

**BAŐKENT ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**ÇOK PROJELİ KAYNAK KISITLI PROJE ÇİZELGELEME
PROBLEMİ: BİR YAZILIM FİRMASINDA UYGULAMA
ÇALIŐMASI**

Pelin AKYIL KURT

YÜKSEK LİSANS TEZİ

2018

**ÇOK PROJELİ KAYNAK KISITLI PROJE ÇİZELGELEME
PROBLEMİ: BİR YAZILIM FİRMASINDA UYGULAMA
ÇALIŞMASI**

**SCHEDULING PROBLEM OF MULTIPLE PROJECTS WITH
LIMITED RESOURCES: APPLICATION IN A SOFTWARE
COMPANY**

Pelin AKYIL KURT

Başkent Üniversitesi
Lisansüstü Eğitim Öğretim ve Sınav Yönetmeliğinin
ENDÜSTRİ Mühendisliği Anabilim Dalı İçin Öngördüğü
YÜKSEK LİSANS TEZİ
olarak hazırlanmıştır.
2018

“Çok Projeli Kaynak Kısıtlı Proje Çizelgeleme Problemi: Bir Yazılım Firmasında Uygulama Çalışması” başlıklı bu çalışma, jürimiz tarafından, 07 / 02 / 2018 tarihinde, **ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI 'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan : Yrd. Doç. Dr. Tusan DERYA

Üye (Danışman) : Yrd. Doç. Dr. Barış KEÇECİ

Üye : Yrd. Doç. Dr. Gerçek BUDAK

ONAY

.... / 02 / 2018

Prof. Dr. Emin AKATA
Fen Bilimleri Enstitüsü Müdürü



BAŞKENT ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
YÜKSEK LİSANS TEZ ÇALIŞMASI ORJİNALLİK RAPORU

Tarih:19 /02 / 2018

Öğrencinin Adı, Soyadı : Pelin AKYIL KURT
Öğrencinin Numarası : 21420109
Anabilim Dalı : Endüstri Mühendisliği
Programı : Endüstri Mühendisliği Tezli Yüksek Lisans
Danışmanın Adı, Soyadı : Yrd. Doç. Dr. Barış KEÇECİ
Tez Başlığı :Çok Projeli Kaynak Kısıtlı Proje Çizelgeleme
Problemi: Bir Yazılım Firmasında Uygulama
Çalışması

Yukarıda başlığı belirtilen Yüksek Lisans tez çalışmamın; Giriş, Ana Bölümler ve Sonuç Bölümünden oluşan, toplam 43 sayfalık kısmına ilişkin, 19 /02 / 2018 tarihinde tez danışmanım tarafından Turnitin adlı intihal tespit programından aşağıda belirtilen filtrelemeler uygulanarak alınmış olan orijinallik raporuna göre, tezimin benzerlik oranı % 12'dir.

Uygulanan filtrelemeler:

1. Kaynakça hariç
2. Alıntılar hariç
3. Beş (5) kelimededen daha az örtüşme içeren metin kısımları hariç

“Başkent Üniversitesi Enstitüleri Tez Çalışması Orijinallik Raporu Alınması ve Kullanılması Usul ve Esaslarını” inceledim ve bu uygulama esaslarında belirtilen azami benzerlik oranlarına tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Öğrenci İmzası

Onay

19 / 02 / 2018

Yrd. Doç. Dr. Barış KEÇECİ

TEŐEKKÜR

Çalıőmanın gerekleőmesi sũresince tezimin çoęunluęunu birlikte alıőtıęım bilgi ve deneyimleri ile bana her zaman destek olan, karőılan gũçlũklerin aőılmasında katkıda bulunan ve alıőmanın sonuca ulaőmasında bũyũk emeęi olan deęerli hocam Yrd. Do. Dr. Barıő KEECCI'ye,

Çalıőma sũresince geliőtirmiő olduęum yazılımin olgunlaőması iin tecrũbelerini benimle paylaőan ve manevi olarak her daim bana destek olan sevgili eőim Onur Deniz KURT'a sonsuz teőekkũrlerimi sunarım.

ÖZ

ÇOK PROJELİ KAYNAK KISITLI PROJE ÇİZELGELEME PROBLEMİ: BİR YAZILIM FİRMASINDA UYGULAMA ÇALIŞMASI

Pelin AKYIL KURT

Başkent Üniversitesi, Fen Bilimleri Enstitüsü

Endüstri Mühendisliği Anabilim Dalı

Proje yönetimi temel olarak projenin hedeflerine ulaşması için planlama ve çizelgeleme faaliyetlerinin birlikte ele alındığı faaliyetler bütünüdür. Proje çizelgeleme problemleri günümüzde proje temelli üretim yapan firmaların ortak problemi olup birtakım kısıtlar altında en kısa zamanda veya en az maliyetle projelerin tamamlanmasını amaçlayan çizelgeleme problemidir. Bu çalışmada proje bazlı çalışan ve aynı anda birden fazla projeyi eş zamanlı yürüten bir yazılım firmasındaki projelerin çizelgelenmesi için bir çözüm yolu önerilmiştir. Firmanın karşı karşıya kaldığı problemde öncüllük ilişkileri olan farklı faaliyetlere ve kısıtlı kaynaklara sahip birden çok projenin en kısa tamamlanma süresinin bulunması amaçlanmaktadır. Problemin çözümü için literatürde yer alan doğrusal karar modelleri araştırılmış ve ele alınan karar modeli optimizasyon programlama dili (OPL) yardımıyla kodlanmıştır. Karar modeli CPLEX çözücüsü ile çözdürülerek sonuçları incelenmiştir. Ayrıca firmanın proje çizelgeleme probleminin karar modeli ile çözümü kolaylaştırmak adına JAVA programlama dilinde bir yazılım geliştirilerek ara yüz tasarlanmıştır. Tasarlanan ara yüz sayesinde firma çalışanlarının önerilen yöntemi daha etkin kullanması sağlanmıştır. Sonuç bölümünde yapılan tez çalışması değerlendirilerek, ileride yapılabilecek çalışmalardan bahsedilmiştir.

ANAHTAR SÖZCÜKLER: Kaynak Kısıtlı Çizelgeleme Problemleri, Çok Projeli Çizelgeleme, Proje Yönetimi, Tam Sayılı Doğrusal Programlama

Danışman: Yrd. Doç. Dr. Barış KEÇECİ, Başkent Üniversitesi, Endüstri Mühendisliği Bölümü.

ABSTRACT

SCHEDULING PROBLEM OF MULTIPLE PROJECTS WITH LIMITED RESOURCES: APPLICATION IN A SOFTWARE COMPANY

Pelin AKYIL KURT

Başkent University, Institute of Science and Engineering

Department of Industrial Engineering

Project management is basically a combination of planning and scheduling activities with the aim to meet project's targets. Nowadays project scheduling is a very common problem in project based production companies. The problem is aim to complete the projects with the shortest time or minimum cost under some constraints. In this thesis, a new method was proposed for generating a schedule for a software company which runs multiple projects simultaneously. The problem faced by the company is aimed at finding the smallest completion time of multi-projects which has different precedence and constrained resources in different activities. In order to solve the problem, the linear decision models were investigated and the chosen decision model was coded using Optimization Programming Language (OPL). The results of the decision model were investigated with the help of CPLEX Optimizer solver. In order to ease the use of the decision model and the solver, a user interface was added using JAVA programming language. As a result, the end users were able to use the proposed scheduling method more affectively. In the conclusion section, the current thesis was evaluated and the possible future studies were called out.

KEYWORDS: Resource Constrained Project Scheduling, Multiproject Scheduling, Project Management, Integer Linear Programming

Advisor: Assistant Professor Barış KEÇECİ, Başkent University, Industrial Engineering Department.

