu durumların bir ya da bir kaçına aynı proje ağı içinde karşılaşılabileceği de mümkün olmaktadır. Bu bölümde sırasıyla her bir duruma ait örnekler verilmiş ve bu durumlarla karşılaşılabileceği halinde üst üste çakıştırma yapılacak faaliyetlerin nasıl belirlenebileceği anlatılmıştır.

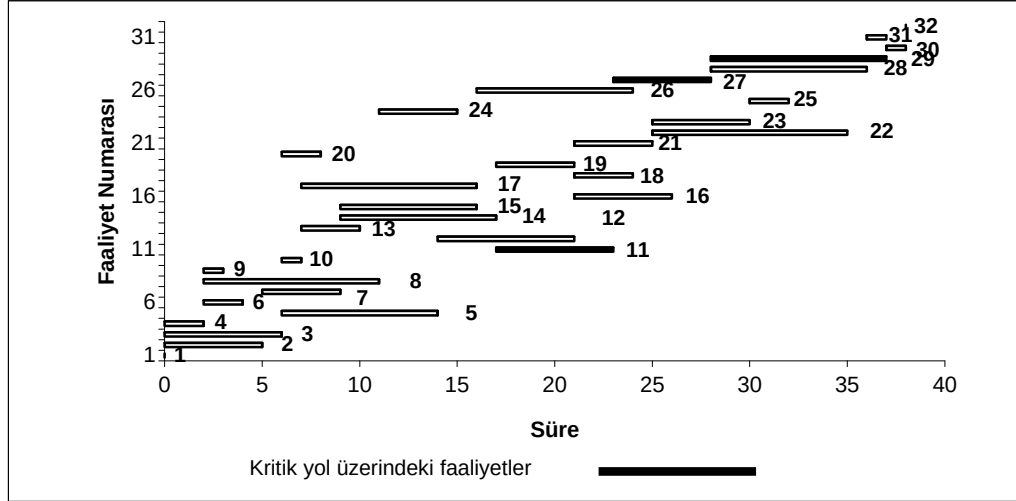


Şekil 4.4. Parçalı kritik yol içeren problemlerde üst üste çakıştırma yapılacak faaliyetlerin belirlenmesi-Adım 1-

Şekil 4.4.'de 1 numaralı probleme ait olan GANNT şeması verilmiştir. Mevcut şekil incelendiğinde, işlem gören son faaliyetin 30 numaralı faaliyet olduğu görülmektedir. 30 numaralı faaliyet ile öncelik ilişkisi olan 26, 22, ve 16 numaralı faaliyetlerin bitiş sürelerinin sırasıyla 24, 35 ve 26 olduğu görülür. Bu durumda öncelik ilişkileri açısından bakıldığında, 30 numaralı faaliyet $t=35$ 'de başlayabilecekken, $t=37$ de başlamıştır. Yani 30 numaralı faaliyet ile kendinden hemen önce gelen faaliyetler arasında kritik bir başlama bitiş süresi ilişkisi yoktur. Bu nedenle mevcut GANNT şeması örneği düzenli bir kritik yol içermemektedir.

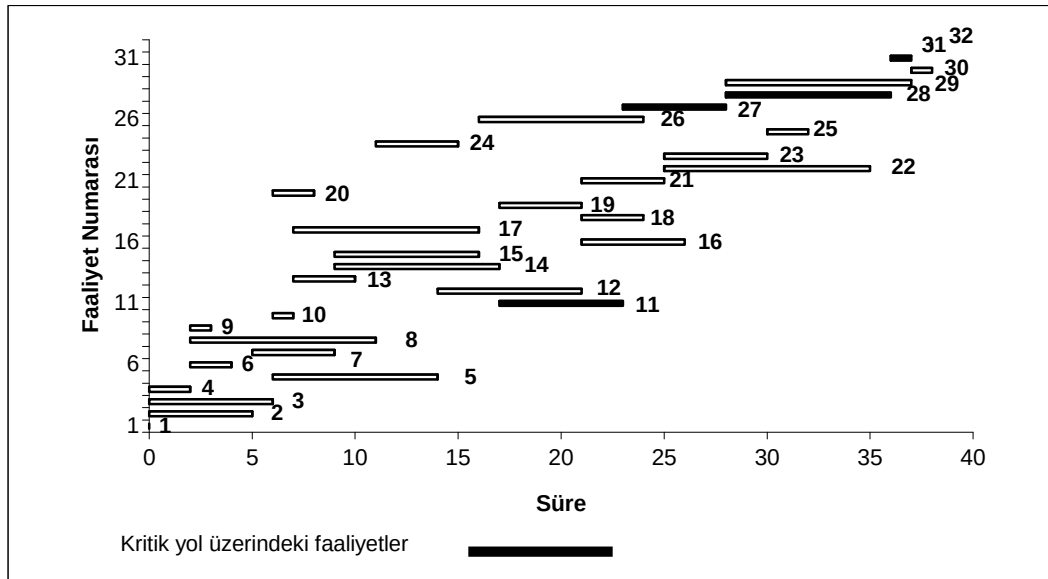
Bu şekilde bir durumla karşılaşıldığında üst üste çakıştırma yapılacak faaliyetlerin belirlenmesi açısından izlenebilecek en uygun yol, 30 numaralı faaliyetin daha erken başlamasına engel olan diğer faaliyetleri incelemektir. Bu amaçla, kukla bitiş faaliyetinden hemen önce gelen diğer iki faaliyet olan 29 ve 31 numaralı faaliyetler incelenmiştir. 29 numaralı faaliyetten hemen önce gelen faaliyetler sırası ile, 6 ve 27 numaralı faaliyetlerdir. Bu faaliyetler arasında 27 numaralı faaliyetin bitişinin hemen ardından 29 numaralı faaliyetin başladığı görülmektedir. 27 numaralı faaliyetten hemen önce gelen faaliyet ise 11 numaralı faaliyettir. Aynı şekilde 11 numaralı faaliyetin bitiş süresi ile 27 numaralı faaliyetin başlama süresinin aynı olduğu Şekil 4.4.'den görülmektedir. 11 numaralı faaliyetten hemen önce gelen 8 numaralı faaliyetin bitiş süresi ve 11 numaralı faaliyetin başlama

süresi arasında herhangi bir kritik başlama-bitiş ilişkisi olmadığı için kritik yol 11 numaralı faaliyetten sonra kesilmektedir. Bu nedenle, 11-27-29 numaralı faaliyetlerin bir kritik yol oluşturduğu söylenebilir(Şekil 4.5).



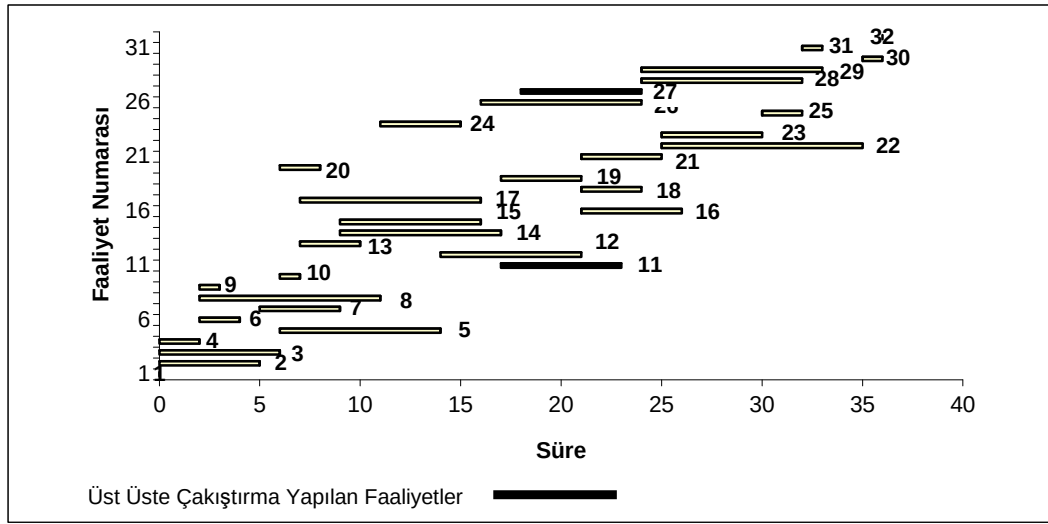
Şekil 4.5. Parçalı kritik yol içeren problemlerde üst üste çıkıştırma yapılacak faaliyetlerin belirlenmesi-adım 2-

Aynı inceleme, 31 numaralı faaliyet içinde yapıldığında Şekil 4.6.'da siyah ile vurgulanmış faaliyetler kritik yol oluşturduğu görülmektedir. Bu faaliyetler, sırasıyla 11-27-28-31 numaralı faaliyetlerdir.



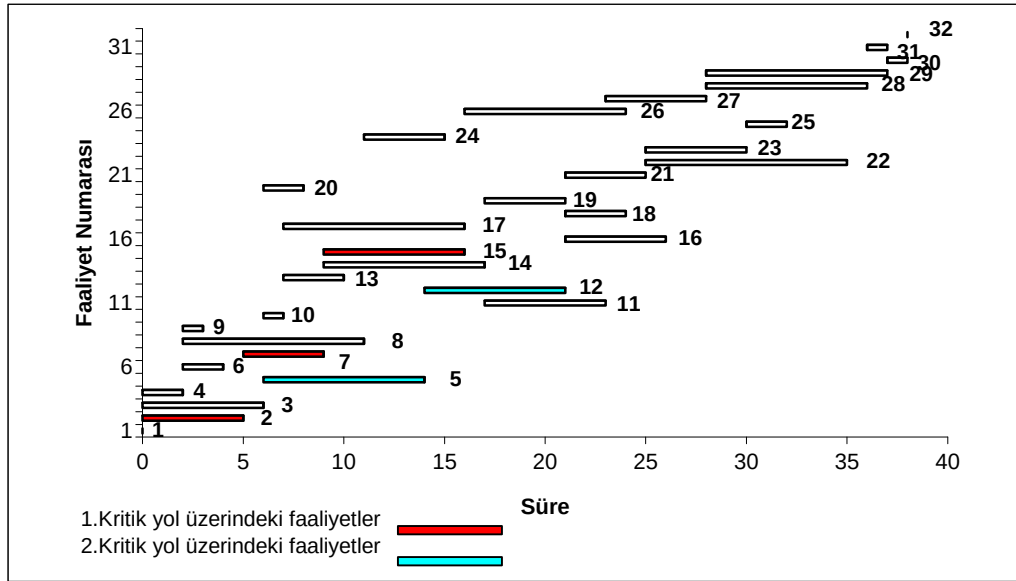
Şekil 4.6. Parçalı kritik yol içeren problemlerde üst üste çıkıştırma yapılacak faaliyetlerin belirlenmesi-adım 3-

Yapılan bu incelemelerin ardından, hem 29 hem de 31 numaralı faaliyetin öne çekilebilmesi ve dolayısıyla projenin bitiş süresini belirleyen 30 numaralı faaliyetin öne çekilebilmesi için 11 ve 27 numaralı faaliyetlerin önemli olabileceği anlaşılmaktadır. Bu nedenle, 11 ve 27 üst üste çakıştırma yapılacak faaliyet aday faaliyet çifti olarak düşünülebilir. Bu 11-27 arasında üst üste çakıştırma yapılması durumunda GANNT şeması Şekil 4.7’de, verilmiştir. Şekil 4.7’den de görüldüğü üzere, 11 ve 27 numaralı faaliyetler arasında üst üste çakıştırma yapıldığında elde edilen yeni proje bitiş süresi 36 olmuştur.



Şekil 4.7. 1 numaralı problemde 11 ve 27 numaralı faaliyetleri arasında üst üste çakıştırma içeren GANNT şeması

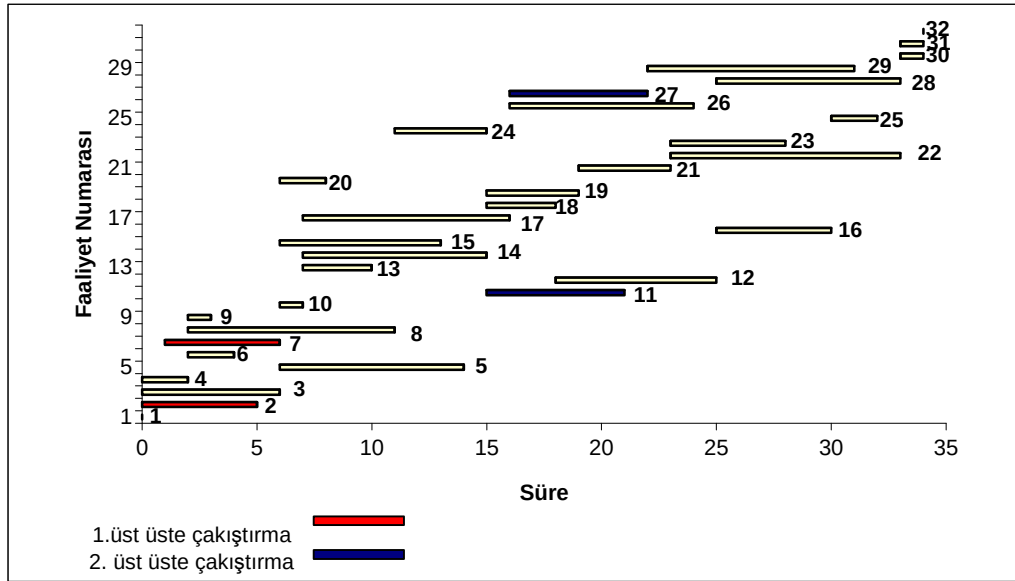
Ancak, 11 numaralı faaliyetten öncesi içinde analiz derinleştirilerek, bitiş süresinin kısaltılmasına katkı sağlayabilecek farklı aday faaliyetler bulunabilir. 8 numaralı faaliyetin bitiş süresinin 11 olmasına rağmen, 11 numaralı faaliyet 16’da başlamaktadır. Bu durumda, kaynak kısıtı nedeniyle 11 numaralı faaliyetin daha erken başlamasına engel olan faaliyetler incelenerek analize devam edilebilir.



Şekil 4.8. Parçalı kritik yol içeren problemlerde üst üste çakıştırma yapılacak faaliyetlerin belirlenmesi-adım 4-

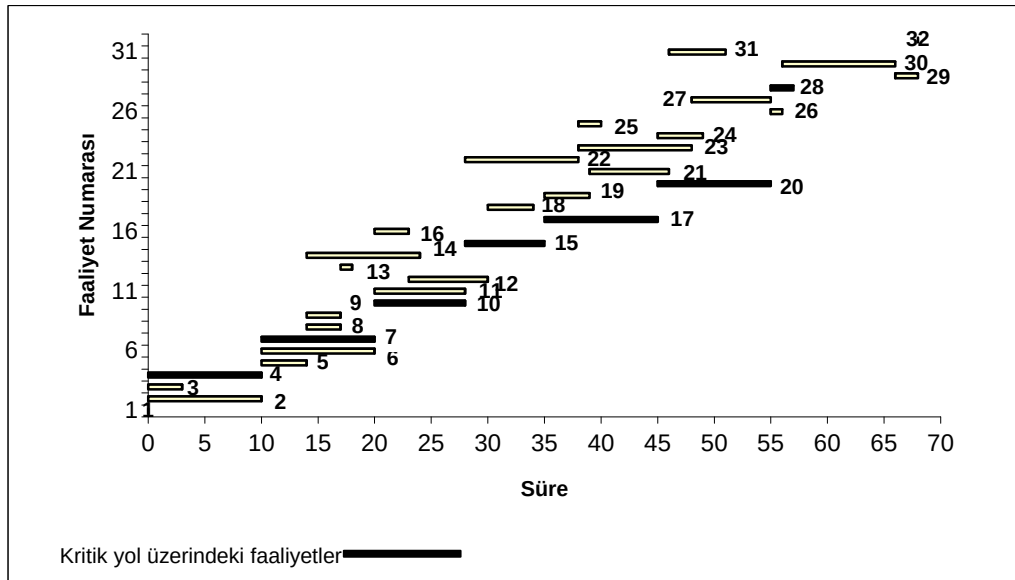
11 numaralı faaliyetin daha erken başlamasına etki edebilecek faaliyetlerin incelenmesinin ardından iki farklı kritik yol bilgisine ulaşılmıştır. Bunlar, 5-12 ve 2-7-15'dir. Bu faaliyetler, Şekil 4.8'de kırmızı ve turkuaz renkleri ile vurgulanmıştır. Bu bilginin ardından, mevcut çalışmada önerilen çoklu üst üste çakıştırma içeren PSO algoritması kullanılarak 11-27 numaralı faaliyet çiftinin yanı sıra 5-12 , 2-7 veya 7-15 alternatifleri de eklenebilir. Bu şekilde, 11 numaralı faaliyetin daha da öne çekilmesi sağlanarak proje bitiş süresinin kısaltılması sağlanabilir.