İÇİNDEKİLER LİSTESİ

	<u>Sayfa</u>
ÖZ	i
ABSTRACT	ii
ŞEKİLLER LİSTESİ.....	v
ÇİZELGELER LİSTESİ.....	vi
SİMGELER ve KISALTMALAR LİSTESİ	vii
1. GİRİŞ.....	1
2. LİTERATÜR ARAŞTIRMASI.....	8
2.1 Proje Çizelgeleme Çözüm Yaklaşımları	10
2.1.1 Kesin çözüm veren algoritmalar.....	11
2.1.2 Sezgisel algoritmalar	12
2.2 Kesin Algoritma Çözümleri İçin Geliştirilen Matematiksel Modeller	12
2.2.1 Zaman endeksli değişkenleri kullanan modeller	12
2.2.1.1 KKPÇP kesikli zaman formülasyonu	13
2.2.1.2 KKPÇP ayırık kesikli zaman formülasyonu.....	15
2.2.1.3 KKPÇP akış tabanlı sürekli zaman formülasyonu.....	16
2.2.2 Olay endeksli değişkenleri kullanan formülasyon	18
3. KKPÇP İÇİN DOĞRUSAL KARAR MODELİ.....	20
3.1 Firmaya Ait Genel Bilgi.....	20
3.2 Problemin Tanımı.....	20
3.3. Doğrusal Karar Modeli	22
3.4 Örnek Problem	24
3.4.1 Örnek problemin verileri.....	24
3.4.2 Örnek uygulamanın çözümü	26
4. GERÇEK HAYAT PROBLEMİ	30
4.1 Problemin Verileri	30

	<u>Sayfa</u>
4.2 Problemin Çözümü	34
5. ARAYÜZ GELİŞTİRİLMESİ	38
6. SONUÇ ve ÖNERİLER	42
KAYNAKLAR LİSTESİ	44
EKLER LİSTESİ	46

ŞEKİLLER LİSTESİ

Sayfa

Şekil 1.1 Gantt Şeması.....	3
Şekil 1.2 Aon Faaliyet Ağ Diyagramı	4
Şekil 1.3 Aoa Faaliyet Ağ Diyagramı	4
Şekil 2.1 Proje Çizelgeleme Çözüm Yöntemleri Sınıflandırması.....	11
Şekil 3.1 Birinci Kaynak Tipi 3 Proje Gantt Şeması.....	28
Şekil 3.2 İkinci Kaynak Tipi 3 Proje Gantt Şeması.....	29
Şekil 5.1 Yazılım Sayesinde Tasarlanan Arayüz.....	39
Şekil 5.2 Arayüz Çıktısı.....	39
Şekil 5.3 Veri Giriş Dosyası.....	40

ÇİZELGELER LİSTESİ

	<u>Sayfa</u>
Çizelge 3.1 Birinci Proje Verileri	25
Çizelge 3.2 İkinci Proje Verileri.....	25
Çizelge 3.3 Üçüncü Proje Verileri.....	25
Çizelge 4.1 A Projesinin Verileri	31
Çizelge 4.2 B Projesinin Verileri	312
Çizelge 4.3 C Projesinin Verileri	312
Çizelge 4.4 D Projesinin Verileri	314
Çizelge 4.5 E Projesinin Verileri	314

SİMGELER VE KISALTMALAR LİSTESİ

KKPÇP	Kaynak Kısıtlı Proje Çizelgeleme Problemi
ÇPKKPÇP	Çok Projeli Kaynak Kısıtlı Proje Çizelgeleme Problemi
SS	Başlangıç–Başlangıç
FF	Bitiş–Bitiş
FS	Bitiş–Başlangıç
TA	Teknoloji A
TB	Teknoloji B
OPL	Optimizasyon Programlama Dili
CPM	Critical Path Method (Kritik Yol Yöntemi)
AON	Activity On Node (Düğüm Üzerinde Faaliyet)
AOA	Activity On Arrow (Ok Üzerinde Faaliyet)
PERT	Program Evaluation and Review Technique (Program Değerlendirme ve Gözden Geçirme Tekniği)

1. GİRİŞ

Proje; belirli bir yerde, belirli bir süre içerisinde, belirli bir bütçe ile net olarak tanımlanan amaçların, gerçekleştirilmesine yönelik planlanan faaliyetler bütünüdür. Projeler bir hedefe yöneliktir ve fayda sağlamak, ihtiyacı gidermek ve bir probleme çözüm bulmak gibi amaçları vardır.

Projelerin sonuçlarında ürün veya hizmet çıkmaktadır. Örneğin, bütün sanayi üretim araçları, binalar, bütün mamuller, etrafımızda gördüğümüz her şey çeşitli projelerin ürünleridir.

Proje yönetimi ise proje faaliyetlerinin proje hedeflerine ulaşması için planlanması, çizelgelenmesi, izleme ve kontrolüdür. Bir projenin başarılı olarak tanımlanabilmesi için, projenin planlama aşamasında belirlenen maliyet, takvim ile proje sonunda gerçekleşen maliyet, takvim verilerinin aynı olması beklenmektedir. Proje yönetiminde en kritik noktalardan biri projelerin kapsamı, zamanı ve maliyeti arasındaki dengeyi kurmaktır. Özetle, projenin kalitesinin yüksek olması için müşterinin talebinin belirlenen takvim ve bütçe dâhilinde teslim edilmesi gerekmektedir.

Proje yönetimini kullanan kurumlarda; proje yönetiminin avantajları bulunmaktadır. Sınırlı kaynakların proje faaliyetlerini zamanında gerçekleştirilmesi için dengeli şekilde kullanılmasını, takvimsel aktivitelerin daha önceden belirlenmiş zamanlarda tamamlanabilmesi için gerekli izleme ve kontrol, ölçme ve analiz gibi değerlendirme yöntemlerinin oluşturulmasını sağlamaktadır. Ayrıca proje yönetimi sayesinde proje için risk taşıyan unsurlar takip edilmektedir. Riskler incelenerek soruna dönüşmemesi için yapılması gereken faaliyetler planlanarak, zarar en aza indirgenmeye çalışılmaktadır. Ek olarak, planlama ve gerçekleşme karşılaştırıp performans ölçülmesi sağlanmaktadır. Planlama ve gerçekleşme arasında sapma oluştuğunda soruna yönelik kök neden analizi yapılarak sorunun neden kaynaklandığı tespit edilmektedir. Bu sayede yapılan hataların tekrar yapılması ve başka projelerde aynı hataların tekrarlanması önlenerek projelerdeki verimlilik artmaktadır. Özetle, proje yönetimini kullanıldığında proje daha iyi kontrol edilmekte ve bu sayede müşteri memnuniyetinin arttığı belirtilmiştir. Son olarak,

proje yönetimi sayesinde kalite ve kar artarken geliştirme süresinin ve maliyetlerinin azaldığı gözlemlenmektedir [22].

Proje yönetimi doğru uygulandığında verimliliği artar; çünkü söz konusu iş, kaliteden ödün verilmeden, daha az zamanda, daha az kaynakla gerçekleştirilebilir. Bu nedenle örgüt kültürü, risk toleransı, enflasyon, krizler, kaynak ve yeteneklerin seviyesi, rekabet faktörleri ve teknolojik değişikliklere bağlı olarak proje yönetiminin karlılığı artırdığı kabul edilmektedir. Proje yönetimi sayesinde sonradan gerekebilecek ve maliyeti artıracak kapsam değişiklikleri önlenir. Proje yönetimi işlerin farklı yönlerden yürütülmesini sağladığından örgütün istikrarına da katkı sağlayacaktır. Müşteriye daha yakın bir yönetim şekli olması dolayısıyla müşteri memnuniyetini de artırmaktadır. Proje yönetimi daha kısa sürede problem çözme mümkün kılmaktadır [8].

Proje yönetimi;

- Görevlerin karmaşık olması
- Çevrenin değişken olması
- Zaman, bütçe, insan kaynağı vb. gibi kısıtların olması

Projedeki faaliyetlerin birbirleriyle ilişkilerinin olması ve birçok farklı fonksiyonel birimi içermesi gibi durumlarda tercih edilmektedir.

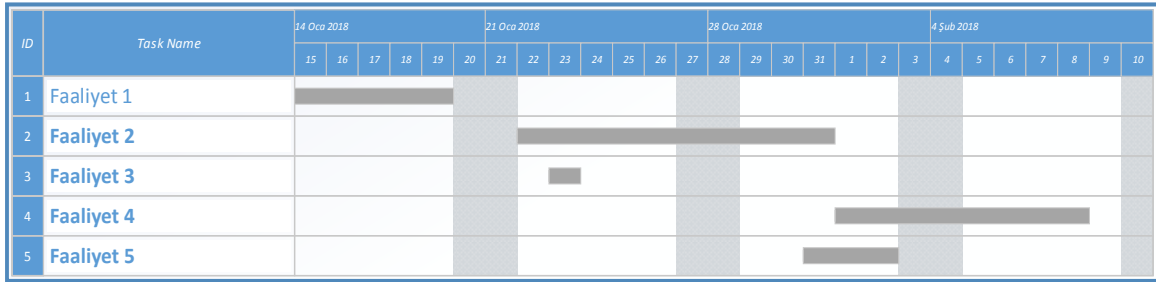
Proje yönetiminde planlama yapılabilmesi için Gantt Şemaları, Program Değerlendirme ve Gözden Geçirme Tekniği (PERT) ve Kritik Yol Metodu (CPM) gibi proje planlama ile ilgili teknikler geliştirilmiştir.

Gantt şemaları, 1915 yılında Henry Gantt tarafından bulunmuş olup dünya tarafından kabul edilmiş olan bir proje yönetim aracıdır. Proje hakkında genel bir fikir verir. Kullanımı kolaydır, sadedir ve kullanıcılar diyagramı kolaylıkla anlayabilir.

Gantt şemaları büyük projelerin yönetimi için tercih edilmemektedir. Bu yöntemde aktiviteler arasındaki bütün bağlantılar gösterilememektedir. Dolayısıyla bir aşamadaki gecikmenin başka aşamaları nasıl etkileyeceği anlaşılamamaktadır. Ek olarak bu yöntemde planlama faaliyetlerinin dışında, sadece faaliyetlerin tamamlanma yüzdeleri takip edilmektedir. Faaliyetlerin zamanında tamamlanması gerektiği ya da her bir faaliyetin proje süresindeki ağırlığı gibi bilgilere

ulaşılamamaktadır. Herhangi bir işin uygulanmasında değişiklik olduğu takdirde bütün şema baştan çizilmesi gerekmektedir [1].

Şekil 1.1 'de örnek bir gantt şeması yer almaktadır. Yatay eksen zamanı gösterirken çubuklar hangi faaliyetin gerçekleştiğini belirtmektedir. Bu durumda yatay eksendeki çubukların uzunlukları da faaliyet süreleri ile doğru orantılıdır.



Şekil 1.1 Gantt Şeması

Critical Path Method(CPM), kritik yol yöntemi olarak dilimizde çevrilmektedir. 1956 yılında İngiltere'de Dupont ve Remington Rand Şirketleri tarafından geliştirilmiş, proje yönetimindeki faaliyetlerin görselliğini sağlayarak grafik ile gösterme ihtiyacı sonucu ortaya çıkmıştır. Ağ modelleri birden fazla faaliyete sahip büyük boyutlu projeleri çizelgelemek için kullanılabilir. Aktivitelerin süreleri daha önceden belirlenmiş ve kesin ise proje süresi, kritik yol yöntemi kullanılarak bulunur. Faaliyetlerin proje süresini uzatmadan ne kadar ertelenebileceğini bulmak için de CPM kullanılabilir.

Program Evaluation and Review Technique (PERT), program değerlendirme ve gözden geçirme tekniği 1958 yılında Amerika Birleşik Devletlerinde Deniz Kuvvetleri tarafından geliştirilmiştir. Faaliyetlerin süreleri belirsiz olduğunda kullanılmaktadır. Bu yöntem ile daha önceden proje için belirlenmiş teslim zamanında bitirme olasılığı hesaplanmaktadır. Program değerlendirme ve gözden geçirme tekniği ile planlama, programlama ve kontrol faaliyetlerinde uygulanmaya başlanmıştır.

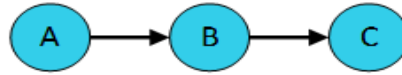
CPM ve PERT'de ortak aşamalar bulunmaktadır. İlk olarak proje ve kritik faaliyetler belirlenir, faaliyetlerdeki öncüllük ardıllık ilişkileri tanımlanır, faaliyet ağ diyagramları çizilir, faaliyetin ne kadar sürede yapılacağı tahmin edilir ve son olarak ağdaki en uzun yol dediğimiz kritik yol hesaplanır. Yapılan hesap sayesinde proje yönetimi daha verimli hale gelmektedir.

Faaliyet Ağ Diyagramları, birçok faaliyet içeren büyük ve karmaşık problemleri çizelgelemek için kullanılır. Projenin hangi faaliyetler ile yapıldığını ve faaliyetler arası sıralamayı gösteren bir ağ yapısıdır. Faaliyetlerin öncüllük ilişkileri de çizilen diyagramlarda net olarak anlaşılmaktadır.

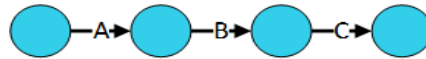
Faaliyet ağ diyagramları faaliyetin ağıdaki düğüm veya ayrıt üzerinde tanımlanmasına bağlı olarak, Activity On Node (AON) düğüm üzerinde faaliyet veya Activity On Arrow (AOA) ayrıt üzerinde faaliyet olmak üzere ikiye ayrılır.

AOA diyagramlarında bağlantılar faaliyetleri gösterirken, düğümler faaliyetlerin başlama ve bitişini gösterir. AON diyagramlarında ise noktalar faaliyetleri gösterirken bağlantılar faaliyetler arası öncelik ilişkilerini gösterir.

AON düğümler ve işlemler ile oluşmaktadır. Şekil 1.2 ' de belirtilen oklar sayesinde öncüllük ilişkileri tanımlanır. İşleme gelen okun öncelikli olarak çıkan ok ise daha sonra yapılacağını gösteren faaliyetlerin olduğunu ifade eder. Şekil 1.2 düğüm üzerinde faaliyeti gösterirken Şekil 1.3 ağ üzerinde faaliyet diyagramını göstermektedir. İki diyagramda da C faaliyetinden önce B, B faaliyetinden önce A geldiği anlaşılmaktadır.



Şekil 1.2 AON Faaliyet Ağ Diyagramı



Şekil 1.3 AOA Faaliyet Ağ Diyagramı

Öncüllük ilişkileri, gereksinimler sonucu ya da yönetimin verdiği bir karar sonucu ortaya çıkabilir. Faaliyetlerin ne zaman başlayıp ne zaman biteceğini öncüllük ilişkileri sayesinde anlaşılabilir. Bir faaliyetin başlaması için ondan önce gelen bütün faaliyetlerin tamamlanmış olması gerekmektedir. Faaliyetler arasındaki ilişkileri belirlemek için Başlangıç–Başlangıç (SS), Bitiş Bitiş (FF) ve Bitiş–Başlangıç (FS) gibi kurallar belirlenmiştir [21].

SS, birinci ve ikinci aktiviteler birlikte başlayabilir. FF, birinci ve ikinci aktiviteler birlikte bitmelidirler. FS ise birinci aktivite bitince ikincisi başlayabilir anlamına gelmektedir.