Yapılan denemeler neticesinde de, 11-27 ve 2-7 faaliyet çiftleri arasında üst üste çakıştırma içeren projenin bitiş süresinin 34 olduğu görülmüştür. Bu şekilde, sadece 11 ve 27 arasında üst üste çakıştırma yapılması durumuna göre 2 zaman birimlik daha iyi bir sonuç elde edilmesi sağladığı görülmüştür. 11-27 ve 2-7 faaliyet çiftleri arasında üst üste çakıştırma yapılması durumunda GANNT şeması ise Şekil 4.9.'da verilmiştir.



Şekil 4.9. Problem 1 için 11-27 ve 2-7 numaralı Faaliyetler Arasında Üst Üste Çakıştırmayı içeren GANNT şeması

Şekil 4.9.'dan da görüldüğü gibi, 2 ve 7 numaralı faaliyetler arasında üst üste çakıştırma yapılması 11 numaralı faaliyetin daha erken başlamasını sağlayarak proje bitiş süresinin 34'e kadar kısaltmasını sağlamıştır.

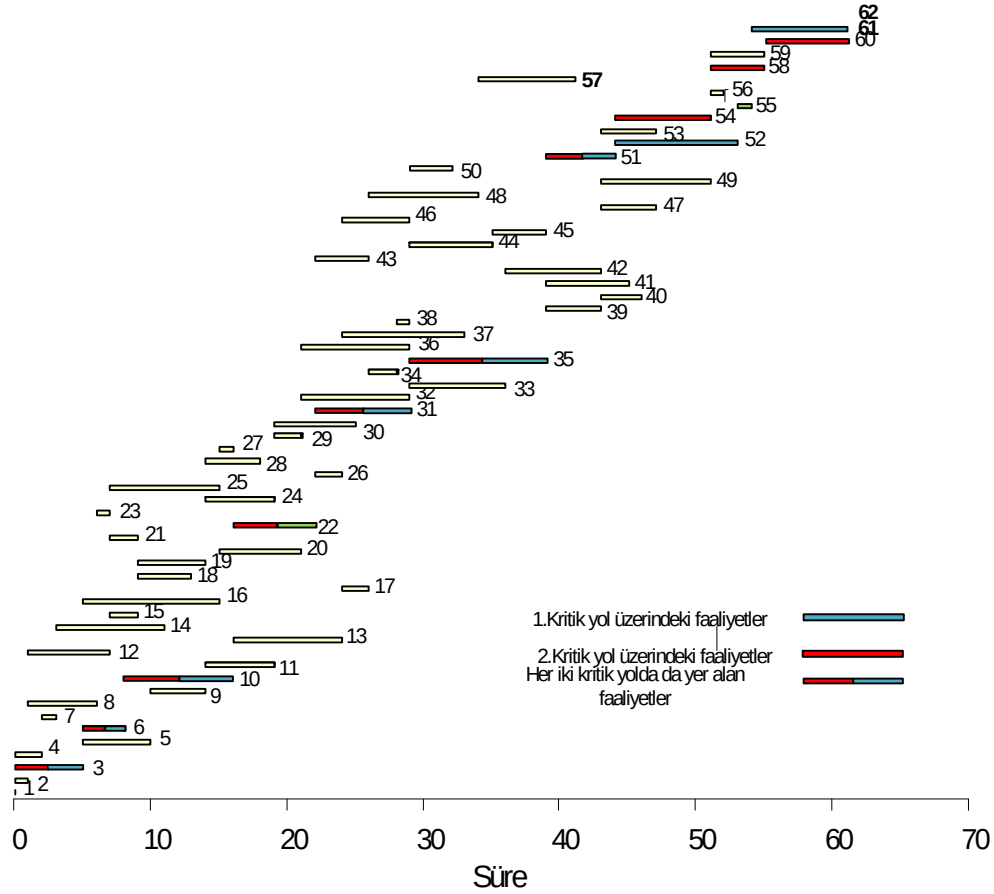


Şekil 4.10. Parçalı kritik yol için 2. örnek-8 numaralı problemin GANNT şeması-

Şekil 4.10’da verilen, 8 numaralı faaliyete ait GANNT şeması da başka bir parçalı kritik yol örneğidir. Burada da kritik yol analizin başında kesilmekle beraber sonrasında analizin derinleştirilmesi ile düzenli bir yol ortaya çıktığı görülmektedir. Şekilden görüldüğü gibi, proje bitiş süresini belirleyen faaliyet 29 numaralı faaliyetdir ancak 29 numaralı faaliyetten hemen önce gelen 23, 24 ve 28 numaralı faaliyetlerin bitmesi ile 29 numaralı faaliyetin başlama süresi arasında boşluk bulunmaktadır. Bu nedenle 29 numaralı faaliyetten herhangi bir kritik yol oluşmamaktadır. Eğer, analiz bu aşamada kesilirse, ele alınan problem için üst üst çakıştırmada kazanç sağlayabilecek alternatif herhangi bir faaliyet çiftine ulaşamamış olur. Ancak, analize devam edilirse, Şekil 4.10’da siyah renkle belirtilmiş olan faaliyetlerin 28-20-17-15-10-7-4 şeklinde bir kritik yol oluşturduğu görülür. Bu yol üzerindeki faaliyetler arasında alternatif üst üste çakıştırma çiftleri dikkate alındığında, kaynak kısıtı nedeniyle erken başlayamayan 29 numaralı faaliyetin daha öne çekilmesi mümkün olabilir. Nitekim, yapılan denemeler sonucunda, yalnızca 4 ve 7 faaliyetleri arasında üst üste çakıştırma yapılması ile proje bitiş süresinin 60’a düştüğü, 4 ve 7 faaliyet çiftleri ile 17 ve 20 faaliyet çiftleri arasındaki çakıştırma birlikte yapılırsa proje bitiş süresinin 57’ye kadar düştüğü görülmüştür.

Bu örnekten anlaşılacağı üzere, kaynak kısıtlarının olması nedeniyle, her zaman düzenli bir kritik yol ortaya çıkmamaktadır. Bu durumda izlenmesi gereken yol, kaynak kısıtına neden olan diğer faaliyetlerin yer aldığı kritik yolları belirleyerek bunlar arasındaki üst üste çakıştırma alternatifinin değerlendirilmesidir. Bu sayede, proje bitişini etkileyen asıl faaliyetlerin öne çekilmesi sağlanabilir.

Karşılaşılabilecek diğer bir özel durum ise, paralel kritik yol oluşması olarak adlandırılmıştır. Bu durumda, proje bitişini belirleyen iki veya üç farklı faaliyet ve bu faaliyetleri takip eden farklı yollar bulunmaktadır. Bazen bu farklı kritik yollar bir noktadan sonra tek bir yol üzerinde kesişebilir. Bu duruma örnek, Şekil 4.11.’de verilmiştir.

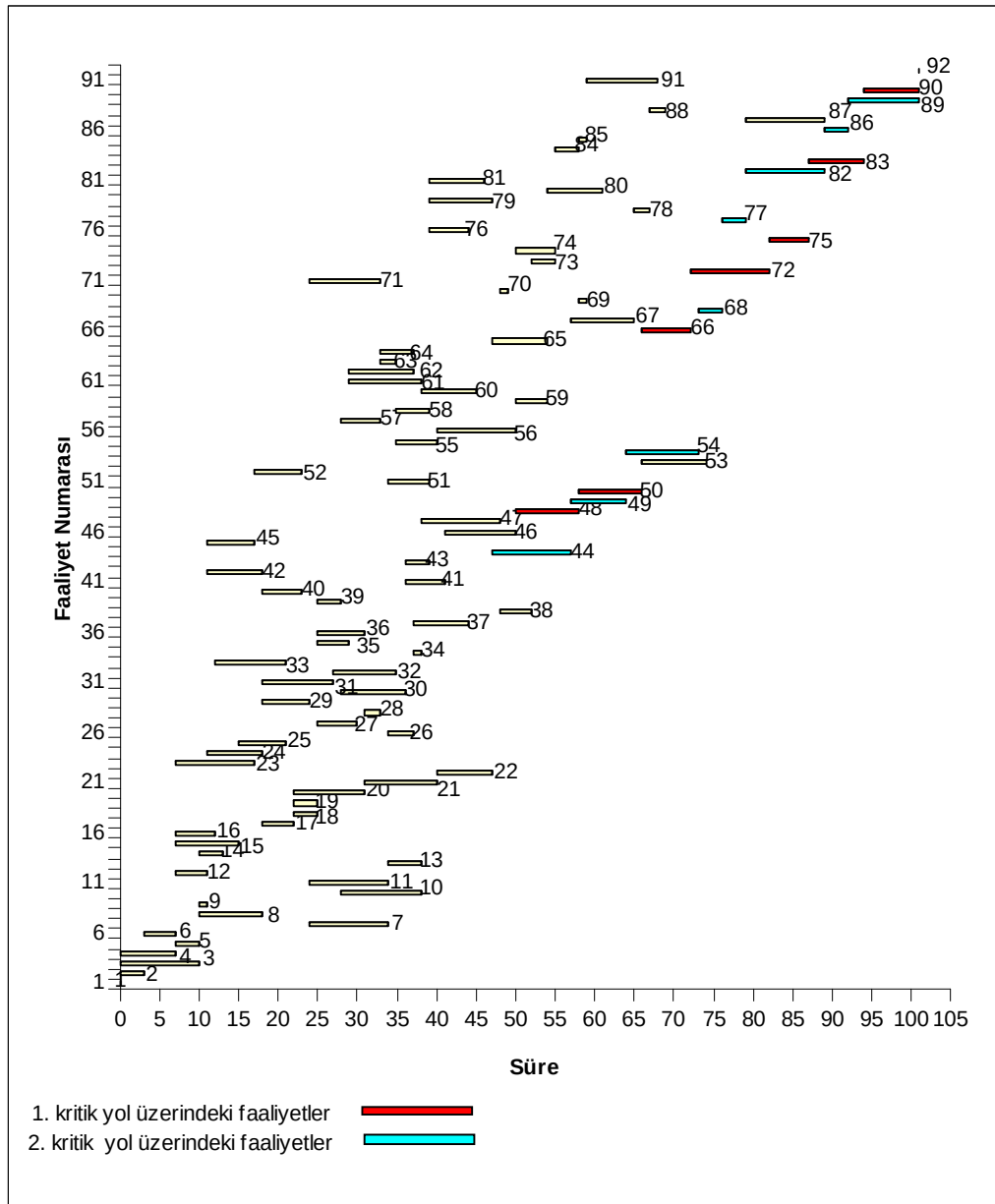


Şekil 4.11. Bir noktada kesişen paralel kritik yol örneği –18 numaralı problemin GANNT şeması

Şekil 4.11’de proje bitişini belirleyen faaliyetler 60 ve 61 numaralı faaliyetler olarak karşımıza çıkmaktadır. Bu faaliyetler öncelik ilişkilerine göre takip edildiğinde, 60 numaralı faaliyetin yer aldığı kritik yok mavi renkle, 61 numaralı faaliyetin yer aldığı kritik yol ise kırmızı renkle gösterilmiştir. Her iki yolunda 51 numaralı faaliyetten sonra tek bir kritik yolda kesiştiği görülmektedir. Bunu belirtebilmek için 51 numaralı faaliyetten itibaren kritik yol üzerindeki faaliyetler her iki renkle birlikte gösterilmiştir. Böyle bir durumla karşılaşıldığında, tek bir faaliyet çifti arasında üst üste çakıştırma yapılacaksa, her iki kritik yolun çakıştığı faaliyetler üst üste çakıştırma yapılacak alternatif faaliyetler olarak düşünülmelidir. Bu sayede hem 60 hem de 61 numaralı faaliyetin öne çekilmesi sağlanarak proje bitiş süresi kısaltılmış olur. Aksi takdirde, yalnızca 52-55 veya 54-58 arasında üst üste

çakıştırma yapılması bir kazanç sağlamayabilir. Yapılan denemelerde de, 54-58, 58-60 veya 52-55, 55-61 faaliyetleri arasındaki tekli üst üste çakıştırmalar ayrı ayrı denendiğinde, bu proje bitiş süresinin değıştirmediğı görölmüşür. Ancak, sadece 10-22 faaliyetleri arasındaki üst üste çakıştırmanın proje bitiş süresini 57 olarak değıştirdiğı görölmüşür.

Paralel kritik yol içeren diğeri bir örnek 22 numaralı problemin temel çizelgesinde karşımıza çıkmaktadır(Şekil 4.12).



Şekil 4.12. Paralel kritik yol örneğı –22 numaralı problemin GANNT Şeması

Bu örneğin, Şekil 4.11’de verilen örnekten farkı, paralel kritik noktaların herhangi bir faaliyetten sonra kesişmemesidir. Bu şekilde bir durumla karşılaşıldığında, dikkate alınması gereken temel nokta hangi kritik yolda yer alırsa alsın tekli üst üste çakıştırmanın proje süresinin kısaltılması açısından herhangi bir fayda sağlayamayacağıdır. Bu nedenle bunun gibi bir örnekte proje süresini kısaltabilmek için, çoklu üst üste çakıştırmaya olanak veren PSO algoritması kullanılarak, her iki yoldan da alternatif faaliyetleri içeren çoklu üst üste çakıştırma uygulamalarının denenmesidir.

4.4. Üst Üste Çakıştırma İçeren PSO Algoritmasının Değerlendirilmesi

Üst üste çakıştırma içeren PSO algoritmasının, farklı kaynak kısıtlılığı, faaliyet sayısı, ve proje ağı karmaşıklığı içeren problemlerin çözümündeki etkinliğinin incelenmesi açısından Bölüm 3.1’de verilen test problemleri kullanılmıştır.

Bu bölümde, programın çalıştırılması için gerekli olan çakışma etkisi parametresi, düşük alt seviye faaliyet duyarlılığı varsayımı ile alt seviye faaliyetin süresinin %10’u kadar alınmıştır. Gelişim fonksiyonu parametresi ise analitik modeldeki kabule uygun olarak 1 alınmıştır. Diğer parametreler ise, Bölüm 4.2.’de açıklandığı gibi alınmıştır.

Her bir test problemi için, öncelikle tekli üst üste çakıştırmadan elde edilen sonuçlar verilmiş ve sonrasında her bir örnek için denenene çoklu üst üste çakıştırma sonuçları sunulmuştur.

Ele alınan 30 faaliyetli test problemleri için denenene üst üste çakıştırma alternatifleri, elde edilen sonuçlar ve en iyi sonuca kaçınıcı yinelemede ulaşıldığına dair sonuçlar Tablo 4.6’da verilmiştir. Bunlara ek olarak mevcut tabloda, GAMS kodları kullanılarak optimum çözümleri bulunabilen problemlerin optimum sonuçlarına da yer verilmiştir. Bu şekilde sınırlı sayıda problemler içinde olsa, algoritmanın sonuçlarının optimuma yakınlık derecesi görülebilmektedir. Üst üste çakıştırma içermeyen problemlerde olduğu gibi mevcut tabloda 100 yinelemenin

bitiř süresine yer verilmemesinin nedeni bitiř süresinin üst üste çakıřtırma içermeyen durumla hemen aynı olmasıdır.

Tablo 4.6. 30 faaliyetli test problemleri için tekli üst üste çıkıştırma sonuçları

Problem Numarası	Üst Üste Çıkıştırma Alternatifi	PSO Sonuç	Optimum Sonuç	Yineleme Sayısı
1	2 ve 7	36	-	2
	11 ve 27	36	-	2
2	18 ve 24	69	-	1
	24 ve 26	70	70	0
	26 ve 27	71	71	0
	27 ve 31	66	66	0
3	5 ve 21	47	47	0
	21 ve 23	48	48	0
	23 ve 25	42	42	0
	25 ve 26	45	45	0
	26 ve 31	46	46	0
4	14 ve 25	50	-	2
	25 ve 27	52	-	0
5	4 ve 8	37	-	1
	8 ve 10	38	-	0
	10 ve 18	38	-	8
	18 ve 23	39	-	0
	9 ve 20	40	-	0
6	3 ve 10	56	-	0
	10 ve 16	56	-	0
	16 ve 17	57	-	0
	17 ve 18	53	53	0
	18 ve 21	54	54	0
	21 ve 22	53	53	0
	22 ve 29	54	-	0
7	7 ve 10	65	-	2
	10 ve 15	65	-	1
	15 ve 17	65	-	4
	17 ve 20	65	-	0
	20 ve 28	68	-	0
8	2 ve 12	56	-	0
	12 ve 13	56	-	8
	13 ve 22	56	-	0
	22 ve 27	56	-	0
	27 ve 28	54	-	1
	28 ve 30	51	-	0
9	3 ve 6	65	65	2
	6 ve 10	61	61	0
	10 ve 14	65	65	0
	14 ve 21	65	65	0
	21 ve 24	66	66	1
	24 ve 25	61	61	3
	25 ve 28	62	62	0
	28 ve 31	59	59	0

Üst üste çakıştırma içermeyen problemlerde olduğu gibi, bu problemlerin çözümlerine de ilk yinelemelerde ulaşıldığı görülmektedir. Mevcut algoritmanın çözümleri, GAMS kodları ile çözüm veren problemlerin optimum çözümleri ile kıyaslandığında, mevcut algoritmanın bu problemler için optimum sonuca ulaştığı görülmektedir. Aynı zamanda, geliştirilen algoritma, analitik olarak çözilemeyen diğer bütün problemler içinde oldukça kısa süreler içerisinde çözüm vermiştir. Tablodan görüldüğü gibi, tekli üste çakıştırma ile elde edilen en iyi çözümler her bir problem için sırası ile, 36, 66, 42, 50, 37, 53, 65, 51 ve 59'dur. Bu değerler tablo üzerinde koyu renk ile belirtilmiştir.

Buna ek olarak, çoklu üst üste çakıştırmaya imkan veren PSO programı kullanılarak, farklı faaliyet çiftleri arasındaki üst üste çakıştırma alternatiflerini birlikte uygulayarak proje bitiş süresinin daha da kısaltılması mümkündür. Bu amaçla her bir problem için denenmiş olan çoklu üst üste çakıştırma alternatifleri ve bunların sonuçları Tablo 4.7'de verilmiştir.

Tablo 4.7. 30 Faaliyetli projeler için çoklu üst üste çakıştırma sonuçları

Problem Numarası	Çoklu Üst Üste Çakıştırma Alternatifleri	Sonuç	Yineleme Sayısı
1	2 ve 7; 11 ve 27	34	1
2	18 ve 24; 26 ve 27	69	0
	18 ve 24; 27 ve 31	63	0
	24 ve 26; 27 ve 31	64	0
	5 ve 7; 11 ve 17; 27 ve 31	62	0
	5 ve 7; 11 ve 17; 24 ve 26; 27 ve 31	60	0
	5 ve 7; 11 ve 17; 18 ve 24; 27 ve 31	59	0
3	5 ve 21; 23 ve 25	40	0
	5 ve 21; 26 ve 31	44	0
	23 ve 25; 26 ve 31	42	0
	5 ve 21; 23 ve 25; 26 ve 31	40	0
4	5 ve 16; 14 ve 25	49	3
5	4 ve 8; 10 ve 18	36	1
	4 ve 8; 18 ve 23	36	4
	4 ve 8; 10 ve 18; 9 ve 20	35	0
	4 ve 8; 18 ve 23; 9 ve 20	34	1
6	17 ve 18; 21 ve 22	49	0
	10 ve 16; 17 ve 18; 21 ve 22	48	0
7	4 ve 7; 10 ve 15	59	10
	4 ve 7; 10 ve 15; 17 ve 20	57	1
8	2 ve 12; 27 ve 28	52	1
	28 ve 30; 22 ve 27; 12 ve 13	51	0
9	28 ve 31; 24 ve 25	54	2
	28 ve 31; 24 ve 25; 14 ve 21	53	2
	28 ve 31; 24 ve 25; 14 ve 21; 6 ve 10	49	3

Tablo 4.7’den, çoklu üst üste çakıştırma içeren problemlerin çözümlerine de diğer problemlerde olduğu ilk yinelemelerde ulaşıldığı görülmektedir. Bu tablolardaki sonuçlar yardımıyla üst üste çakıştırma ile ilgili bazı özel durumlarla karşılamak mümkündür. İlk olarak, bazı faaliyetler arasındaki tekli üst üste çakıştırma tek başına sürenin kısaltılması açısından bir avantaj sağlayamazken, diğer bazı faaliyetlerle birlikte düşünüldüğünde bitiş süresinin kısaltılmasında olumlu katkılar yaptığı görülmektedir. Örneğin, 5 numaralı problem için denen 9 ve 20 numaralı faaliyetler arasındaki üst üste çakıştırma alternatifi proje bitiş süresini değiştirmezken aynı faaliyetler arasındaki üst üste çakıştırma, 4 ve 8 ile 18 ve 23 faaliyetleri arasındaki üst üste çakıştırma ile birlikte denendiğinde proje bitiş süresinin 34’e kadar düşürülebildiği görülmektedir. Bu nedenle, bir faaliyet çifti

arasındaki üst üste çakıştırma tek başına yeterli olmuyorsa, bu çakıştırmayı tamamen göz ardı etmek bazı durumlarda yanlış olabilir. Diğer bir duruma ise, 8 numaralı örnekte rastlanmıştır. Bu örnekte, 28 ve 30 numaralı faaliyetlerin tekli üst üste çakıştırma sonucunda proje bitiş süresi 51'e kadar düşürülmüştür. Ancak, yapılan çoklu üst üste çakıştırma denemelerinde 28 ve 30'a ek olarak kritik yolda bulunan diğer faaliyetlerin eklenmesi ile proje bitiş süresinin daha fazla düşürülemediği görülmüştür.

Üste çakıştırma ile elde edilen kazancın özetlenmesi amacıyla, her bir problem için tekli üst üste çakıştırma ile elde edilen en iyi sonuçlar, çoklu üst üste çakıştırma ile elde edilen en iyi sonuçlar ve çakıştırma olmaması durumundaki sonuçlar Tablo 4.8'de özetlenmiştir.

Tablo 4.8. 30 faaliyetli projeler için üst üste çakıştırma ile elde edilen süre kazançları

Problem Numarası	Üst Üste Çakıştırma İçermeyen Sonuç	Tekli Üst Üste Çakıştırma İle Elde Edilen En İyi Sonuç	Kazanç (%)	Çoklu Üst Üste Çakıştırma ile Elde Edilen En İyi Sonuç	Kazanç (%)
1	38	36	5,26	34	10,53
2	72	66	8,33	59	18,06
3	49	42	14,29	40	18,37
4	53	50	5,66	49	7,55
5	40	37	7,50	34	15
6	57	53	7,02	48	15,79
7	68	65	4,41	57	16,18
8	57	51	10,53	51	10,53
9	66	59	10,61	49	25,76

Tablo 4.8'den görüldüğü gibi, 8 numaralı örnek haricinde diğer bütün problemler için çoklu üst üste çakıştırma alternatiflerinin denenmesi ile projelerin bitiş süresinde daha fazla kısalma sağlanmıştır. 1 numaralı problem için denenen tekli üst üste çakıştırma alternatifleri ile bitiş süresi en fazla 36'ya kadar düşürülerek proje bitiş süresinde %5.26'lık bir kısalma sağlanmışken; çoklu üst üste çakıştırma alternatiflerinin denenmesi ile sürenin 34'e kadar düşürüldüğü görülmüştür. Bu durumda, üst üste olmayan duruma göre proje süresinde % 10,53'lük bir kısalma

sağlanmıştır. Özellikle, 2, 5, 6, 7, ve 9 numaralı problemlerde çoklu üst üste çakıştırma ile proje bitiş sürelerinde oldukça iyi oranlarda düşüş sağlamıştır. Pratikte uygulanabilecek bir durum olan çoklu üst üste çakıştırmanın algoritmaya dahil edilmesi ile, tekli üst üste çakıştırmaya göre daha fazla süre kazancı sağlayabilecek bir çok farklı çoklu üst üste çakıştırma alternatifleri ortaya konulabilmektedir.

60 faaliyetli problemlerin tekli üst üste çakıştırma sonuçları, Tablo 4.9’da verilmiştir.

Tablo 4.9. 60 faaliyetli test problemlerinin tekli üst üste çakıştırma sonuçları

Problem Numarası	Üst üste Çakıştırma	Sonuç	Yineleme Sayısı	Problem Numarası	Üst üste Çakıştırma	Sonuç	Yineleme Sayısı
10	3 ve 7	65	0	14	3 ve 12	59	1
	7 ve 11	59	0		12 ve 15	59	3
	11 ve 14	59	0		15 ve 24	60	2
	14 ve 44	57	9		24 ve 54	60	5
	44 ve 52	59	0		54 ve 57	60	0
	52 ve 53	64	0		57 ve 58	59	0
	53 ve 54	65	0		58 ve 59	60	0
	54 ve 55	60	0	15	3 ve 9	57	0
	55 ve 59	62	0		9 ve 21	56	0
11	2 ve 7	59	1		21 ve 23	57	0
	7 ve 11	59	3		23 ve 34	56	0
	11 ve 12	59	1		34 ve 37	56	0
	12 ve 48	60	3		37 ve 53	59	0
	48 ve 51	58	1		53 ve 55	59	0
	51 ve 52	58	0		55 ve 60	59	0
	52 ve 53	58	0	16	49 ve 51	70	2
	53 ve 58	58	3		51 ve 56	71	9
	58 ve 61	58	0		56 ve 61	71	3
12	3 ve 6	80	0	17	4 ve 7	75	1
	6 ve 9	79	0		7 ve 18	76	0
	9 ve 29	79	0		18 ve 19	76	2
	29 ve 30	80	0		19 ve 27	76	8
	30 ve 35	82	0		27 ve 31	76	0
	35 ve 41	83	0		31 ve 35	76	9
	41 ve 45	80	0		35 ve 45	76	3
	45 ve 54	77	0		45 ve 51	77	1
	54 ve 55	78	0		51 ve 53	74	10
13	55 ve 59	77	0		53 ve 58	75	1
	6 ve 11	78	5		58 ve 61	76	21
	11 ve 13	78	1	18	3 ve 6	59	0
	13 ve 22	76	1		6 ve 10	60	0
	22 ve 30	77	6		10 ve 22	57	0
	30 ve 32	81	1		22 ve 31	58	0
	32 ve 51	81	0		31 ve 35	60	0
	56 ve 58	80	0		35 ve 51	60	0
	58 ve 60	81	0				

60 faaliyetli proje grubu için yapılan tekli üst üste çakıştırma denemelerinin sonuçlarına da ilk yinelemelerde ulaşılmaktadır. Yapılan denemeler sonucunda, her bir problem için proje bitiş süresini en fazla düşüren alternatif ve bu alternatifin proje bitiş süresi Tablo 4.9’da koyu olarak vurgulanmıştır.

Proje süresinde daha fazla kısaltmalar sağlayabilecek çoklu üst üste çakıştırma alternatiflerinin sonuçları ise Tablo 4.10'da verilmiştir.

Tablo 4.10. 60 faaliyetli test problemlerinin çoklu üst üste çakıştırma sonuçları

Problem Numarası	Çoklu Üst Üste Çakıştırma Alternatifleri	Sonuç	Yineleme Sayısı
10	11 ve 14; 44 ve 52	53	25
	7 ve 11; 44 ve 52; 54 ve 55	53	1
	14 ve 44; 54 ve 55	52	56
11	2 ve 7; 48 ve 51; 58 ve 61	55	8
	2 ve 7; 48 ve 51; 52 ve 53	55	0
	7 ve 11; 51 ve 52; 58 ve 61	54	7
	2 ve 7; 51 ve 52; 58 ve 61	53	21
12	55 ve 59; 41 ve 45	73	0
	9 ve 29; 55 ve 59	72	0
	45 ve 54; 55 ve 59	70	0
	3 ve 6; 9 ve 29; 55 ve 59	68	0
	3 ve 6; 9 ve 29; 45 ve 54; 55 ve 59	66	0
13	6 ve 11; 13 ve 22	76	11
	13 ve 22; 56 ve 58	75	1
14	3 ve 12; 57 ve 58	59	0
	12 ve 15; 57 ve 58	59	0
	3 ve 12; 15 ve 24; 57 ve 58	59	0
15	3 ve 9; 21 ve 23	56	0
	3 ve 9; 21 ve 23; 55 ve 60	56	0
16	49 ve 51; 56 ve 61	70	30
	49 ve 51; 56 ve 61; 41 ve 46; 37 ve 39	70	4
17	4 ve 7; 51 ve 53	72	11
	4 ve 7; 31 ve 35; 51 ve 53	71	35
	4 ve 7; 31 ve 35; 53 ve 58	71	5
	4 ve 7; 31 ve 35; 19 ve 27; 53 ve 58	70	2
18	10 ve 22; 51 ve 52; 58 ve 60	56	0
	6 ve 10; 22 ve 31; 35 ve 51	57	0

Tablo 4.9 ve Tablo 4.10'daki sonuçlar birlikte incelenecek olursa, 60 faaliyetli proje grupları için 14 ve 15 numaralı problemler dışında bütün problemler için çoklu üst üste çakıştırma denemeleri neticesinde proje süresinde daha fazla

kısalma sağlanmıştır. Tekli ve çoklu üst üste çakıştırmanın, üst üste çakıştırma olmaması durumunda göre proje bitiş süresi üzerindeki etkisinin özetlenmesi amacıyla Tablo 4.11 oluşturulmuştur.

Tablo 4.11. 60 faaliyetli projeler için üst üste çakıştırma ile elde edilen süre kazançları

Problem Numarası	Üst Üste Çakıştırma İçermeyen Sonuç	Tekli Üst Üste Çakıştırma ile Elde Edilen En İyi Sonuç	Kazanç (%)	Çoklu Üst Üste Çakıştırma ile Elde Edilen En İyi Sonuç	Kazanç (%)
10	65	57	12,31	52	20
11	60	58	3,33	53	11,67
12	84	77	8,33	66	21,43
13	81	76	6,17	75	7,41
14	62	59	4,84	59	4,84
15	60	56	6,67	56	6,67
16	72	70	2,78	70	2,78
17	78	74	5,13	70	10,26
18	61	57	6,56	56	8,20

14, 15 ve 16 numaralı problemlerin denen çoklu üst üste çakıştırma alternatifleri ile, mevcut problemlerin tekli üst üste çakıştırma alternatifinden elde edilen süre kazancından daha iyi bir kazanç elde edilememiştir. Bunun dışında, diğer bütün projeler için, çoklu üst üste çakıştırma ile bitiş süresinde farklı oranlarda daha fazla azalma sağlanabilmiştir. Özellikle, çoklu üst üste çakıştırma ile, 10, 11, 12, ve 17 numaralı projelerin bitiş sürelerinde oldukça önemli kısaltmalar sağlandığı Tablo 4.14'den görülmektedir.

90 faaliyetli projelerin tekli üst üste çakıştırma alternatiflerinin sonuçları ise Tablo 4.12'de verilmiştir.

Tablo 4.12. 90 faaliyetli test problemlerinin tekli üst üste çıkıştırma sonuçları

Problem Numarası	Üst üste Çıkıştırma	Sonuç	Yineleme Sayısı	Problem Numarası	Üst üste Çıkıştırma	Sonuç	Yineleme Sayısı
19	5 ve 11	95	0	20	4 ve 7	80	0
	11 ve 38	92	0		7 ve 8	78	0
	38 ve 39	94	1		8 ve 12	77	2
	39 ve 49	95	0		12 ve 16	77	0
	49 ve 57	88	1		16 ve 19	78	3
	57 ve 68	88	1		19 ve 28	80	0
	68 ve 71	89	0		51 ve 59	80	0
	71 ve 80	89	0		66 ve 75	80	0
	80 ve 85	90	0		78 ve 86	76	0
	88 ve 89	92	0		86 ve 89	80	0
21	11 ve 13	92	0	23	2 ve 5	92	0
	13 ve 17	91	0		5 ve 7	93	0
	17 ve 19	92	0		7 ve 8	97	0
	19 ve 36	89	1		8 ve 15	98	0
	36 ve 50	90	1		15 ve 25	98	0
	50 ve 51	90	1		25 ve 34	96	0
	51 ve 70	88	2		34 ve 47	97	3
	70 ve 78	89	0		47 ve 51	98	1
	78 ve 79	88	0		51 ve 55	90	0
	79 ve 80	89	0		55 ve 61	94	0
	80 ve 82	90	0		61 ve 80	95	0
	82 ve 88	91	0		80 ve 84	95	2
24	6 ve 18	84	0		84 ve 88	96	0
	18 ve 21	84	0		88 ve 91	97	0
	21 ve 27	84	0	25	5 ve 22	79	6
	49 ve 72	83	2		30 ve 40	80	5
	72 ve 77	84	3	27	2 ve 7	89	3
	84 ve 87	84	0		7 ve 8	96	1
	87 ve 88	82	0		8 ve 9	96	0
	88 ve 89	82	0		9 ve 11	96	0
26	8 ve 9	92	0		11 ve 15	94	2
	9 ve 11	93	0		15 ve 30	94	4
	40 ve 45	97	0		30 ve 31	95	0
	45 ve 51	97	0		31 ve 36	94	1
	51 ve 54	97	0		36 ve 41	95	1
	54 ve 58	97	0		41 ve 42	93	0
	58 ve 68	97	0		42 ve 70	94	6
	68 ve 72	97	0		70 ve 78	90	0
	72 ve 90	97	0		78 ve 79	96	0
					79 ve 87	96	0
					87 ve 90	91	0

Tablo 4.12’den görüldüğü gibi 90 faaliyetli proje grubu için yapılan tekli üst üste çakıştırma denemelerinin sonuçlarına da ilk yinelemelerde ulaşılmaktadır. Yapılan denemeler sonucunda her bir problem için en düşük proje bitiş süresini veren tekli üst üste çakıştırma sonuçları sırasıyla; 19 numaralı problem için 88, 20 numaralı problem için 77, 21 numaralı problem için 88, 23 numaralı problem için 86, 24 numaralı problem için 82, 25 numaralı problem için 79, 26 numaralı problem için 92, 27 numaralı problem için ise 88’dir. 22 numaralı problemin temel çizelgesinden elde edilen kritik yol, Bölüm 4.2’de açıklanan paralel kritik yol durumuna uyması nedeniyle yapılan tekli denemeler proje bitiş süresinin kısaltılması için yeterli değildir. Bu nedenle bu problemin tekli denemelerinin verilmesine gerek duyulmamıştır. 90 faaliyetli proje grubu için yapılan çoklu denemeler ise Tablo 4.13’de verilmiştir.