Proje yönetiminde geliştirilen planlama ve çizelgeleme modelleri, belirli kısıtlar altında oluşturulur ve proje süresi optimize etmeye çalışılır. Kısıtlı kaynak altında proje süresini optimize eden projeler kaynak kısıtlı projeler olarak adlandırılır.

Proje çizelgeleme Tabldot'a göre her işin zamanında bitmesi için yapılan planlamadır. Bu planlama yapılırken faaliyetin başlangıç ve bitiş zamanları değerlendirilip, kaynaklar da göz önünde bulundurulmalıdır [11].

Proje çizelgeleme problemlerinde kaynaklar büyük önem teşkil etmektedir. Materyal, para, insan gücü ve enerji kaynaklara örnek olarak gösterilebilir. Kaynaklar yapıları itibari ile 4 sınıfta incelenmektedir; yenilenebilir kaynaklar, yenilenemez kaynaklar, çift yönden kısıtlı kaynaklar ve kısmen yenilenebilir kaynaklar [12].

Yenilenebilir kaynaklar birim zaman dilimi içerisinde kullanım miktarları kısıtlı olup, o zaman dilimi bittikten sonra tekrar kullanıma hazırdırlar. Kaynaklar belirli bir faaliyet bittikten sonra tekrar kullanılabilir. İş makineleri, iş gücü, enerji ve malzeme yenilenebilir kaynaklara örnek olarak gösterilebilir. Yapılan tez çalışmasında iş gücü olarak yenilenebilir kaynaklar kullanılacaktır.

Yenilenemez kaynaklarda, bütün proje boyunca kullanılabilir fakat kaynakların toplam miktarı kısıtlıdır. Bir inşaat projesinde kullanılacak çimento ve demir örnek olarak gösterilebilir.

Çift yönden kısıtlı kaynaklarda, bir kaynağın hem birim zamanda kullanılan miktarı, hem de projede toplam tüketimi kısıtlıdır. Para örnek olarak verilebilir. Proje süresi boyunca birim zamanda kullanımı sınırlıdır [10].

Kısmen yenilenebilir kaynaklarda ise bazı kaynakların kullanım miktarları sınırlıdır. Örneğin, proje planı çerçevesinde işçilerle haftalık olarak belirli bir çalışma saati karşılığında kontrat yapılması verilebilir. Bu örnekte ele alınan kaynak hem yenilenebilir hem de yenilenemez kaynaklara örnek teşkil etmektedir. [13]

Ek olarak kaynaklar ayrık ve sürekli olmak üzere iki sınıfa ayrılabilir. Ayrık kaynaklar grubunu makine sayısı, kablo sayısı gibi sayılabilir kaynaklar oluştururken; sürekli kaynak grubunu ise elektrik enerjisi gibi bölünemeyen kaynaklar oluşturmaktadır [11].

Sonuç olarak Kaynak Kısıtlı Proje Çizelgeleme Problemi (KKPÇP), kısıtlı kaynaklar kullanılarak bir projeyi oluşturan faaliyetlerin, öncüllük ilişkilerini de dikkate alarak amaç fonksiyonunu optimum yapmak için çizelgelenmesidir. KKPÇP karar vericinin hedeflerine yönelik çıkabilecek birçok amaç fonksiyonu olabilir. Fakat literatürde en çok karşılan, proje süresini en küçükmektir.

Bu çalışmada proje bazlı çalışan ve aynı anda birden fazla projeyi eş zamanlı yürüten bir yazılım firmasındaki projelerin çizelgelenmesi için bir çözüm yolu önerilmiştir. Firmanın karşı karşıya kaldığı problemde öncüllük ilişkileri olan farklı faaliyetlere ve kısıtlı kaynaklara sahip birden çok projenin en kısa tamamlanma süresinin bulunması amaçlanmaktadır.

Çalışma özel bir firmada birden çok projenin en kısa tamamlanma süresi bulunamaması üzerinde hayata geçirilmiş bir uygulama çalışmasıdır. Tez çalışması toplam altı bölümden oluşmaktadır.

İlk bölümde, proje yönetiminin tanımı, avantajları ve proje çizelgeleme yöntemlerinden bahsedilerek çalışmanın amacı ve kapsamı hakkında bilgi verilmiştir.

İkinci bölümde, kaynak kısıtlı proje çizelgeleme probleminin sınıfları anlatılmış, problemin çözümü için literatürde yer alan çözüm yaklaşımları özetlenerek doğrusal karar modelleri incelenmiştir.

Üçüncü bölümde, çalışmanın yapıldığı firmaya ait genel bilgiler anlatılmış ve problemin tanımına detaylı olarak yer verilmiştir. Problemin çözümü için kullanılacak doğrusal karar modeli belirtilmiştir. Anlatılan gerçek hayat probleminin boyutu büyük olduğundan çözüme daha kolay ulaşmak adına öncelikle daha küçük boyutlu örnek veriler kullanan bir uygulama yapılmıştır. Örnek uygulamanın CPLEX çözümü incelenerek, Gantt şemaları çizilmiştir.

Dördüncü bölümde, gerçek hayat probleminin verileri yer almakta olup, projelerin faaliyet ağ diyagramları çizilmiş ve kurulan model OPL Programında kodlanmıştır. Model çalıştırılarak sonuçları incelenip analizler yapılmıştır.

Beşinci bölümde, gerçek hayat problemine bir çözüm yolu olarak sunulan modelin kullanımını kolaylaştırmak adına, JAVA uygulaması ile geliştirilen yazılım sayesinde tasarlanan ara yüz hakkında detaylı bilgi verilmiştir.

Altıncı bölümde ise, çalışmanın sonuçları verilmiş ve yapılan çalışmanın eksikleri ile birlikte nasıl genişletilebileceğine dair öneriler getirilmiştir.

2. LİTERATÜR ARAŞTIRMASI

Bu bölümde, kaynak kısıtlı proje çizelgeleme problemlerinin (KKPÇP) literatürdeki sınıflandırılması anlatılmış ve kesin çözüm veren algoritmalar ile sezgisel çözüm veren algoritmalar olmak üzere proje çizelgeleme çözüm yaklaşımlarından bahsedilmiştir. Kesin algoritma çözümleri için geliştirilen matematiksel modellerin literatürdeki indeksleri, parametreleri, karar değişkenleri ve amaç fonksiyonları ile birlikte doğrusal karar modelleri açıklanmıştır. Çalışma yapılan firmada, kesin sonuçlara ihtiyaç olduğundan sezgisel algoritma çözümleri için geliştirilen matematiksel modellerin detayına değinilmemiştir.

Kaynak kısıtlı proje çizelgeleme problemleri altı farklı sınıfta ele alınmaktadır.

- Tek modlu kaynak kısıtlı proje çizelgeleme problemleri
- Çok modlu kaynak kısıtlı proje çizelgeleme problemleri
- Düzensiz amaç fonksiyonuna sahip kaynak kısıtlı proje çizelgeleme problemleri
- Stokastik yapıllı proje çizelgeleme problemleri
- Çok projeli Kaynak kısıtlı proje çizelgeleme problemi (ÇPKKPÇP)
- Amacı birden fazla olan KKPÇP

Tek modlu kaynak kısıtlı proje çizelgeleme problemleri arasında en basiti olup amaç tek bir projenin tamamlanma süresini en kısa yapmaktır. Kaynak kullanımı tek modludur. 1977 yılında Elmaghraby çok modlu kaynak kısıtlı proje çizelgeleme modelleri arasında çalışmıştır. Tek modlu KKPÇP 'ine göre bu problemdeki fark kaynak kullanımında birden fazla modun olmasıdır [24].

Örneğin iki modlu bir problemde, bir işçinin bir makineyi altı günde tamir etmesi mod 1 iken, 2 işçinin bir makineyi 3 günde tamir etmesi mod 2 olarak tanımlanabilir. Görüldüğü gibi toplam 6 günde bitirilebilen bir iş ek kaynak atamaları yapılarak daha az sürede gerçekleşmektedir. Bahsedilen kaynaklar yenilenebilir ya da yenilemeyen tipte olabilir.