Tablo 4.13. 90 faaliyetli test problemlerinin çoklu üst üste çıkıştırma sonuçları

Problem Numarası	Çoklu Üst Üste Çıkıştırma Alternatifleri	Sonuç	Yineleme Sayısı
19	49 ve 57; 68 ve 71	81	0
	49 ve 51; 71 ve 80	81	0
	11 ve 38; 49 ve 57; 68 ve 71	77	0
	38 ve 39; 49 ve 57; 68 ve 71; 80 ve 85	73	0
20	16 ve 19; 78 ve 86	73	0
	12 ve 16; 78 ve 86	72	0
	8 ve 12; 16 ve 19; 78 ve 86	69	0
	7 ve 8; 12 ve 16; 78 ve 86	69	0
	7 ve 8; 12 ve 16; 51 ve 59; 78 ve 86	68	0
	8 ve 12; 16 ve 19; 51 ve 59; 78 ve 86	68	0
21	19 ve 36; 51 ve 70	84	0
	36 ve 50; 51 ve 70; 79 ve 80	81	0
	36 ve 50; 51 ve 70; 78 ve 79	80	0
	19 ve 36; 51 ve 70; 78 ve 79	79	0
	19 ve 36; 51 ve 70; 78 ve 79; 80 ve 82	76	0
	13 ve 17; 19 ve 36; 51 ve 70; 78 ve 79; 80 ve 82	74	0
22	83 ve 90; 86 ve 89	100	0
	83 ve 90; 77 ve 82 ; 86 ve 89	99	1
	44 ve 49; 77 ve 82; 83 ve 90; 86 ve 89	93	1
	48 ve 50; 44 ve 49; 50 ve 66	95	2
	48 ve 50; 44 ve 49; 50 ve 66; 77 ve 82	94	1
23	2 ve 5; 55 ve 61; 80 ve 84	85	0
	5 ve 7; 55 ve 61; 80 ve 84	86	0
	2 ve 5; 25 ve 34; 51 ve 55; 80 ve 84	79	0
24	84 ve 87; 88 ve 89	80	0
	49 ve 72; 84 ve 87; 88 ve 89	78	0
25	5 ve 22; 30 ve 40	79	20
	59 ve 69; 64 ve 66; 77 ve 79	80	2
	77 ve 79; 30 ve 40; 22 ve 24	80	3
	77 ve 79; 30 ve 40; 5 ve 22	79	13
26	9 ve 11; 40 ve 45	92	0
	8 ve 9; 40 ve 45	91	0
	8 ve 9; 40 ve 45; 72 ve 90	91	0
27	2 ve 7; 70 ve 78; 87 ve 90	78	0
	2 ve 7; 41 ve 42; 70 ve 78; 87 ve 90	75	0
	2 ve 7; 15 ve 30; 41 ve 42; 70 ve 78; 87 ve 90	73	0

90 faaliyetli proje grubu için yapılan çoklu üst üste çakıştırma denemeleri için en düşük sonucu veren alternatifler Tablo 4.13’de koyu renkle belirtilmiştir. Tekli üst üste çakıştırma ile proje bitiş süresinde herhangi bir kısalma sağlanamayan 22 numaralı problemin çoklu sonuçları incelendiğinde projenin bitiş süresinin 93’e kadar düşürülebildiği görülmüştür. Bu sonuçtan da görüldüğü gibi, herhangi bir faaliyetten sonra kesişmeyen paralel yol içeren projelerde, bitiş süresinin kısaltılabilmesi için çoklu üst üste çakıştırma içeren algoritmanın kullanılması gerekir.

90 faaliyetli proje grubunun tekli ve çoklu üst üste çakıştırma sonucunda proje süresinde elde edilen düşme oranları Tablo 4.14’de özetlenmiştir.

Tablo 4. 14. 90 Faaliyetli Projeler için Üst Üste Çakıştırma ile Elde Edilen Süre Kazançları

Problem Numarası	Üst Üste Çakıştırma İçermeyen Sonuç	Tekli Üst Üste Çakıştırma ile Elde Edilen En İyi Sonuç	Kazanç (%)	Çoklu Üst Üste Çakıştırma ile Elde Edilen En İyi Sonuç	Kazanç (%)
19	96	88	8,33	73	23,96
20	81	76	6,17	68	16,05
21	93	88	5,38	74	20,43
22	101	-	Kazanç yok	93	7,92
23	98	90	8,16	79	19,38
24	85	82	3,53	78	8,24
25	83	79	4,82	79	4,82
26	98	92	6,12	91	7,14
27	97	89	8,25	73	24,74

90 faaliyetli projeler için tekli üst üste çakıştırma ile elde edilen proje bitiş süresindeki kısaltmalar, 22 numaralı problem haricinde, %3,53 ile % 8.51 arasında değişmektedir. 22 numaralı problemde tekli üst üste çakıştırma ile bitiş süresinde kısalma sağlanamazken, yapılan çoklu üst üste çakıştırma denemeleri ile bitiş süresinde %7,92’ye kadar düşüş sağlanabilmiştir. En fazla süre kazancı ise % 24,74 ile 27 numaralı projede elde edilmiştir.

Tablo 4.15’de ise, 120 faaliyetli projelerin tekli üst üste karşılaştırma sonuçları verilmiştir.

Tablo 4.15. 120 faaliyetli test problemlerinin tekli üst üste karşılaştırma sonuçları

Problem Numarası	Üst üste Karşılaştırma	Sonuç	Yineleme Sayısı	Problem Numarası	Üst üste Karşılaştırma	Sonuç	Yineleme Sayısı
29	102 ve 112	71	52	30	21 ve 24	91	1
	8 ve 37	73	10		24 ve 50	87	0
31	4 ve 20	104	2	36	95 ve 100	88	0
	20 ve 21	101	10		100 ve 106	88	0
	21 ve 32	102	6		106 ve 107	91	0
	32 ve 44	101	7		109 ve 120	86	0
	44 ve 47	99	10		4 ve 7	90	27
	47 ve 84	103	29		7 ve 9	90	11
	84 ve 87	104	1		9 ve 12	90	24
	87 ve 95	105	2		12 ve 13	90	10
	102 ve 103	100	43		13 ve 47	91	5
	103 ve 108	101	8		47 ve 50	91	24
	108 ve 112	105	1		50 ve 52	91	4
	112 ve 116	106	2		52 ve 61	91	6
	116 ve 121	103	4		61 ve 84	90	12
32	107 ve 108	92	2		113 ve 117	89	36
	100 ve 102	92	0		117 ve 118	90	13

Tablo 4.15’de 28, 33, 34 ve 35 numaralı projelerin tekli üst üste karşılaştırma sonuçları yer almamaktadır. Bu projelerin paralel kritik yol içermesi nedeniyle tekli üst üste karşılaştırma ile herhangi bir süre kısalması sağlanamamaktadır. 120 faaliyetli projelerin bazı karşılaştırma alternatiflerinin çözüme ulaştığı yineleme sayısı, diğer proje gruplarındaki problemlerin (J30, J60, J90) çözüme ulaştığı yineleme sayısına göre daha fazladır. 120 faaliyetli projelerin, gerek faaliyet sayısı gerekse çözümü zorlaştıran diğer faktörler (kaynak kısıtlılığı, ağ karmaşıklığı) dikkate alındığında bu yükselme beklenen bir durumdur. Mevcut proje grubunun çözümü zorlaştıracı bu özelliklerine rağmen, çözüme ulaşma açısından 100 yinelemenin yeterli olduğu görülmektedir. Mevcut proje grubu içinde tekli üst üste karşılaştırma ile elde edilen en düşük süreler Tablo 4.15’de koyu renk ile belirtilmiştir.

120 faaliyetli problemlerin çoklu üst üste karşılaştırma sonuçları ise Tablo 4.16’da verilmiştir.

Tablo 4.16. 120 faaliyetli test problemlerinin çoklu üst üste çakıştırma sonuçları

Problem Numarası	Çoklu Üst Üste Çakıştırma Alternatifleri	Sonuç	Yineleme Sayısı
28	15 ve 32; 46 ve 64; 116 ve 119	70	150
29	8 ve 32; 102 ve 112	70	65
30	100 ve 106; 107 ve 109	88	1
	95 ve 100; 106 ve 107	87	1
	21 ve 24; 95 ve 100; 106 ve 107	86	0
	24 ve 50; 95 ve 100; 106 ve 107	82	2
	24 ve 50; 109ve 120	81	4
31	4 ve 20; 21 ve 32; 44 ve 47; 102 ve 103; 116 ve 121	95	14
	4 ve 20; 21 ve 32; 44 ve 47; 84 ve 87; 102 ve 103; 116 ve 121	93	51
32	100 ve 102; 107 ve 108	91	2
33	110 ve 121; 78 ve 98; 37 ve 41	81	36
	4 ve 6; 53 ve 97; 78 ve 98; 110 ve 121	81	28
	4 ve 6; 17 ve 26; 23 ve 58; 98 ve 109; 108 ve 110	80	22
34	91 ve 99; 101 ve 104; 115 ve 120	105	34
35	74 ve 75; 97 ve 102; 95 ve 105; 106 ve 117	98	9
	70 ve 74; 97 ve 102; 104 ve 109; 106 ve 117	96	37
	49 ve 58; 87 ve 97; 95 ve 105; 104 ve 109	92	13
36	12 ve 13; 61 ve 84; 113 ve 117	86	50
	4 ve 7; 12 ve 13; 61 ve 84; 113 ve 117	86	23
	4 ve 7; 12 ve 13; 61 ve 84; 117 ve 118	87	49

Tablo 4.16'dan, tekli üst üste çakıştırma ile proje bitiş süresinde herhangi bir süre kısalma sağlanamayan 28, 33, ve 35 numaralı projelerin bitiş sürelerinin denen çoklu üst üste çakıştırma alternatifleri ile sırasıyla 70, 80 ve 92'ye kadar düşürülebildiği görülmüştür. 28 numaralı problemin hem kaynak gücü hem de ağ karmaşıklığının düşük olması nedeniyle yineleme sayısının diğer problemlere göre daha yüksek alınmıştır. Bu problem için yineleme sayısı 400 alınmış ve mevcut sonuca 150. yinelemede ulaşılmıştır.

120 faaliyetli projeler için tekli ve çoklu üst üste çakıştırma ile elde edilen süre kazançları Tablo 4.17'de verilmiştir.

Tablo 4.17. 120 faaliyetli projeler için üst üste çakıştırma ile elde edilen süre kazançları

Problem Numarası	Üst Üste Çakıştırma İçermeyen Sonuç	Tekli Üst Üste Çakıştırma İle Elde Edilen En İyi Sonuç	Kazanç (%)	Çoklu Üst Üste Çakıştırma ile Elde Edilen En İyi Sonuç	Kazanç (%)
28	71	-	Kazanç yok	70	1,4
29	74	71	4,05	70	5,41
30	92	86	6,52	81	11,96
31	107	99	7,48	93	13,08
32	93	92	1,08	91	2,15
33	82	-	Kazanç yok	80	2,44
34	105	-	Kazanç yok	105	Kazanç yok
35	100	-	Kazanç yok	92	8
36	91	89	2,19	86	5,49

Tekli üst üste çakıştırma ile kazanç elde edilemeyen 28, 33, ve 35 numaralı problemler için, çoklu üst üste çakıştırma yapılarak proje bitiş süresinde sırasıyla %1,4, %2,44 ve %8'lik kısaltmalar sağlanmıştır. Diğer projeler için de tekli üst üste çakıştırma ile elde edilen süre kazançlarının, çoklu üst üste çakıştırmanın uygulanması ile önemli oranlarda arttığı görülmektedir.

Yapılan denemeler neticesinde, farklı ağ karmaşıklığı ve kaynak gücüne sahip 30, 60, 90 ve 120 faaliyetli projelere ait bütün üst üste çakıştırma alternatifleri için makul sürelerde çözümlerin elde edildiği görülmüştür. Algoritmanın çoklu üst üste çakıştırmaya imkan verecek şekilde düzenlenmesi ile, tekli üst üste çakıştırma ile bitiş süresinde iyileşme sağlanamayacak olan örnekler için çoklu üst üste çakıştırma ile kısalma sağlayan alternatiflerin ortaya çıkarılması mümkün hale gelmiştir. Ayrıca, tekli üst üste çakıştırma ile belirli bir seviyeye kadar kısalma sağlansa bile çoklu üst üste çakıştırma alternatiflerinin denenmesi ile proje bitiş sürelerinde daha yüksek oranlarda düşüşler sağlanabilmektedir.

Çoklu üst üste çakıştırmaya imkan veren programın diğer bir avantajı ise, düşünülen çoklu üst üste çakıştırma alternatiflerinin sonuçlarını kısa süreler içerisinde vermesi ve bu sayede istenilen sayıda farklı alternatiflerin makul süreler

içerisinde değerlendirilmesine imkan vermesidir. Bu özellikle, sadece tekli üst üste çakıştırma sonuçlarına bakarak çoklu alternatiflere karar vermenin yanlış olduğu durumlarda önemli bir avantaj sağlar. Örneğin, yapılan denemelerde bazı örneklerde, tekli üst üste çakıştırma durumunda oldukça iyi sonuç veren iki faaliyet çifti arasındaki üst üste çakıştırma durumu birlikte ele alındığında beklenen sonucu vermediği görülmüştür. Eğer algoritma çoklu durumu içerecek şekilde genişletilmiş olmasaydı bu gibi bir durumda yanlış bir çoklu üst üste çakıştırma kararı verilebilirdi. Benzer şekilde, tekli olarak iyi sonuç vermeyen bir üst üste çakıştırma başka faaliyet çiftleri arasındaki üst üste çakıştırma ile birlikte düşünüldüğünde daha iyi bir sonuç verebildiği de karşılaşılan özel durumlardan biridir. Sadece tekli üst üste çakıştırma sonuçlarına bakılarak çoklu üst üste çakıştırmaya karar verilmesi durumunda bu alternatif gözden kaçırılmış olacaktır.

4.5. Senaryo Analizleri

Bu bölümde, üst üste çakıştırmadan elde edilebilecek süre kazancını etkileyen iki faktör olan alt seviye faaliyetin duyarlılığı ve üst seviye faaliyetin gelişim fonksiyonu parametreleri dikkate alınarak farklı senaryo durumları ortaya konmaktadır. Bu amaçla, üst üste çakıştırma içeren PSO programı, Wang(2009) makalesinde verilen bir elektronik yeni ürün geliştirme projesi için farklı derecelerdeki alt seviye faaliyet duyarlılığı ve üst seviye faaliyet gelişimi durumları dikkate alınarak çalıştırılmıştır.

Ele alınan proje ağı, başlangıç ve bitiş kukla faaliyetler haricinde 51 faaliyetten oluşmaktadır. Projede toplam 4 adet yenilenebilir kaynak kullanılmaktadır. İlgili makalede her bir faaliyetin mevcut kaynaklardan ihtiyaç duydukları miktar, proje faaliyetlerinin öncelik ilişkileri ve süreleri verilmiştir. Ancak, makalede yapılan çalışma gereği faaliyetlerin süreleri bulanık sayılarla olarak verilmiştir. Yapılan çalışmada, faaliyetlerin süreleri ile ilgili verilen bulanık sayıların orta değeri alınmıştır.

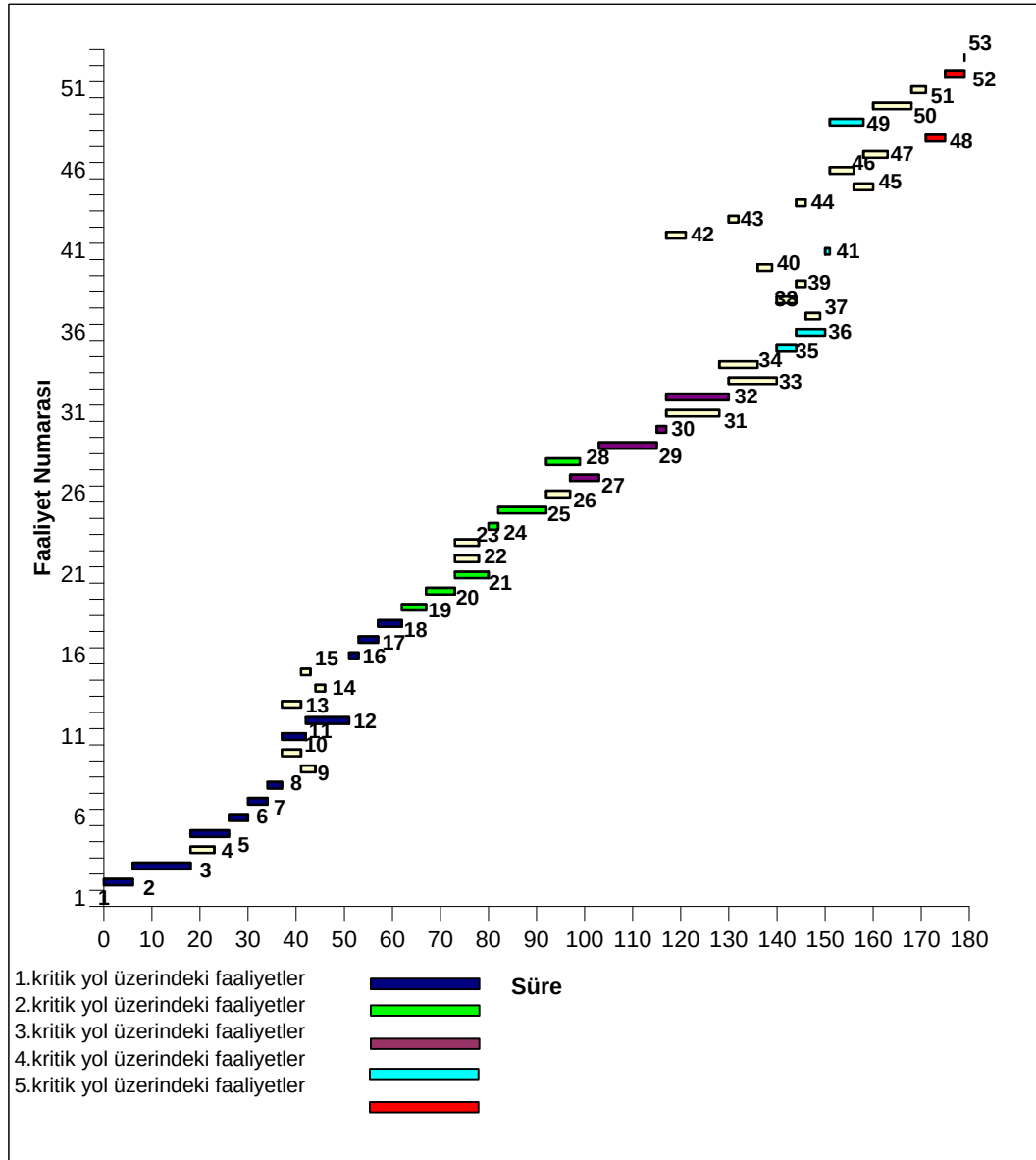
Senaryo analizinde, alt seviye faaliyet duyarlılığı için yüksek ve düşük olmak üzere iki farklı seviye belirlenmiştir. Düşük seviyede; alt seviye faaliyetin

duyarlılığını belirleyen çakıştırma etkisi, faaliyetin süresinin %10'u kadar alınmış, yüksek seviyede ise; faaliyetin süresinin %40'ı kadar alınmıştır. Üst seviye faaliyetin gelişim parametresi ise üç sevide ele alınmıştır ($\alpha=0,5$; $\alpha=1$; $\alpha=2$). Bu şekilde, alt seviye faaliyet duyarlılığı ve üst seviye gelişim parametresi faktörlerinin farklı seviyelerinde üst üste çakıştırmadan elde edilebilecek süre kazancı ortaya konulmaya çalışacaktır. Ele alınan farklı senaryo durumları Tablo 4.18'de verilmiştir.

Tablo 4.18. Senaryo durumları

Senaryo Numarası	Alt Seviye Faaliyetin Duyarlılığı	Üst Seviye Faaliyetin Gelişim Parametresi
Senaryo 1	$(ovrimp_k = 0,4.td_k)$	$(a = 3)$
Senaryo 2		$(a = 1)$
Senaryo 3		$(a = 0,3)$
Senaryo 4	$(ovrimp_k = 0,1.td_k)$	$(a = 3)$
Senaryo 5		$(a = 1)$
Senaryo 6		$(a = 0,3)$

Test problemlerinde de yapıldığı gibi, yeni ürün geliştirme projesinde de öncelikli olarak üst üste çakıştırma yapılabilecek alternatif faaliyetlerin belirlenmesi gerekmektedir. Bu amaçla öncelikle projenin üst üste çakıştırma içermeyen temel çizelgesi oluşturulmuş ve kritik yol analizi yapılmıştır. Projenin GANNT Şeması Şekil 4.13'de verilmiştir.



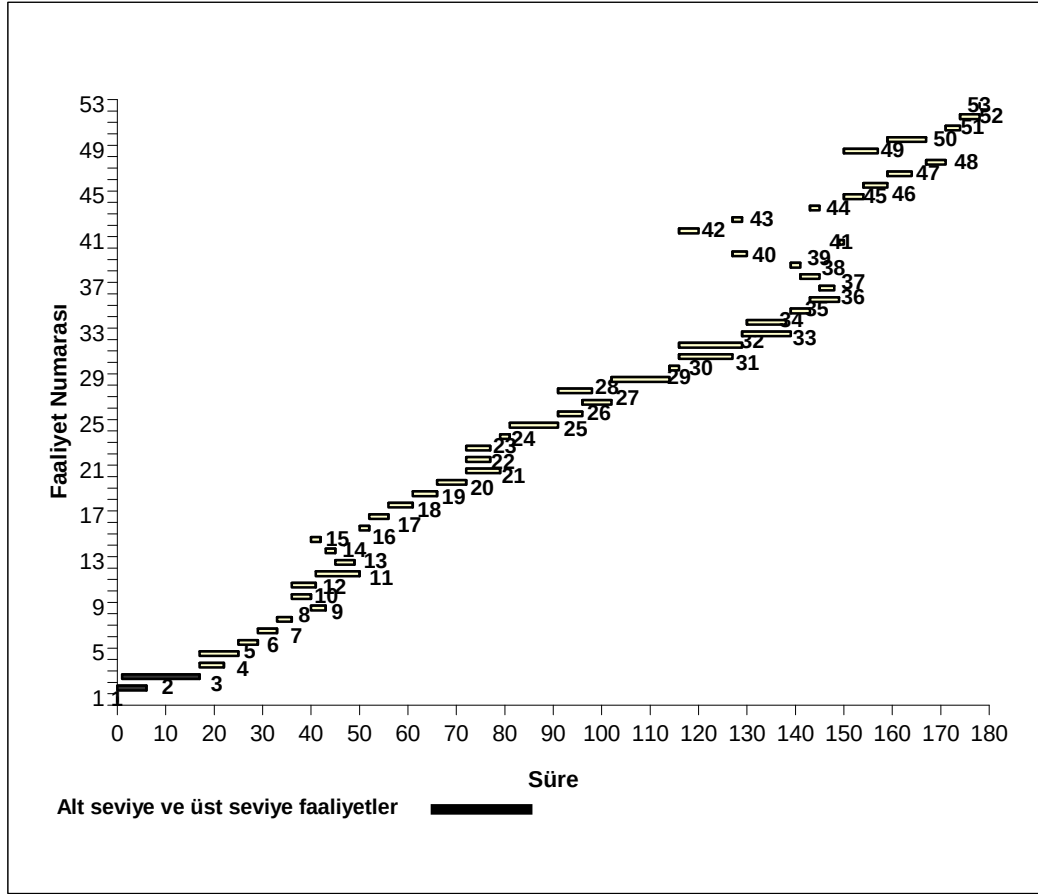
Şekil 4.13. Bir elektronik ürün geliştirme projesine ait GANNT şeması

Temel çizelge üzerinde yapılan analizler sonucunda Şekil 4.13’de farklı renklerle belirtilmiş 5 tane kritik yol olduğu ortaya çıkmıştır. Üst üste karşılaştırma yapılacak faaliyet çiftleri, bu kritik yollardan herhangi biri üzerinde bulunan ve aralarında bilgi alışverişi şeklinde ilişki bulunan faaliyetler arasından seçilmiştir. Farklı tekli üst üste karşılaştırma alternatifleri ve bu alternatiflerin her bir senaryo durumundaki proje bitiş süreleri Tablo 4.19’da verilmiştir.

Tablo 4.19. Elektronik ürün geliştirme projesinin farklı senaryo analizlerine göre tekli üst üste çakıştırma sonuçları

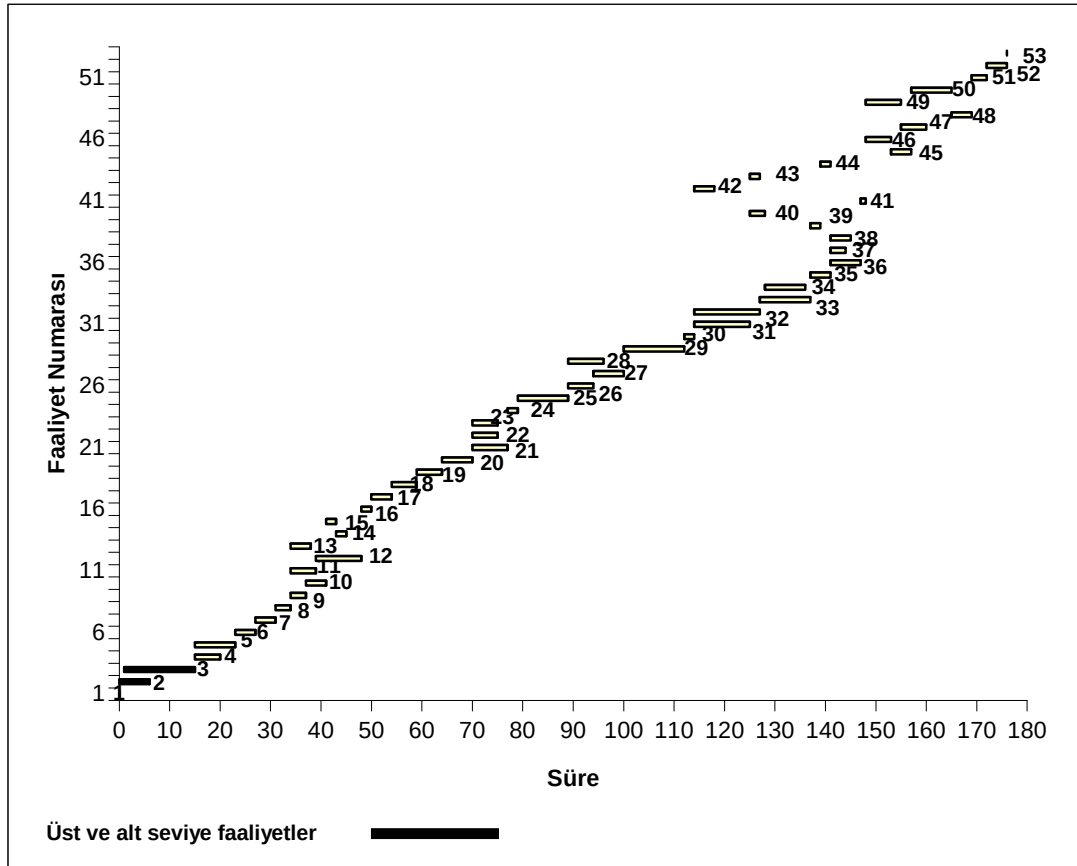
Üst Üste Çakıştırma Alternatifleri	Proje Bitiş Süreleri					
	S1	S2	S3	S4	S5	S6
2 ve 3	179	178	176	176	175	175
3 ve 5	176	176	176	176	176	176
5 ve 6	178	177	177	177	177	177
6 ve 7	178	178	177	177	177	177
7 ve 8	178	177	177	177	177	177
11 ve 12	179	179	179	178	178	178
20 ve 21	177	177	177	177	177	177

2 ve 3 faaliyetleri arasındaki üst üste çakıştırma sonuçları incelendiğinde, hem alt seviye faaliyetin duyarlılığının yüksek olduğu hem de α 'nın 3 alındığı ilk senaryo için hesaplanan yeniden çalışma miktarının çok yüksek olması nedeniyle, 2 ve 3 faaliyetleri arasında üst üste çakıştırma yapılmamış ve bitiş süresi 179 olarak bulunmuştur. Alt seviye faaliyetin duyarlılığının yüksek olduğu ancak, α parametresinin 1 alındığı senaryo 2'de ise proje bitiş süresi 1 zaman birimi düşmüştür. Bu senaryonun GANNT şeması Şekil 4.14'de verilmiştir. Şekilde görüldüğü gibi üst seviye faaliyet başladıktan 1 zaman birimi sonra alt seviye faaliyet başlamış ve üst seviye faaliyetin bitiş süresi yeniden çalışma miktarıyla birlikte 17 olmuştur. Üst üste çakıştırma olmayan durumda 3 numaralı faaliyetin bitiş süresi 18 iken bu durumda bahsedilen faaliyetin bitiş süresi 1 zaman birimi öne çekilmiş ve sonuç olarak proje bitiş süresi 1 zaman birimlik kısalma sağlanmıştır.



Şekil 4.14. Senaryo 2 durumunda, elektronik ürün geliştirme projesinin 2 ve 3 numaralı faaliyetleri arasında üst üste çakıştırmaya içeren GANNT şeması

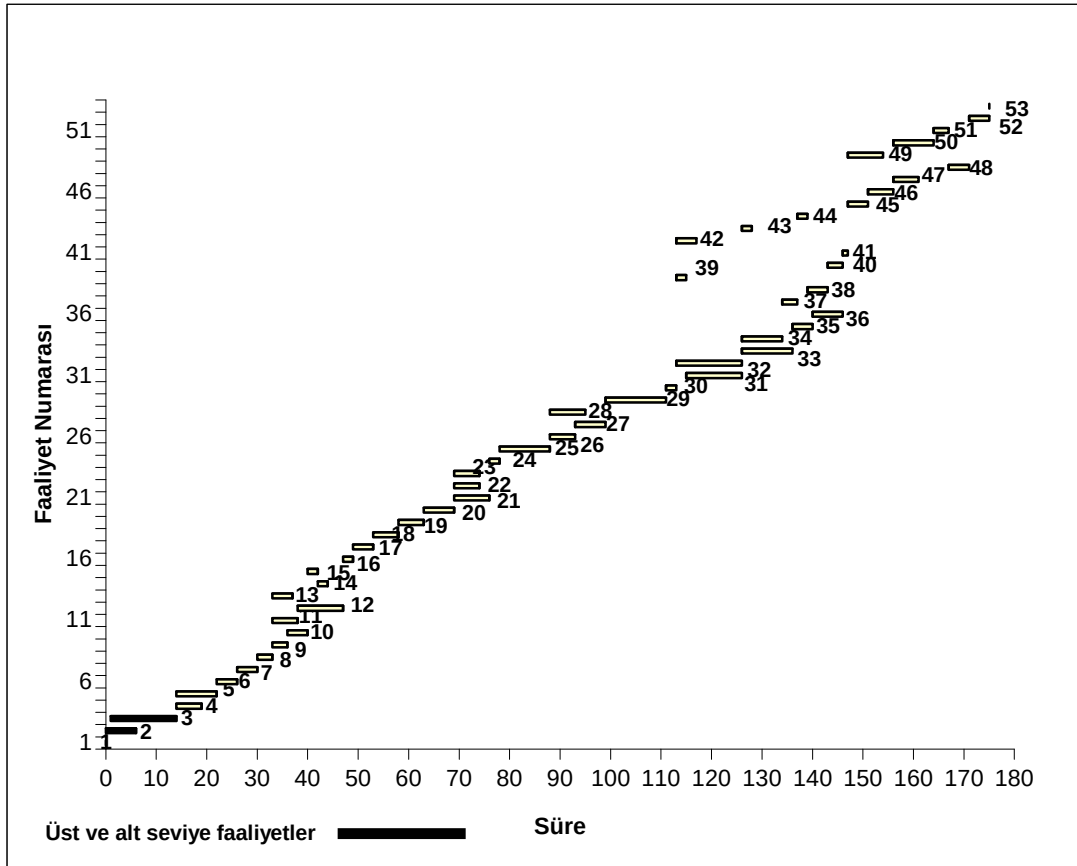
Senaryo 3’de ise senaryo 2’de olduğu gibi üst seviye faaliyet başladıktan 1 zaman birimi sonra alt seviye faaliyet başlamış ancak α parametresi daha da düştüğü için hesaplanan yeniden çalışma miktarı 2 zaman birimine düşmüş ve 3 numaralı faaliyet $t=15$ ’de bitmiştir. Bu nedenle proje bitiş süresi 176’ya düşürülmüştür (Şekil 4.15).