Düzensiz amaç fonksiyonuna sahip kaynak kısıtlı proje çizelgeleme problemlerinde asıl amaç genel olarak projenin finansal durumunun optimize edilmesidir [3].

Stokastik yapılı proje çizelgeleme problemlerinde, amaç fonksiyonu proje süresinin kısaltılması yerine projenin beklenen bitirme süresinin en küçüklenmesidir. Problemden faaliyet süreleri kesin değerler olmayıp, belirli olasılık dağılımları sonucu rassal değerler almaktadır.

Çok projeli kaynak kısıtlı proje çizelgeleme problemleri (ÇPKKPÇP) gerçek hayattaki problemleri de yansıttığından literatürde konu hakkında birçok araştırma bulunmaktadır. Bir şirkette birden fazla proje aynı anda yürütülmesi gerekiyor ve kısıtlı kaynaklar ortak kullanılıyorsa, kaynak kısıtlı proje çizelgeleme problemlerine örnek olarak gösterilebilir. Genellikle proje yönetiminde, ortak bir kaynak havuzundan beslenilerek birden fazla proje yönetilmektedir. Tez çalışmasında da ÇPKKPÇP üzerinde çalışılacaktır.

ÇPKKPÇP için iki doğrusal karar modeli yaklaşımı olmasına rağmen amaç fonksiyonları aynıdır. İlk yöntemde projelerin başlangıç ve bitiş zamanları vardır ve ayrı projeler olarak ele alınır. İkinci yöntemde ise sistemde bir çok proje olmasına rağmen bütün projeler tek bir proje olarak ele alınır [24].

Tsubakitani ve Deckro (1990), ÇPKKPÇP üzerinde çalışmış ve yüzden fazla faaliyet içeren en az elli projeli inşaat firması için dinamik bir algoritma önermişlerdir. Problemin çözümü için sezgisel algoritmalar kullanmışlardır [23].

Bouleimen ve Lecocq (1999), sezgisel yöntemleri kullanarak ÇPKKPÇP üzerinde çalışmışlardır. Problemin çözümü için kesin çözüm yöntemleri yerine sezgisel çözüm yöntemlerine başvurmuşlardır. Bu çalışmada bütün projelerin sıfıncı zaman diliminde başlamasına gerek yoktur. Farklı zamanlardada başlayabilmektedir [4].

Lawrence ve Morton (1993), projelerin gecikmelerinden doğacak maliyetlerin birbirinden farklı olduğunu ve dolayısıyla firmalara verilen zararın farklı olduğunu düşünerek ağırlıklı gecikme maliyetleri hesaplamışlardır. Oluşan maliyetlere göre de kaynakları fiyatlandırmışlardır [14].

Lova ve Tormos (2000), projeleri tek tek değerlendirerek iki aşamalı sezgisel yöntem kullanmışlardır. İki aşamalı sezgisel yöntemlerden tek aşamalı yöntemlere göre daha iyi sonuçlar alınmaktadır [15].

Literatürdeki araştırmalara göre ÇPKKPÇ probleminde amaç fonksiyonu aynı anda birden çok durumu optimize etmeye çalışan modeller olabilir. Bu problemler çok amaçlı kaynak kısıtlı proje çizelgeleme problemi olarak adlandırılır. Örneğin, maliyeti azaltıp karı arttırmak ya da toplam proje süresinin en küçüklenmesine yönelik amaçlar bulunabilir.

1981 ve 1989 yıllarında Slowinski projeleri en kısa süre ve en az maliyet ile bitirmeyi amaçlayan iki amaçlı kaynak kısıtlı proje çizelgeleme problemi üzerinde çalışmışlardır[19,20].

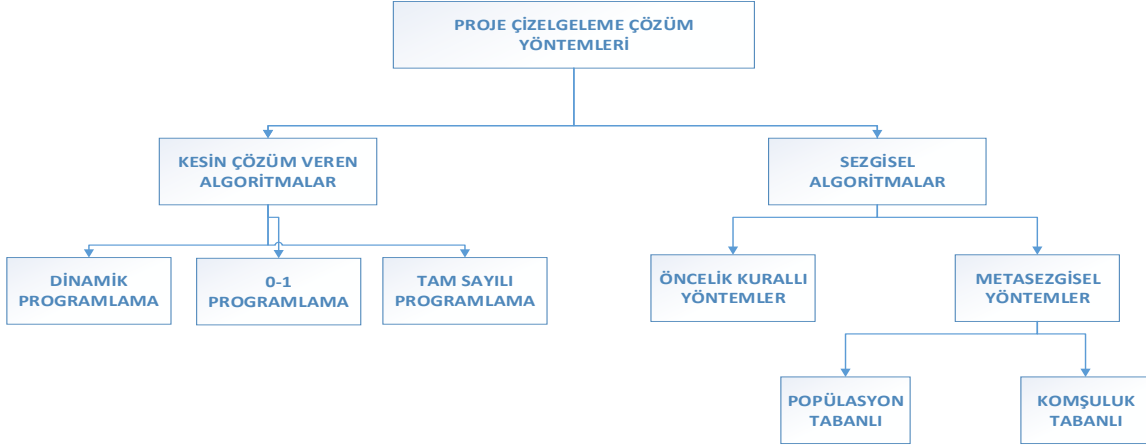
Dean (1992), ÇPKKPÇP üzerinde çalışmıştır. Amaç fonksiyonu en kısa proje süresi ile en az kaynak kullanım miktarını bulmaktır. Kaynak kullanım miktarları birim zamanda sabit değildir. Sezgisel algoritmalar kullanarak problemin çözümüne ulaşmıştır. Yapılan çalışma personel çizelgeleme için uygulanmıştır [6].

Browning ve Yassine çalışmasında, KKÇPÇP'ni 20 öncelik kuralı ve paralel çizelgeleme yapısı kullanarak incelemiştir. Toplam tamamlanma süresi en aza indirilmeye çalışılırken aynı zamanda her bir proje içinde gecikme süresi azaltılmaya çalışılmaktadır [5].

2.1 Proje Çizelgeleme Çözüm Yaklaşımları

Literatürde, proje çizelgeleme problemi çözüm yollarını, “kesin metotlar” ve “sezgisel metotlar” olmak üzere iki ana sınıfa ayırmışlardır. Amaç fonksiyonu proje süresini en küçüklemek olan kaynak kısıtlı proje çizelgeleme problemi NP-zor bir problemdir ve kesin çözüm yöntemleri ancak belirli problem büyüklükleri için geçerlidir. Kesin çözüm yöntemlerine alternatif olarak sezgisel metotlar geliştirilmiştir.

Şekil 2.1 'de görüldüğü gibi kesin çözüm veren algoritmalar; dinamik programlama, 0-1 programlama ve tam sayılı programlama olmak üzere sınıflandırılırken, sezgisel algoritmalar ise öncelik kurallı yöntemler ve meta sezgisel yöntemler olmak üzere 2 sınıfa ayrılır.



Şekil 2.1 Proje Çizelgeleme Çözüm Yöntemleri Sınıflandırması

2.1.1 Kesin çözüm veren algoritmalar

Kesin çözüm veren algoritmalar çözümleri dal-sınır algoritması, 0-1 programlama, dinamik programlama şeklinde sınıflandırılmıştır [24].

Kesin çözüm yöntemlerinden olan KKPÇP çözümü için Pritsker 0-1 doğrusal programlama formülasyonu önermiştir. Önerilen modelde projeler için, toplam üretim zamanını, toplam tamamlanma zamanını veya toplam gecikme zamanını en küçükleme gibi üç değişik amaç fonksiyonu dikkate alınmıştır[17].