Şekil 4.15. Senaryo 3 durumunda, elektronik ürün geliştirme projesinin 2 ve 3 numaralı faaliyetleri arasında üst üste çakıştırma içeren GANNT şeması

Senaryo 4 de senaryo 2 ve senaryo 3 deki gibi 2 ve 3 faaliyetleri arasındaki en uygun üst üste çakıştırma miktarı 5 zaman birimi olarak bulunmuş, ve bu miktara bağlı olarak yeniden çalışma miktarı 2 olarak hesaplanmıştır. Bu durumda da, proje bitiş süresi tekrar 176 olarak bulunmuştur. Bu senaryonun GANNT şeması Şekil 4.15.ile aynıdır.

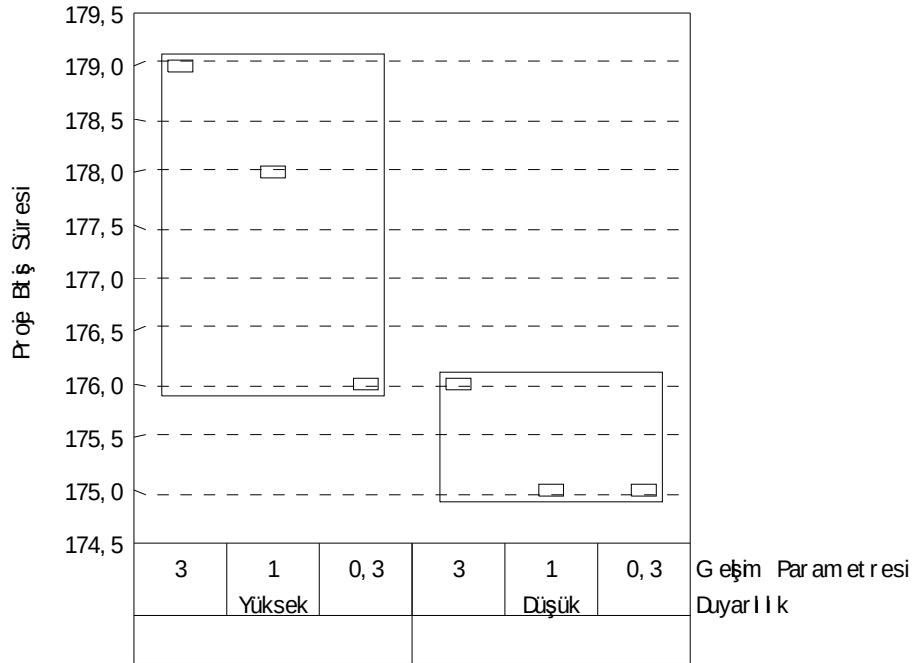
Alt seviye faaliyetin duyarlılığının düşük olduğu ve gelişim hızı parametresinin 1 olduğu senaryo 5’de de, en uygun üst üste çakıştırma miktarı 5 olarak hesaplanmış ancak hem gelişim hızın parametresinin düşük olması hem de alt seviye faaliyetin duyarlılığının düşük olması nedeniyle yeniden çalışma miktarı 1 olarak hesaplanmış ve 3 numaralı faaliyetin bitişine eklenmiştir. Bu durumda 3 numaralı faaliyet 14 de tamamlanmıştır. Sonuç olarak proje bitiş süresi de 175 düşmüştür(Şekil 4.16)



Şekil 4.16. Senaryo 5 durumunda, elektronik ürün geliştirme projesinin 2 ve 3 numaralı faaliyetleri arasında üst üste çakıştırma içeren GANNT şeması

Senaryo 6'da ise üst üste çakıştırma miktarı ve buna bağlı olarak hesaplanan yeniden çalışma miktarı senaryo 5 ile aynıdır. Bu nedenle bu senaryonun GANNT şeması da Şekil 4.16'da verildiği gibidir.

Alt seviye faaliyetin duyarlılığı ve üst seviye faaliyetin gelişim parametresinin etkilerini daha net görebilmek amacıyla 2 ve 3 faaliyetleri arasındaki üst üste çakıştırmayı içeren proje çizelgesinin tanımlanan 6 senaryo için bitiş süreleri şekil 4.17'de özetlenmiştir.

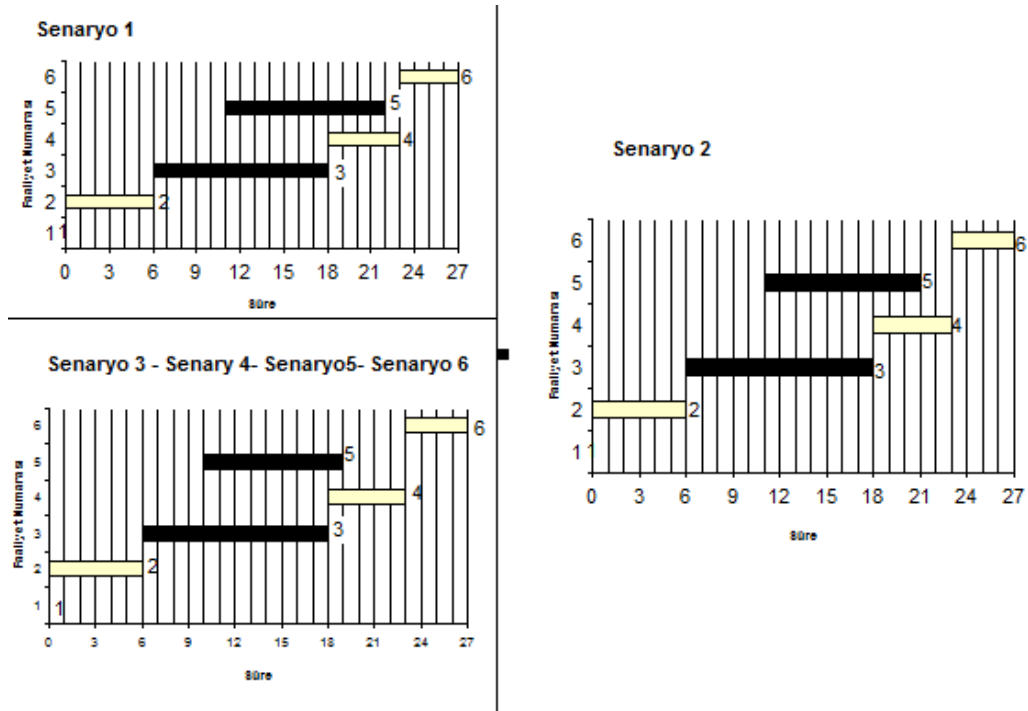


Şekil 4.17. Elektronik ürün geliştirme projesinin 2 ve 3 numaralı faaliyetleri arasında üst üste çakıştırma içeren çizelgenin farklı senaryolara göre sonuçları

Şekilde görüldüğü gibi alt seviye faaliyetin duyarlılığı arttıkça yeniden çalışma miktarındaki artışa bağlı olarak üst üste çakıştırmadan elde edilecek kazanç düşebilmektedir. Aynı şekilde, üst seviye faaliyetin gelişim parametresinin değerinin artması da yeniden çalışma miktarını artırabilmekte ve süre kazancının düşmesine neden olabilmektedir. Gelişim parametresinin, Bölüm3’de verilen gelişim fonksiyonu üzerindeki etkisi dikkate alındığında bu beklenen bir durumdur. α parametresi, 1 den büyük bir değer olduğunda bu değer arttıkça yeniden çalışma miktarı da artış yönünde bir eğilim gösterecektir. Mevcut parametre 0 ile 1 arasında olduğunda ise, parametre değeri küçüldükçe yeniden çalışma miktarı azalış yönünde bir eğilim gösterecektir.

3 ve 5 faaliyetleri arasında üst üste çakıştırma yapılması durumunda ise, bütün senaryo durumları için aynı sonucun elde edildiği görülmüştür. Bu gibi sonuçlar ilk bakışta şaşırtıcı gelse de her durum için elde edilen çizelgelerin incelenmesi ile durumu açıklamak mümkündür. 5 numaralı faaliyetin bitişinin

ardından başlayacak olan faaliyet 6 numaralı faaliyetdir ancak 6 numaralı faaliyetin başlama süresini etkileyen faaliyetler kendinden önce gelen hem 5 hem de 4 numaralı faaliyetlerdir. Bu nedenle, bazı senaryolarda, farklı üst üste çakıştırma miktarları hesaplanarak uygulanıp 5 numaralı faaliyet daha erken bitirildiğinde bile, 6 numaralı faaliyet 4 numaralı faaliyet bitmediği için daha erken başlayamamaktadır. Bu durumu daha net gösterebilmek amacıyla ilk 6 faaliyetin çizelgesi Şekil 4.18’de verilmiştir.



Şekil 4.18. Elektronik ürün geliştirme projesinin 3 ve 5 numaralı faaliyetleri arasındaki üst üste çakıştırmanın senaryo analizleri sonuçları

20 ve 21 faaliyetleri arasında üst üste çakıştırma içeren çizelgenin de bütün senaryo durumları için aynı sonucu vermesinin nedeni de 3 ve 5 faaliyetleri arasındaki üst üste çakıştırma da açıklanan nedenle aynıdır.

Farklı üst üste çakıştırma alternatifleri birlikte denenerek proje bitiş süresinde daha fazla kısalma sağlamak mümkündür. Bu nedenle her bir senaryo için ele alınan çoklu üst üste çakıştırma alternatifleri denenmiş ve sonuçlar Tablo 4.23’de verilmiştir.

Tablo 4.20 Elektronik ürün geliştirme projesinin farklı senaryo analizlerine göre çoklu üst üste çakıştırma sonuçları

Çoklu Üst Üste Çakıştırma Alternatifleri	Proje Bitiş Süreleri					
	S1	S2	S3	S4	S5	S6
2 ve 3 ; 5 ve 6	178	176	176	174	173	173
2 ve 3 ; 5 ve 6; 11 ve 12	178	176	173	173	172	172
3 ve 5; 6 ve 7	175	175	174	174	174	174
2 ve 3; 7 ve 8; 20 ve 21	176	174	174	172	171	171

Çoklu üst üste çakıştırma alternatiflerini içeren problemin her bir senaryo durumları için çözülmesi neticesinde Tablo 4.20’de yer alan sonuçlar elde edilmiştir. Bu sonuçlara göre, alt seviye faaliyetin duyarlılık derecesi yüksek ve gelişim parametresinin 3 alındığı senaryo 1’de en düşük proje bitiş süresini sağlayan çoklu üst üste çakıştırma sonucu 3 ve 5 faaliyet çiftleri ve 6 ve 7 faaliyet çiftleri arasında üst üste çakıştırma yapılması ile 175 olarak bulunmuştur. Alt seviye faaliyetin duyarlılık derecesi yüksek ve gelişim parametresinin 1 alındığı senaryo 2’de 2 ve 3, 7 ve 8, 20 ve 21 faaliyet çiftleri arasındaki üst üste çakıştırmayı içeren çizelgenin bitiş süresi aynı senaryoda diğer alternatiflere göre daha iyi sonuç vermiştir. Senaryo 3’de ise, en düşük bitiş süresi 2 ve 3, 5 ve 6, 11 ve 12 arasında üst üste çakıştırmayı içeren çizelgede elde edilmiştir. Diğer senaryolar için denen alternatifler arasında en düşük bitiş süresini veren çizelge ise, 2 ve 3, 7 ve 8, 20 ve 21 faaliyetleri arasında üst üste çakıştırmayı içeren çizelge olmuştur. Görüldüğü gibi her bir senaryo için en iyi çözümü veren alternatif değişebilmektedir. Geliştirilen algoritma, farklı alternatiflerin değişik senaryo durumlarına göre sonuçlarını çok kısa süreler içerisinde vererek, üst üste çakıştırma yapılacak alternatiflerin daha doğru bir şekilde belirlenebilmesine olanak sağlamaktadır.

5. SONUÇLAR VE ÖNERİLER

5.1. Çalışmanın Özeti

Bu çalışmada, yeni ürün geliştirme projelerinde proje bitiş süresinin kısaltılması amacıyla kullanılan bir uygulama olan faaliyetlerin üst üste çakıştırılması uygulamasını da içeren kaynak kısıtlı proje çizelgeleme problemi ele alınmıştır. Söz konusu problemin modellenmesi aşamasında öncelikle, faaliyetlerin üst üste çakıştırma uygulamasının modele nasıl dahil edileceği araştırılmış ve yapılan literatür çalışmaları incelenmiştir. Mevcut çalışmalarda, üst üste çakıştırma miktarının belirlenmesi sırasında, çakıştırmadan elde edilebilecek kazanç ve çakıştırmadan kaynaklanan yeniden çalışma miktarının dengelenmesi gerektiği vurgulanmaktadır. Bu noktadan hareketle, yeniden çalışma miktarının hesaplanması için, yeniden çalışma miktarını etkilediği bilinen, alt seviye faaliyetin duyarlılığı ile üst seviye faaliyetin gelişim hızı faktörlerini de içeren bir formülasyon ortaya konmuştur. Geliştirilen matematiksel model, kaynak kısıtları ve faaliyetler arasındaki öncelik ilişkilerini de dikkate alarak proje bitiş süresini en küçükleyecek şekilde üst üste çakıştırma miktarını belirlemekte ve proje çizelgesini oluşturmaktadır.

Matematiksel modelin ortaya konmasından sonra ikinci adım, mevcut problemi optimum veya optimuma yakın sonuç verecek şekilde çözecek çözüm yönteminin araştırılması olmuştur. İlk olarak problemin optimum çözümü için mevcut modelin GAMS kodları oluşturulmuş ve bu paket programda hangi büyüklüğe kadar problemlerin çözülebildiği ortaya konmuştur. Test amaçlı olarak, kaynak kısıtlı proje çizelgelemede kıyaslama problemi olarak sıklıkla kullanılan J30 proje verileri kullanılmış ve seçilen 9 adet problemin farklı üst üste çakıştırma alternatifleri çözülmeye çalışılmıştır. Mevcut proje ağları, çözümü zorlaştırdığı bilinen kaynak gücü ve proje ağı karmaşıklığı gibi faktörlere göre derecelendirilmiştir. Ele alınan problem, çözüm karmaşıklığı olduğu bilinen bir problemdir ve faaliyet sayısının yanı sıra bahsedilen bu faktörlerde problemin çözüm süresi üzerinde oldukça etkilidir. Bu nedenle, GAMS programında yapılan denemelerde de, 30 faaliyetli proje ağları içerisinde bile çözülemeyen problemler

ortaya çıkmıştır. Özellikle, kaynak gücünün düşük olduğu proje ağlarında tanımlanan yineleme sayısı içerisinde çoğunlukla çözüm bulunamamıştır. Bu nedenle, sonraki aşamada, ortaya konulan problemin, farklı faaliyet sayısı, ağ karmaşıklığı ve kaynak gücüne sahip projeler için makul süreler içerisinde optimum ve optimuma yakın şekilde çözecek sezgisel yöntemin geliştirilmesi hedeflenmiştir. Yapılan araştırmalar sonucunda, Genetik Algoritmaya göre daha az parametre ayarı gerektiren, uygulaması daha kolay olan ve genetik algoritmaya yakın çözümler verdiği belirtilen Parçacık Sürüsü Optimizasyonu kullanılmaya karar verilmiştir. Kullanılacak sezgisel algoritmaya karar verilmesinin ardından bu ileri sezgisel yöntem probleme uyarlanmıştır. Bu aşamada, parçacıkların temsilinde öncelik temelli gösterim kullanılmış, parçacıkların çizelgeye dönüştürülmesinde ise üst üste çakıştırmayı içeren probleme uyarlanmış seri çizelgeleme planı kullanılmıştır. Mevcut algoritma aynı zamanda birden fazla çakıştırmaya olanak verecek şekilde genişletilmiştir. Algoritmanın geliştirilmesinin ardından, algoritmanın bilgisayar ortamında kodlanması C# programlama dili kullanılarak gerçekleştirilmiştir. Kullanıcı kolaylığı sağlaması açısından kodlama aşamasında veri girişini kolaylaştıran bir ara yüz oluşturulmuştur.