Reyck ve Herroelen, genelleştirilmiş öncelik ilişkili kaynak kısıtlı proje çizelgeleme problemi için bir dal-sınır algoritması sunmuş, projenin tamamlanma süresini en küçüklemeyi amaçlamışlardır. Algoritmanın arama ağacındaki düğümler, kendisinden önceki aşamada oluşmuş bir kaynak çatışmasını çözen ekstra öncelik ilişkileriyle genişletilmiş proje ağını temsil eder. Arama ağacının geniş parçalarını aramak için, öncelik ve kaynak bazlı alt sınırlar kadar üstünlük kurallarını da kullanmışlardır [18].

Demeulemeester ve Herroelen, daha önceki çalışmalarını güncelleyerek, 32-bit programlamanın avantajlarını kullanmış, adreslenebilir bilgisayar hafızası, arama stratejisi ve geliştirilmiş bir alt sınırın hesaplama zamanı üzerindeki etkileri üzerine çalışmışlardır [7].

Dorndorf ve diğerleri, zaman odaklı bir dal-sınır algoritması öne sürmüşler ve arama uzayını küçültmek için geçici kısıtlar ve kaynak kısıtlarından yararlanan,

kaynak yayma tekniđi kullanmıřlardır. Test problemleri üzerinde önerdikleri yöntemi kanıtlamıřlardır [9].

2.1.2 Sezgisel algoritmalar

Sezgisel algoritma yaklařımında bekleme zamanı en çok olan faaliyetlere bakılır ve bu faaliyetler tüm projenin bitiř zamanını geciktirmeyecek řekilde ertelemeye çalıřılır. KKPÇP'nin ierdiđi deđiřken sayısının artması çözümler zamanını artırmaktadır. Çözüm zamanındaki artış problem olmaktadır. Bu nedenle, karar vericiler sıklıkla, optimale yakın, iyi sonuçlar üretebilen ve hızlı tekniklerle ilgilenmeye bařlamıřtır. Bu tekniklere sezgiseller adı verilir [16].

Sezgisel algoritmalar řekil 2.1 'de belirtildiđi gibi öncelik kurallı yöntemler ve meta sezgisel yöntemler olarak sınıflandırılmıř ve meta sezgisel yöntemler ise popülasyon tabanlı ve komřuluk tabanlı olmak üzere kabaca ikiye ayrılmıřtır.

Tez kapsamında problemin çözümleri için kesin algoritmalar ile çözümler yolu arandıđından, sezgisel algoritmaların detayına deđinilmemiřtir.

2.2 Kesin Algoritma Çözümleri İin Geliřtirilen Matematiksel Modeller

KKPÇP'nin kesin çözümlerinin bulunabilmesi için geliřtirilen birok matematiksel model literatürde incelenmiřtir. Arařtırmalar sonucu tez kapsamında ele alınan problem için uygulanacak uygun bir model seilmiř ve yazılım firmasındaki problem için uygulanarak sonuçları incelenmiřtir.

Literatürde tam sayılı dođrusal programlamada farklı özelliklerde(yapılarda) modeller üretilmiřtir. Öncelikle zaman endeksli deđiřkenleri kullanan modeller anlatılmıř devamında zamana dayanmak yerine, olay endeksli deđiřkenlere (Bařla/ Bitir Formülasyonu ve A Kapa Formülasyonu) dayalı kaynak kısıtlı proje çizelgeleme tam sayılı dođrusal programlama formülasyonlarından bahsedilmiřtir.

2.2.1 Zaman endeksli deđiřkenleri kullanan modeller

Bu bařlık altında öncelikle Pritsker'in geliřtirmiř olduđu kesikli zamanlı formülasyonun varsayımları, ama fonksiyonu ve kısıtları anlatıldıktan sonra Christofides'in geliřtirdiđi "Disaggregated Discrete-time" formülasyonu yani ayrıık

kesikli zaman formülasyonu olarak dilimizde çevrilen model anlatılmıştır. Son olarak da Artigues 'in geliştirmiş olduğu “ Flow-based continuous-time” olarak tanımlanan akış tabanlı sürekli zaman formülasyonu modelinden bahsedilmiştir [13].

Literatürdeki doğrusal karar modelleri incelendiğinde çalışma yapılan firmada kullanılmak üzere Pritsker'in geliştirmiş olduğu ayrık zamanlı (discrete time formulation) tam sayı lineer programlama modeli kullanılarak küçük uyarlamalar yapılmıştır. Pritsker'in modelinin tercih edilmesinin nedeni modelin, literatürdeki modellere kıyasla daha anlaşılır olması ve amaç fonksiyonun firmadaki problem ile aynı olmasıdır. Ayrıca firmadaki problem için kesin çözüme ulaşmak istenmektedir. Bu nedenle optimala yakın tahmini sonuç veren sezgisel algoritmalar kullanmak yerine kesin çözüm veren algoritmaları kullanarak sonuca gidilmesi tez çalışmasındaki problem için daha anlamlıdır.

2.2.1.1 KKPÇP kesikli zaman formülasyonu

Pritsker 1969 yılında kaynak kısıtlı proje çizelgeleme problemin çözümü için, 0-1 doğrusal programlama formülasyonu önermiştir [2]. Pritsker modelinde zaman ve aktiviteden oluşan ikili karar değişkeni kullanmıştır. Bu nedenle Pritsker'in geliştirmiş olduğu model kesikli zaman formülasyonu olarak adlandırılmıştır.

x_{jt} : j aktivitesi t zamanında başlar ise 1, başlamaz ise 0 değerini alan ikili karar değişkeni olarak tanımlanır.

Pritsker'in geliştirmiş olduğu matematiksel modelinin varsayımları, amaç fonksiyonu ve kısıtları aşağıda belirtilmiştir.

Varsayımlar

- Belirli bir faaliyet kümesinden oluşan tek bir proje vardır.
- Bir aktivitenin öncül faaliyetleri bitmeden sıradaki aktivite başlayamaz.
- Başlatılan faaliyetler kesintisiz bitirmek zorundadır, ara verilemez.
- Sınırlı sayıda kaynak bulunmaktadır.
- Kaynak kullanılabilirliği ve kaynak tüketimi, çizelgeleme boyunca sabittir.
- Kaynaklar birbirinin yerine geçemez.

- Projenin bitiş zamanı kritik faaliyetlerin süreleri ile belirlenir.
- Yapay faaliyetler olan 0 ve $n + 1$ sırayla, “proje başlangıcına” ve “proje bitimine” karşılık gelmektedir [2].

İndeksler

j : faaliyet indeksi ; $j = 1, 2, \dots, N$; N = Projedeki faaliyetlerin sayısı

k : kaynak indeksi ; $k = 1, 2, \dots, K$; K = Kaynak tiplerinin sayısı

t : zaman periyodu ; $t = 1, 2, \dots, T$; T =Projenin tamamlanma zamanındaki üst sınır

Problem Parametreleri

d_j = j faaliyetinin süresi

r_{jk} = j faaliyetini gerçekleştirmek için kullanılan k kaynağı sayısı

R_{kt} = t zamanında uygun olan k kaynağı sayısı

l_j = j faaliyetinin en erken tamamlanma zaman periyodu

u_j = j faaliyetinin en geç tamamlanma zaman periyodu

H_j : { öncül aktivitelerin kümesi}

l_j ve u_j CPM yönetimi kullanılarak hesaplanmaktadır.

Karar Değişkenleri

$$x_{jt} = \begin{cases} 1, & t \text{ zamanında } j \text{ faaliyeti tamamlanırsa} \\ 0, & t \text{ zamanında } j \text{ faaliyeti tamamlanmaz ise} \end{cases}$$

x_{jt} ; $u_j < t < l_j$ periyodları arasında değişken olarak ele alınmış olup tanımlıdır.

Doğrusal Karar Modeli

Matematiksel model amaç fonksiyonu ve kısıtlar olmak üzere detaylı olarak anlatılmıştır.

$$\sum_{t=l_j}^{u_j} x_{jt} = 1 ; j=1, 2, \dots, N \quad (2.1)$$

$$\sum_{t=l_m}^{u_m} tx_{mt} + d_n \leq \sum_{t=l_n}^{u_n} tx_{nt} ; \forall m \in H_n \quad (2.2)$$

$$\sum_{q=t}^{t+d_j-1} r_{jk} x_{jq} \leq R_{kt} ; t=1, 2, \dots, T \quad k=1, 2, \dots, K \quad (2.3)$$

kısıtları altında

$$\text{enk } z = \sum_{j=1}^N \sum_{t=l_j}^{u_j} tx_{jt} \quad (2.4)$$

2.4’de amaç fonksiyonunu verilmiştir. Verilen amaç fonksiyonu aktivitelerin toplam tamamlanma süresini en küçüklemektedir.

2.1 eşitsizlik kısıtını oluşturmaktadır. Her bir faaliyetin belirli bir zaman diliminde gerçekleşeceğini belirtir.

2.2 öncüllük kısıtını anlatmaktadır. m aktivitesi n aktivitesinin öncülü olduğu varsayılın ve t_m , t_n sırasıyla m ve n aktivitelerinin tamamlanma zamanı olsun.

Bu durumda;

$t_m + d_n \leq t_n$ $t_m = \sum_{t=l_m}^{u_m} tx_{mt}$ ve $t_n = \sum_{t=l_n}^{u_n} tx_{nt}$ olduğuna göre öncüllük kısıtı 2.2 ‘deki formülasyon olarak tanımlanır.

2.3 kaynak kapasite kısıtını tanımlamaktadır. Herhangi bir periyotta toplam kullanılan k kaynak sayısı uygun olan k kaynak sayısını geçemez. t zamanında faaliyet gösteren herhangi bir j aktivitesi q periyodunda tamamlanıyorsa;

$t \leq q \leq t + d_j - 1$ olduğundan kaynak kısıtı aşağıdaki şekilde tanımlanabilir. l_j ve u_j değerleri tam sayı olduğunda bu kısıt çalışır.

2.2.1.2 KKPÇP ayrık kesikli zaman formülasyonu

Christofides’in 1987 yılında geliştirmiş olduğu ayrık kesikli zaman (disaggregated discrete-time) formülasyonunu yani dilimizde ayrık kesikli zaman formülasyonu olarak çevrilen model Pritsker’in geliştirmiş olduğu modele çok benzemektedir [14].

Tek farkları öncüllük kısıtının formülasyonundan kaynaklanmaktadır. Christofides öncüllük kısıtı 2.5 numaralı formülasyonda tanımlamıştır [13].

$$\sum_{T=t}^T x_{iT} + \sum_{T=0}^{t+p_i-1} x_{jT} \leq 1 \quad , \forall t \in T, \forall (i,j) \in N \quad (2.5)$$

2.5 numaralı formülasyonda kullanılan 0,1 karar değişkenlerinin sayısı çizelgeleme ufku olan T ile orantılı olarak artar. T ne kadar fazla olursa 0-1 karar değişkeni sayısı o kadar fazla olur, dolayısıyla problemi çözmek zorlaşır.

2.2.1.3 KKPÇP akış tabanlı sürekli zaman formülasyonu

Son olarak Artigues 'ın geliştirmiş olduğu akış tabanlı sürekli zaman (flow-based continuous-time) formülasyonundan bahsedilmiştir [14].

Karar Değişkenleri

Artigues geliştirmiş olduğu model de sürekli değişken, sıralı değişken ve süreli akış değişkeni olmak üzere 3 çeşit değişken bulunmaktadır. Sürekli değişken olan S_i , i aktivitesinin başlangıç zamanını tanımlarken; x_{ij} sıralı değişkeni, i faaliyeti bittikten sonra j faaliyeti başladığını belirtir. Son olarak süreli akış değişken olan f_{ijk} i aktivitesi bitip j aktivitesi başlarken aktarılan k kaynak sayısını göstermektedir[14].

Parametreler

V : Aktiviteler Kümesi

p :Aktivite Süreleri

n : Aktivitelerin Sayısı

m : Uygun Kaynakların Sayısı

E : Öncüllük İlişkileri Seti

R : Yenilenebilir Kaynak Sayısı

b : Kaynak Tüketim Miktarı

H : Zamanların Toplamı

B : Uygun Kaynak Vektörü

İndeksler

$i,j = \{ 0, \dots, n+1 \}$ aktivite indeks;

$R = \{ 1, \dots, m \}$ kaynak indeks;

$H = \{ 0, 1, \dots, T \}$ zaman periyodu indeks;

Doğrusal Karar Modeli

Matematiksel model amaç fonksiyonu ve kısıtlar olmak üzere aşağıda detaylı olarak anlatılmıştır.

$$x_{ij} + x_{ji} \leq 1 \quad \forall (i,j) \in (A \cup \{0, n+1\})^2, i < j \quad (2.6)$$

$$x_{ik} \geq x_{ij} + x_{jk} - 1 \quad \forall (i,j,k) \in (A \cup \{0, n+1\})^3 \quad (2.7)$$

$$S_j - S_i \geq -M + (p_i + M) x_{ij} \quad \forall (i,j,k) \in (A \cup \{0, n+1\})^2 \quad (2.8)$$

$$f_{ijk} \leq \min(b_{ik}, b_{jk}) x_{ij} \quad \forall (i,j) \in (A \cup \{0\}) \times A \cup \{n+1\}, \forall k \in R \quad (2.9)$$

$$\sum_{j \in A \cup \{0, n+1\}}^T f_{ijk} = \widetilde{b}_{ik} \quad \forall (i) \in (A \cup \{0, n+1\}), \forall k \in R \quad (2.10)$$

$$\sum_{i \in A \cup \{0, n+1\}}^T f_{ijk} = \widetilde{b}_{jk} \quad \forall (i) \in (A \cup \{0, n+1\}), \forall k \in R \quad (2.11)$$

$$x_{ij} = 1 \quad \forall (i,j) \in E \quad (2.12)$$

$$f_{ijk} \geq 0 \quad \forall (i,j) \in (A \cup \{0, n+1\})^2, \forall k \in R \quad (2.13)$$

$$f_{(n+1)0k} = B_k \quad \forall k \in R \quad (2.14)$$

$$S_i \geq 0 \quad \forall i \in (A \cup \{0, n+1\}), \quad (2.15)$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in (A \cup \{0, n+1\})^2 \quad (2.16)$$

kısıtları altında

$$\text{enk } S_{n+1} \quad (2.17)$$

2.17' de belirtilen amaç fonksiyonu son faaliyetin başlangıç zamanını en küçüklemektedir. 2.6. kısıt öncüllük kısıtını anlatmaktadır. i faaliyeti j faaliyetinin yada j faaliyeti i faaliyetinin öncülü olabilirken, bu iki faaliyet paralel de yürütülebilir.

2.7.kısıt öncüllük ilişkileri arasındaki geçişi ifade eder. Bu kısıt faaliyetler arasında öncüllük ilişkilerinin gerçekleşmesini sağlamaktadır. 2.8.kısıt i ve j 'nin başlangıç zamanlarını bağlayan kısıttır. Bu kısıt $x_{ij} = 1$ iken aktif olur ve bu durumda $S_j \geq S_i + p_i$ öncüllük kısıtını uygular.

2.9. kısıt x_{ij} değişkenlerini birbirine bağlar. İ faaliyeti j faaliyetinin öncülü olduğu bir durumda i'den j'ye gönderilen maksimum akış en az $\{b_i, b_j\}$ olarak ayarlanırken i faaliyeti j'den önce gelmezse akış sıfır olmalıdır.