Probleme uyarlanmış parçacık sürüsü optimizasyonu algoritmasının geliştirilmesi ve kodlama işlemlerinin bitmesinden sonra algoritmanın test edilmesi aşamasına geçilmiştir. Bu aşamada, farklı faaliyet sayısı, ağ karmaşıklığı ve kaynak gücü faktörlerine göre sınıflandırılmış olan J30, J60, J90 ve J120 test problemlerinin her birinden dokuz tane örnek kullanılmıştır. Öncelikle, üst üste çakıştırmayı içermeyen algoritma test edilmiş ve sonuçlar verilmiştir. Bunun ilk nedeni, literatürde, üst üste çakıştırmayı içeren kaynak kısıtlı modelleme problemi ilk defa bu şekilde ele alındığı için kıyaslama yapacak veri bulunmamaktadır. İlk olarak üst üste çakıştırmayı içermeyen algoritma test edilmiş ve PSPLIB’de yer alan mevcut sonuçlarla kıyaslanarak, algoritmanın etkinliği hakkında fikir sahibi olunmuştur. Ayrıca, üst üste çakıştırmayı içermeyen algoritmanın kullanılması ile elde edilen temel çizelge, üst üste çakıştırmayı yapılacak faaliyetlerin belirlenmesinde kullanıldığı için bu algoritmanın en iyi veya en iyiye yakın çözüm vermesi üst üste çakıştırmayı yapılacak faaliyetleri doğru şekilde belirlenmesi açısından oldukça önemlidir. Ele

alınan her bir test problemi için temel çizelgeye göre belirlenmiş tekli ve çoklu üst üste çakıştırma alternatifleri ise üst üste çakıştırma içeren algoritma ile çözülmüş ve bu algoritmanın farklı faaliyet sayısı, kaynak gücü ve ağ karmaşıklığı içeren problemleri çözme üzerindeki etkinliği araştırılmıştır.

Son aşamada ise, Wang (2009) makalesinde verilen bir elektronik yeni ürün geliştirme projesi verileri kullanılarak farklı derecelerdeki alt seviye faaliyet duyarlılığı ve üst seviye faaliyetin gelişim parametresi faktörlerini dikkate alan senaryolar oluşturulmuş ve üst üste çakıştırmadan elde edilecek kazanç her senaryo durumu için incelenmiştir.

5.2. Sonuçlar

Yapılan çalışmada elde edilen sonuçlar beş grupta değerlendirilmiştir. Birinci grupta oluşturulan matematiksel modelin analitik çözümünden elde edilen sonuçlar incelenmiştir. İkinci grupta, probleme uyarlanmış Parçacık Sürü Optimizasyonu Algoritması ile elde edilen sonuçların incelenmesi ve algoritmanın test edilmesine dair analizler yer almaktadır. Bu aşamada ilk olarak, üst üste çakıştırma içermeyen algoritmanın sonuçları değerlendirilmiştir. Üçüncü grupta ise, bu algoritmadan elde edilen temel çizelge yardımıyla üst üste çakıştırma yapılması durumunda proje bitiş süresinin kısaltılmasını sağlayacak alternatiflerin nasıl belirlendiğine dair bilgilere yer verilmiştir. Dördüncü grupta, ele alınan test problemleri için belirlenen tekli üst üste çakıştırma ve çoklu üst üste çakıştırma sonuçları incelenmiştir. Son grupta ise, bir elektronik ürün geliştirme projesi verileri kullanılarak üst üste çakıştırmadan elde edilebilecek kazancı etkileyen faktörleri dikkate alan senaryo analizlerinin sonuçlarına yer verilmiştir.

İlk aşamada geliştirilen matematiksel model kullanılarak analitik olarak çözebilecek problemlerin sınırları belirlenmeye çalışılmıştır. Test amacıyla, kaynak gücü ve ağ karmaşıklığı faktörlerine göre sınıflandırılmış olan J30 verilerden 9 adet örnek proje ağı kullanılmıştır. Dikkate alınan tekli üst üste çakıştırma alternatifleri ile birlikte toplam 43 adet problem ele alınmıştır. Her bir problem için maksimum

yineleme sayısı 3500000 olarak girilmiştir. Elde edilen sonuçlar aşağıda özetlenmiştir;

1. 2 ve 6 numaralı test problemlerinde olduğu gibi bazı üst üste çakıştırma alternatifleri için optimum çözümler kısa süre içerisinde bulunmuşken, diğer bazı üst üste çakıştırma alternatifleri için girilen yineleme sayısı boyunca ya herhangi bir çözüm bulunamamış ya da sadece olurlu çözüm elde edilebilmiştir. Bu durumda, test problemlerinin özelliklerinin yanı sıra seçilen üst üste çakıştırma alternatiflerinin de modelin analitik çözümünü etkileyebildiği görülmüştür.
2. Kaynak gücünün yüksek olduğu 3 ve 9 numaralı problemlerin bütün üst üste çakıştırma alternatifleri için optimum çözüm bulunmuş, 6 numaralı problemin 3 alternatifi için optimum çözüm, 1 alternatifi için tamsayılı çözüm bulunmuştur. Kaynak gücünün düşük olduğu 1, 4 ve 7 numaralı problemlerin hiçbir alternatifi için çözüm bulunamamıştır. Bu durumda kaynak gücünün problemin analitik çözümü üzerindeki etkisi net bir şekilde görülmüştür.

Elde edilen bu sonuçlar dikkate alınarak, farklı kaynak gücü ve ağ karmaşıklığına sahip büyük çaptaki yeni ürün geliştirme projeleri için makul sürelerde en iyi veya en iyiye yakın şekilde çözüm verecek bir sezgisel algoritmanın önemi anlaşılmaktadır. Bu nedenle, ele alınan problem için PSO temelli bir sezgisel algoritma geliştirilmiştir. Öncelikle, mevcut algoritmanın üst üste çakıştırma içermeyen hali, literatürde kıyaslama amacıyla sıklıkla kullanılan J30, J60, J90 ve J120 problemleri arasından seçilen proje ağları kullanılarak test edilmiştir. Elde edilen sonuçlar PSPLIB’de verilen sonuçlarla kıyaslanmıştır. Yapılan denemeler neticesinde;

1. 30 faaliyetli proje grubundaki problemlerin hepsi için optimum çözümlere çok kısa süreler içerisinde ulaşılmıştır.
2. 60 faaliyetli test problemlerinde, bütün problemler için PSPLIB’deki sonuçların aynısı elde edilmiştir. 100 yinelemenin tamamlanma süresi ise mevcut problemler için 47 saniye ile 64 saniye arasında değişmektedir.
3. 90 faaliyetli proje ağlarında, bütün projelerin bitiş süresi PSPLIB’de yayınlanan sonuçlarla aynı elde edilmiştir.

4. 120 faaliyetli test problemleri için de PSPLIB’de yayınlanan sonuçlarla aynı sonuçlar elde edilmiştir. Gerek faaliyet sayısının artması, gerekse problemi güçleştiren kaynak gücü ve ağ karmaşıklığı faktörlerinin etkisi ile 120 faaliyetli proje ağlarında çözümün bulunduğu yineleme sayısı diğer proje gruplarına göre daha yüksektir. Ancak problemin zorluk derecesi dikkate alındığında bu beklenen bir durumdur. Bu nedenle, 120 faaliyetli proje grubu için, özellikle kaynak gücü ve ağ karmaşıklığının her ikisinin de düşük olduğu problemler için yineleme sayısı 100 den fazla alınabilir ve mevcut problem için program birkaç defa çalıştırılarak daha iyi bir çözüm elde edilip edilemeyeceği kontrol edilebilir.

Elde edilen bu sonuçları özetlemek gerekirse; üst üste karşılaştırma içermeyen PSO algoritması, makul süreler içerisinde, optimum sonucu bilinen 30 faaliyetli proje ağlarının hepsi için optimum çözümü vermiş, diğer proje ağları için ise PSPLIB’de yayınlar sonuçlarla aynı sonucu vermiştir. Bu sonuç iki açıdan önemlidir. İlki, algoritmanın etkinliği çözümü bilinen problemlerle test edilmiş ve ikincisi, üst üste karşılaştırma içermeyen algorithmadan elde edilen temel çizelgeler yardımıyla üst üste karşılaştırma yapılacak faaliyetlerin doğru bir şekilde belirlenmesi sağlanmıştır. Ayrıca, üst üste karşıtırmadan elde edilen kazanç temel çizelge sonuçları ile kıyaslanarak doğru bir şekilde ortaya konulmuştur.

Üst üste karşıtırma içeren PSO algoritması, temel çizelgeler üzerinden yapılan kritik yol analizleri sonucunda belirlenen tekli ve çoklu üst üste karşıtırma alternatifleri için çalıştırılmış ve sonuçlar aşağıda özetlenmiştir;

1. 30 faaliyetli proje ağları için denenen tekli üst üste karşıtırma alternatifleri ile, üst üste karşıtırma olmayan duruma göre proje bitiş sürelerinde %5,26 ile %14,29 arasında değişen oranlarda kısalmalar sağlanmıştır. Mevcut sonuçlara oldukça kısa süreler içerisinde ve genellikle ilk 10 yinelemede ulaşılmıştır. GAMS kodları ile çözülebilen problemlerin sonuçları ile kıyaslandığında, hemen hepsinde optimum çözüme ulaşıldığı görülmüştür. Aynı proje grubu için, çoklu üst üste karşıtırma alternatiflerinin denenmesi ile, karşıtırma olmayan duruma göre proje bitiş süresinde %7,55 ve %25,56 arasında

değişen oranlarda iyileşmeler sağlanmıştır. Çoklu üst üste karşılaştırma sonuçlarına da, çok kısa süreler içerisinde ulaşılmıştır.

2. 60 faaliyetli proje ağları için denen tekli üst üste karşılaştırma alternatifleri ile, %2,78 ile %12,81 arasında değişen oranlarda proje bitiş sürelerinde kısaltmalar sağlanmıştır. Denen çoklu üst üste karşılaştırma alternatifleri ile 16 numaralı problem haricinde diğer bütün problemler için proje bitiş sürelerinde önemli ölçüde düşüşler sağlanmıştır. Mevcut test problemleri için gerçekleştirilen hem tekli hem de çoklu üst üste karşılaştırma sonuçlarına ilk 100 yinelemede ulaşılmıştır.
3. 22 numaralı problem haricinde diğer bütün 90 faaliyetli test problemleri için belirlenen tekli üst üste karşılaştırma alternatifleri ile, %3,53 ile %8,3 arasında değişen oranda iyileşmeler sağlanmıştır. 22 numaralı problem paralel kritik yol içermesi nedeniyle tekli üst üste karşılaştırma ile proje bitiş süresinde herhangi bir kısalma sağlanamamıştır. Ancak, denen çoklu üst üste karşılaştırma alternatifleri ile bu projenin bitiş süresinde %7,92'ye kadar düşüş sağlanabilmiştir. Aynı şekilde, denen çoklu üst üste karşılaştırma alternatifleri ile diğer projeler içinde proje bitiş süresinde önemli düşüşler sağlanmıştır. Diğer proje gruplarında olduğu sonuçlara ilk yinelemelerde ulaşılmıştır.
4. 120 faaliyetli test problemleri içerisinde yer alan 28, 33, 34 ve 35 numaralı test problemlerinin paralel kritik yol içermesi nedeniyle tekli üst üste karşılaştırma ile proje bitiş süresinde herhangi bir kısalma sağlanamamıştır. Aynı gruptaki diğer bütün test problemleri için tekli üst üste karşılaştırma ile proje bitiş süresinde belirli ölçülerde kısaltmalar sağlanırken, denen çoklu üst üste karşılaştırma alternatifleri ile bu oranlarda artış sağlanmıştır.
5. 120 faaliyetli test problemleri içinde hem tekli hem de çoklu üst üste karşılaştırma içeren programda, özellikle kaynak gücü ve ağ karmaşıklığının her ikisinin de düşük olduğu problemlerde yineleme sayısı diğer problemlere göre daha yüksek alınabilir.

Yukarıda özetlenen maddeler dikkate alındığında, çoklu üst üste karşılaştırma içeren programın, farklı üst üste karşılaştırma alternatiflerinin birlikte değerlendirilmesini sağlayarak, proje bitiş süresini daha fazla kısaltabilecek çoklu

çakıştırma alternatiflerinin ortaya konmasını sağladığı görülmüştür. Tekli üst üste çakıştırmanın proje bitiş süresini kısaltmadığı, örneğin temel çizelgenin paralel kritik yol içerdiği durumlarda, çoklu üst üste çakıştırma içeren program proje bitiş süresini kısaltan alternatiflerin belirlenmesini ve belirlenen faaliyetler arasında uygun üst üste çakıştırma miktarını içeren çizelgenin oluşturulmasını sağlaması açısından ayrıca bir öneme sahiptir.

Sonuç olarak, yapılan denemeler neticesinde, farklı seviyelerde proje büyüklüğü, ağ karmaşıklığı, ve kaynak kısıtlılığı içeren proje ağlarında, proje bitiş süresini en küçükleyecek şekilde uygun üst üste çakıştırma miktarını da içeren optimum yada optimuma yakın çizelgeyi makul sürelerde veren bir yöntem sezgisel yöntem geliştirildiği görülmüştür.

Daha sonraki adımda, Wang (2002) makalesinde verilen bir elektronik ürün geliştirme projesi verileri kullanılarak, üst üste çakıştırma sonucunda ortaya çıkan yeniden çalışma miktarını etkileyen ve dolayısıyla üst üste çakıştırmadan elde edilebilecek süre kazancını değiştirebilen gelişim ve duyarlılık parametrelerinin farklı seviyeleri için duyarlılık analizleri yapılmıştır. Yapılan analizler sonucunda, alt seviye faaliyetin duyarlılığı arttıkça yeniden çalışma miktarındaki artışa bağlı olarak üst üste çakıştırmadan elde edilecek kazanç düşebilmektedir. Aynı şekilde, üst seviye faaliyetin gelişim parametresi değerinin artması da yeniden çalışma miktarını artırabilmekte ve süre kazancının düşmesine neden olabilmektedir. Gelişim parametresinin, gelişim fonksiyonu üzerindeki etkisi dikkate alındığında bu beklenen bir durumdur. α parametresi, 1 den büyük bir değer olduğunda bu değer arttıkça yeniden çalışma miktarı da artış yönünde bir eğilim gösterecektir. Mevcut parametre 0 ile 1 arasında olduğunda ise, parametre değeri küçüldükçe yeniden çalışma miktarı azalış yönünde bir eğilim gösterecektir.

5.3. Sonraki Çalışmalar İçin Öneriler

Mevcut çalışmanın devamı olarak aşağıdaki öneriler sıralanabilir;

1. Ele alınan yeni ürünün, buluşlar ve teknolojik gelişmelerden yararlanarak ortaya çıkarılan tamamen yeni ürünler kategorisinde yer alması durumunda,

ürün geliştirme projesinde yer alan faaliyetlerin sürelerinin kesin olarak belirlenmesi mümkün olmayabilir. Bu durumda, faaliyet sürelerindeki belirsizliği dikkate alan bulanık bir sezgisel yöntem geliştirilebilir.

2. Alt seviye faaliyetin duyarlılık parametresi ve üst seviye faaliyetin gelişim parametresi değerlerinin tam olarak belirlenemediği ancak uzman görüşlerinden faydalanılarak tanımlanabileceği durumlar dikkate alınarak bir sezgisel yöntem geliştirilebilir. Bu parametrelerin belirsizliği de bulanık sayılar kullanılarak temsil edilebilir.
3. Geliştirilen model ve sezgisel algoritma, ortak kaynakları kullanan birden fazla ürün geliştirme projesinin birlikte gerçekleştirildiği durumlar için çoklu proje çizelgelemeyi içerecek şekilde genişletilebilir.
4. Geliştirilen PSO algoritması, parçacıkların temsil edilmesi için öncelik temelli gösterimi ve parçacıkların çizelgeye dönüştürülmesi için probleme uyarlanmış seri çizelgeleme prosedürünü kullanmaktadır. Kaynak kısıtlı proje çizelgeme probleminde kullanılan diğer gösterimleri (faaliyet temelli gösterim ve başlama süresi gösterimi) dikkate alan farklı algoritmalar geliştirilerek performansları önerilen algoritma ile kıyaslanabilir.
5. Mevcut problem diğer ileri sezgisel yöntemler kullanılarak da çözülüp sonuçlar kıyaslanabilir.
6. Önerilen program, çizelge sonucu elde edildikten sonra GANNT şemasını hazırlayacak şekilde geliştirilip kritik yol analizi kolaylaştırılabilir. Ayrıca, üst üste çakıştırma yapılan durum için de GANNT şeması oluşturularak üst üste çakıştırmanın görsel olarak gösterilmesi mümkün hale getirilebilir.

KAYNAKLAR

- ABDELSELAM, H.M.E., BAO, H.P., 2006. A simulation- based optimization framework for product development cycle time reduction. IEEE Transactions on Engineering Management, 53(1): 69-85.
- AGARWAL, A., COLAK, S., ERENGUC, S., 2011. A neurogenetic approach for the resource-constrained project scheduling problem. Computers and Operations Research, 38:44-50.
- AKYÜZ.B.,2007. Ürün geliştirme çalışmalarında kullanılan araç ve teknikler; Türk seramik sektöründe bir uygulama. Doktora tezi.
- ALCARAZ, J., MAROTO, C., RUIZ, R., 2003. Solving the multi-mode resource-constrained Project scheduling problem with genetic algorithms. Journal of the Operational Research Society, 54: 614-626.
- BALASUBRAMANIAN, J., GROSSMANN, I.E.,2003. Scheduling optimization under uncertainty- an alternative approach. Computers and Chemical Engineering, 27:469-490.
- BALAS. E., 1971. Project scheduling with resource constraints. Applications of Mathematical Programming Techniques, English University Press, London, 187-200.
- BELL, C.E., PARK, K., 1990. Solving Resource Constrained Project Scheduling Problems by A Search. Naval Research Logistics, 37: 61-84.
- BOCTOR, F.F., 1990. Some efficient multi-heuristic procedures for resource constrained project scheduling. European Journal of Operational Research, 49(1):3-13.
- BOCTOR, F. F.,1996. A new and efficient heuristic for scheduling projects with resource restrictions and multiple execution modes. European Journal of Operational Research, 90 (2): 349-361.
- BOULEIMEN, K., LECOCC, H., 2003. A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode versions. European Journal of Operational Research, 149 (2): 268-281.

- BOWMAN, E.H., 1959. The schedule-sequencing problem. *Operations Research*, 7: 621-625.
- BRUCKER, P., KNUST, S., SCHOO, A., THIELE, O., 1998. A branch and bound algorithm for the resource-constrained project scheduling problem. *European Journal of Operational Research*, 107 (2): 272-288.
- CARRACOSE, M., EPPINGER, S.D., WHITNEY, D.E., 1998. Using the design structure matrix to estimate product development time. *ASME design Engineering Technical Conferences*, Atlanta, GA.
- CARRUTHERS, J.A., BATTERSBY, A., 1966. Advances in critical path methods. *Operational Research Quarterly*, 17: 359-380.
- CHAKRAVARTY, A.K., 2001. Overlapping design and build cycles in product development. *European Journal of Operational Research*, 134(2):392-424.
- CHEN C.H., LING, S.F., CHEN, W., 2003. Project scheduling for collaborative product development using DSM. *International Journal of Project Management*, 21: 291-299.
- CHEN, W., SHI, Y.J., TENG, H.F., LAN, X.B., HU, L.C., 2010. An efficient hybrid algorithm for resource-constrained project scheduling. *Information Sciences*, 180(6):1031-1039.
- CHRISTOFIDES, N., ALVAREZ-VALDES, R., TAMARIT, J.M., 1987. Project Scheduling With Resource Constraints: A Branch-and-Bound Approach. *European Journal of Operational Research*, 29(3): 262-273.
- CHO, H.J., KIM, Y.D., 1997. Simulated annealing algorithm for resource constrained project scheduling problems. *Journal of the Operational Research Society*, 48: 735-744.
- CHO, S.H., EPPINGER, S.D., 2005. A simulation based process model for managing complex design projects. *IEEE Transactions on Engineering Management*, 52(3): 316-328.
- COOPER, D.F., 1976. Heuristics for scheduling resource-constrained projects: An experimental investigation. *Management Science*, 22:1186-1194.

- CZYŻAK, P., JASKIEWICZ, A., 1996. Metaheuristic technique for Solving multiobjective investment planning problem. *Control and Cybernetics* 25: 177–187.
- DAVIS, E., HEIDORN, G.E., 1971. An Algorithm for Optimal Project Scheduling Under Multiple Resource Constraints. *Management Science*, 17(12):B803-B816.
- DAVIS, E., PATTERSON, J.H., 1975. A Comparison of Heuristic and Optimal Solutions in Resource Constrained Project Scheduling. *Management Science*, 21(8): 944-955.
- DEMEULEMEESTER, E. L., HERROELEN, W. S., 1992. A branch and bound procedure for the multiple resource-constrained project scheduling problems”, *Management Science*, 38(12): 1803-1818.
- DEMEULEMEESTER, E. L., HERROELEN, W. S., 1997. New benchmark results for the resource constrained project scheduling problem. *Management Science*, 43(11): 1485-1492.
- DREXL, A., GRÜNEWALD, J., 1993. Nonpreemptive multi-mode resource-constrained project scheduling. *IIE Transactions*, 25(5):74-81.
- DOERSCH, R.H., PATTERSON, J.H.. 1977. Scheduling a Project to Maximize Its Present Value: A Zero-One Programming Approach. *Management Science*, 23(8): 882-889.
- EASTMAN, R.M.,1980. Engineering Information Release prior to final design freeze. *IEEE Transection and Engineering Management*, EM-27: 37-41.
- ELMAGHRABY, S. E.,1993. Resource allocation via dynamic programming in activity networks. *European Journal of Operational Research*, 64(2): 199-215.
- GOLENKO-GINZBURG, D., GONIK, A., 1997. Stochastic network project scheduling with non-consumable limited resources. *International Journal of Production Economics* 48, 29– 37.
- GÜVENÇ, B., 2000. A genetic algorithm for multi-mode resource constrained Project scheduling. Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi Fen Bilimleri Enstitüsü, Ankara.

- HAPKE, M., JASKIEWICZ, A., SLOWINSKI, R., 1994. Fuzzy Project scheduling system for software development. *Fuzzy Sets and Systems* 21, 101–117.
- HAPKE, M., SLOWINSKI, R., 2000. Fuzzy set approach to multiobjective and multi-mode Project scheduling under uncertainty. Slowinski, R., Hapke, M. (Eds.), *Scheduling Under Fuzziness*. Physica-Verlag, Heidelberg, 197–221 (Chapter 9).
- HARTMANN, S., 1998. A competitive genetic algorithm for resource-constrained Project scheduling. *Naval Research Logistics*, 45 (7): 733-750.
- HARTMANN, S., KOLISCH, R., 2000. Experimental evaluation of state of the-art heuristics for the resource constrained project scheduling problem. *European Journal of Operational Research*, 172(2): 394-407.
- HELMS, R., 2004. Framework for realising preliminary information in product development. *Advanced Engineering Informatics*, 18: 231-240.
- HEGAZY, T., 1999. Optimisation of resource allocation and leveling using genetic algorithms. *Journal of Construction Engineering and Management*, 125 (3): 167-175.
- JOZEFOWSKA, J., MIKA, M., ROZYCKI, R., WALIGORA, G., WEGLARZ, J., 1999. Discretecontinuous scheduling to minimize the makespan for power processing rates of jobs. *Discrete Applied Mathematics*, 94 (3): 263-285.
- KLEIN, R., 2000. *Scheduling of Resource – Constrained Projects*. Kluwer Academic Publisher, Boston, 369s.
- KRISHNAN, V., 1996. Managing the Simultaneous Execution of Coupled Phases in Concurrent Product Development. *IEEE Transection on Engineering Management*, 43(2): 210-217.
- KRISHNAN, V., EPPINGER, S.D., WHITNEY, D.E., 1997. A model based framework to overlap product development activities. *Management Science*, 42(4): 437-451.
- KIRKPATRICK, S., GELATT, C. D., 1983. Optimization by simulated annealing. *Management Science*, 220: 671-680.
- LAWSON, M., KARANDIKAR, H.M., 1994. A survey of concurrent engineering. *Concurrent Engineering : Research and Applications*, 2(1): 1-6.

- LEE, J.K., KIM, Y.D., 1996. Search heuristics for resource-constrained Project scheduling. *Journal of the Operational Research Society*, 47 (5): 678-689.
- LEON, V. J., RAMAMOORTHY, B., 1997. An adaptable problem-space search method for flexible flow line scheduling. *IIE Transactions*, 29 (2): 115-125.
- LIN, J., CHAI, K.H., WONG, Y.S., BROMBACHER, A.C., 2008. A dynamic model for managing overlapped iterative product development. *European Journal of Operational Research*, 185: 378-392.
- LOVA, A, TORMOS,P., 2001. Analysis of Scheduling Schemes and Heuristic Rules Performance in Resource-Constrained Multi-project Scheduling. *Annals of Operations Research*, 102: 263-286.
- LOCH, C.H., TERWIESCH, C., 1998. Communication and Uncertainty in Concurrent Engineering. *Management Science*, 44(8):1032-1048.
- LUH, P.B., LIU, F., MOSER, B., 1999. Scheduling of design process with uncertain number of iterations. *European Journal of Operational Research*, 113: 577-592.
- MORI, M., TSENG, C. C., 1997. A genetic algorithm for multi-mode resource constrained project scheduling problem. *European Journal of Operational Research*, 100 (1): 134-141.
- OZDAMAR, L.,1999. A genetic algorithm approach to a general category Project scheduling problem. *IEEE Transactions on Systems Man and Cybernetics Part C Applications and Reviews*, 29 (1): 44-59.
- OZDAMAR, L., ALANYA, E., 2000. Uncertainty modelling in software development projects (with case study). *Annals of Operations Research* 102: 157–178.
- OZDAMAR, L., ULUSOY, G., 1994. A local constraint-based analysis approach to Project scheduling under general resource constraints. *European Journal of Operational Research*, 79 (2): 287-298.
- OZDAMAR, L., ULUSOY, G.,1995. A survey on the resource-constrained project scheduling problem, *IIE Transactions*, 27 (5): 574-586.

- OZDAMAR, L., ULUSOY, G., 1996a. An iterative local constraint based analysis for solving the resource-constrained project scheduling problem. *Journal of Operations Management*, 14: 193-208.
- OZDAMAR, L., ULUSOY, G., 1996b. A note on an iterative forward backward scheduling technique with reference to a procedure by Li and Willis. *European Journal of Operational Research*, 89 (2): 400-407.
- PATTERSON, J.H., HUBER, W.D., 1974. A Horizon Varying Zero-One Approach to Project Scheduling. *Management Science*, 20(6): 990-998.
- PATTERSON, J.H., ROTH, G.W. 1976. Scheduling a Project under Multiple Resource Constraints: A Zero-One Programming Approach. *AIIE Transactions*, 8: 449-455.
- PET-EDWARDS, J., 1996, 1996. A simulation and genetic algorithm approach to stochastic resource-constrained project scheduling. *Southcon Conference Record, IEEE, Piscataway*, 333–338.
- PRITSKER, A.A.B., WATTERS, L.J., WOLFE, P.M., 1969. Multiproject scheduling with limited resources: A zero-one programming approach. *Management Science*, 16: 93-107.
- ROEMER, T.A., AHMADI, R., WANG, R.H., 2000. Time- cost tradeoff in overlapped product development. *Operation Research* 48(6): 858-865.
- SIVRIKAYA-ŞERİFOĞLU, F., 1997. A new uniform order-based crossover operator for genetic algorithm applications to multi-component combinatorial optimization problems. *Doktora Tezi, Boğaziçi Üniversitesi Fen Bilimleri Enstitüsü, İstanbul*, 16-78.
- SIVRIKAYA-ŞERİFOĞLU, F., ULUSOY, G., ŞAHİN, Ş., 2000. Kaynak kısıtlı proje çizelgelemede indirgenmiş nakit akışı maksimizasyonu için bir GA yaklaşımı”, *Endüstri Mühendisliği Dergisi*, 11(2): 2-12 .
- SMITH, P., REINERTSEN, D., 1995. *Developing Product in Half the Time*. 2.Edition, John Walley and Sons, Van Nostrand Reinhold, NY.
- SMITH-DANIELS, D.E., 1987. Optimal project scheduling with materials ordering. *IIE Transactions*, 19:122-129.

- SPRECHER, A., DREXL, A., 1998. Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm. *European Journal of Operational Research*, 107 (2): 431-450.
- STINSON, J.P., DAVIS, E.W., KHUMAWALA, B.W., 1978. Multiple-Resource Constrained Scheduling Using Branch-and-Bound. *AIIE Transactions*, 10(5): 252-259.
- TALBOT, E. B., PATTERSON, J. H., 1978. An efficient integer programming algorithm with network cuts for solving resource-constrained scheduling problems. *Management Science*, 24(11): 1163-1174.
- TORRES, J.R., FRANCO, E.G., MAYORGA, P.M., 2010. Project scheduling with limited resources using a genetic algorithm. *International Journal of Project Management*, 28: 619-628.
- TSAI, Y.W., GEMMIL, D.D., 1996. Using a simulated annealing algorithm to schedule activities of resource-constrained projects, Working Paper No. 96-124. Iowa State University.
- TSAI, Y.W., GEMMIL, D.D., 1998. Using tabu search to Schedule activities of stochastic resource-constrained projects. *European Journal of Operational Research* 111, 129–141.
- ULUSOY, G. ve ÖZDAMAR, L., 1989. Heuristic Performance and Network/Resource Characteristics in Resource Constrained Project Scheduling. *Journal of the Operational Research Society*, 40(12):1145-1152.
- ULUSOY, G., 2002. Proje planlamada kaynak kısıtlı çizelgeleme. Yöneylem araştırması, Erkip N., Köksalan, M., ODTÜ basım işliği, Ankara, 89-127.
- VALLS, V., BALLESTIN, F., QUINTANILLA, S., 2008. A hybrid genetic algorithm for the resource constrained project scheduling problem. *European Journal of Operational Research*, 185(2):495-508.
- WALL, M. B., 1996. A genetic algorithm for resource-constrained scheduling”, Doctor of Philosophy Thesis, Mechanical Engineering Massachusetts Institute of Technology, USA, 29-55.

- WANG, J.R., 1999. A fuzzy set approach to activity scheduling for product development. *Journal of the Operational Research Society* 50: 1217–1228.
- WANG, J., LIN, Y.I., 2009. An overlapping Process model to asses schedule risk for new product development. *Computers and Industrial Engineering*, 57(2): 460-474.
- WANG, J., 2002. A fuzzy project scheduling approach to minimize schedule risk for product development. *Fuzzy Sets and Systems*, 127(2): 99–116.
- _____, 2004. A fuzzy robust scheduling approach for product development projects. *European Journal of Operational Research*, 152: 180–194.
- _____, 2005. Constrained based schedule repair for product development projects with time-limited constraints. *International Journal of Production Economics*, 95: 399-414.
- YILDIZ, A., 2007. Otomotiv sanayiinde hızlandırma tekniklerinin ürün geliştirme zamanı üzerindeki etkileri. Yüksek lisans tezi.
- ZHANG, H., LI, X., LI, H., HUANG, F., 2005. Particle swarm optimization based schemes for resource-constrained project scheduling. *Automation in Construction*, 14: 393-404.
- ZHANG, H., LI, H., TAM, C.M., 2006. Particle swarm optimization for resource-constrained project scheduling. *International Journal of Project Management*, 24: 83-92.
- ZHANG, K., ZHAO, G., JIANG, J., 2009. Particle swarm optimization method for resource-constrained project scheduling problem. *The ninth International Conference on Electronic Measurement and Instruments*.
- ZHUANG, M., YASINSE, A., 2004. Task Scheduling Of Parallel Development Projects Using Genetic Algorithms. *Proceedings of DETC '04 ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference Salt Lake City, Utah USA*.

ÖZGEÇMİŞ

1980 yılında Muş'da doğdu. İlk, orta ve lise öğrenimini Adana'da tamamladı. 1997 yılında başladığı Çukurova Üniversitesi, Endüstri Mühendisliği Bölümü'nden 2001 yılında mezun oldu. 2001 yılında yüksek lisansa, 2002 yılında Araştırma Görevlisi olarak göreve başladı. Halen görevine devam etmektedir. Evli ve bir çocuk annesidir.

EK 1: Üst üste çakıştırmayı içeren seri çizelge oluşturma planı