2.10. ve 2.11. kısıt kaynak akış korunum kısıtını ifade ederken, 2.12. kısıt akış korunum kısıtı 2.13. Kısıt ise öncelik kısıtlamalarını anlatmaktadır. 2.14. kısıt kaynak tüketimi ile ilgilidir.

2.15. kısıt i aktivitesinin başlangıç zamanının 0'dan büyük olduğunu ifade ederken son 2.16. kısıt ise x_{ij} değişkeninin 0 ya da 1 değeri aldığını ifade eder.

2.2.2 Olay endeksli değişkenleri kullanan formülasyon

Zaman endeksli değişkenleri kullanan formülasyonların aksine, burada, olaylarla endekslenmiş değişkenleri kullanan iki yeni formülasyon vardır. Bu formülasyonlar Başla/Bitir olay tabanlı formülasyonu ve Açık/Kapalı olay tabanlı formülasyonlarıdır.

Grossmann'ın çalışmış olduğu küme süreç problemleri ve Dauter-P'eres ve Lasserre'nin atölye tipi çizelgeleme formülasyonlarından esinlenilmiştir [14].

Bir aktivite başladığında veya bittiğinde bir olayın gerçekleştiğini düşünülmektedir. Kaynak kısıtlı proje çizelgeleme probleminde herhangi bir sola kaydırılmış çizelge için, bir aktivitenin başlangıç zamanı 0 veya başka bir aktivitenin bitiş zamanı ile çakışmaktadır.

Sonuç olarak, olayların sayısı, faaliyetlerin sayısının bir fazlası ile sınırlıdır.

E olayların kümesi n de faaliyet sayısı olduğunu varsayılırsa;
 $E = \{0, 1, n\}$

Olay temelli formülasyonlar, yapay faaliyet içermezler. Sonuç olarak, aktivite sayısı $n + 2$ yerine tüm önceki formülasyonlar için n 'dir. Aktivite süreleri tam sayı

olmadığında olay temelli formülasyonları kullanmak avantaj sağlamaktadır. Daha da önemlisi, uzun çizelgeleme ufuklarında olay temelli formülasyonlar zaman tabanlı formülasyonlara kıyasla daha az değişken içerir.

Daha öncede bahsedildiği gibi olay tabanlı formülasyonda Başla/ Bitir Formülasyonu ve Aç Kapa Formülasyonu olmak üzere 2 çeşit formülasyon vardır.

İlk olay tabanlı formülasyon olan Başla/ Bitir Olay Tabanlı Formülasyonda iki çeşit ikili değişken, iki çeşit de sürekli değişken bulunmaktadır.

x_{ie} (sırasıyla y_{ie}) değişkeninin 1'e eşit olması demek e olayında i aktivitesinin başlayıp bittiğini temsil eder. Böylece x değişkeni her bir aktivite için başlangıç zamanını y değişkeni de her bir aktivitenin bitiş zamanını belirler.

Sürekli değişken olan t_e ise e olayının tarihini temsil ederken, r_{ek} olay e 'nin hemen ardından gerekli olan kaynak k miktarını temsil eder.

İkinci olay tabanlı formülasyonundan olan Açık/Kapalı Formülasyonunda ise her bir olay için bir çeşit ikili değişken, bir çeşit de sürekli değişken bulunmaktadır.

z_{ie} i aktivitesi süresince 1'e eşittir. İşte bu yüzden bu modeli Açık / Kapalı Olay temelli formülasyon olarak adlandırılmıştır. Bu modelde, olayların sayısı aktivitelerin sayısına(n) eşittir. Sürekli değişken ise t_e ise e olayının tarihini temsil eder. Tez çalışmasında zaman endekli değişkenleri kullanan formülasyon kullanılacağından olay endekli değişkenleri kullanan matematiksel modeller detaylı olarak anlatılmamıştır. Formülasyonların detayları için [14] numaraları kaynağa başvurulabilir.

3. KKPÇP İÇİN DOĞRUSAL KARAR MODELİ

Bu bölümde çalışma yapılan firmaya ait genel bilgiler anlatılmış ve problemin tanımı detaylı olarak verilmiştir. Problemin çözümü için kullanılacak doğrusal karar modeline ait varsayımları, indeksler, kümeler, parametreler, karar değişkenleri tanımlanarak karar modeli ve kısıtlar açıklanmıştır.

Anlatılan gerçek hayat probleminin boyutu büyük olduğundan çözüme daha kolay ulaşmak adına öncelikle daha küçük boyutlu örnek bir uygulama yapılmıştır. Örnek olarak ele alınacak problemin verileri detaylı olarak belirtilmiş, matematiksel modeli yazılmış ve OPL programlama dilinde kodlanıp, CPLEX çözümü elde edilmiştir. Ayrıca elde edilen sonucun gantt şemaları çizilerek çözüm daha anlaşılır hale getirilmiştir.

3.1 Firmaya Ait Genel Bilgi

Çalışma yapılan firma 1982 yılında Ankara merkezli kurulmuştur. Firma, yazılım yoğun sistem alanlarında faaliyet göstermekte olup uzmanlığını, Komuta Kontrol Muhabere, Bilgisayar, İstihbarat, Gözetleme ve Keşif Sistemleri kapsamında, Hava Savunma Sistemleri, Deniz Savaş Sistemleri, Simülasyon ve Eğitim Sistemleri, Yönetim Bilgi Sistemleri, Enerji Yönetimi, Siber Güvenlik ve Anayurt Güvenliği alanlarına odaklanmıştır. Bütün bu gelişmelere ilave olarak, yurtdışı pazarında da hedefleri bulunmakta olup yurtdışı pazarda da büyümeye odaklanmıştır.

Firma, birçok alanda faaliyet göstermekte olup birçok projeyi aynı anda yürütebilmektedir. Bu nedenle firma için proje yönetimi oldukça önemlidir. Projeler de belirlenen takvime ve maliyete uygun olarak ilerlemek ve müşteri memnuniyetini yüksek tutmak firmanın hedefleri arasındadır.

3.2 Problemin Tanımı

Yapılan araştırmalar doğrultusunda çok projeli, kaynak kısıtlı bir proje çizelgeleme üzerinde çalışılmıştır. Firma içerisinde birden fazla projeye ait farklı işler aynı anda yapılabilmektedir. Bu çalışmada, en kısa zaman ile bu işlerin çizelgelenmesini sağlamak amaçlanmaktadır. Özetle öncüllük ilişkileri olan farklı faaliyetlere sahip, aynı kaynak havuzundan beslenen birden çok projenin toplam proje süresi optimize edilerek en kısa toplam proje süresi hesaplanması amaçlanmaktadır.

Çalışma yapılan firmada proje çizelgeleme yapabilmek için planlama aşamasında projelerdeki faaliyetlerin sayısı, süresi, kaynak çeşitleri, faaliyetleri gerçekleştirecek kaynak sayısı, faaliyetlerin öncüllük ardılık ilişkileri ve kaynak kapasiteleri verileri firmanın kullandığı proje yönetim yazılımından alınır.

Firmada kullanılan kaynaklar yenilenebilir kaynaklardır. Yenilenebilir kaynaklar birinci bölümde de belirtildiği gibi birim zaman dilimi içerisinde kullanım miktarları kısıtlı olup, o zaman dilimi bittikten sonra tekrar kullanıma hazırdırlar. Kaynaklar belirli faaliyet bittikten sonra tekrar kullanılabilir. Bu çalışmadaki yenilenebilir kaynak olarak iş gücü kullanılmaktadır.

Çalışma yapılan firmada iki çeşit yenilenebilir kaynak tipi bulunmaktadır. Bunlardan bir tanesi Teknoloji A (TA) sınıfı, ikincisi ise Teknoloji B (TB) sınıfıdır. Teknoloji A sınıfı çalışanları ile Teknoloji B sınıfı çalışanları arasında tecrübe farkı bulunmaktadır. Projelerdeki faaliyetlerin kapsamına göre proje yönetimi tarafından kaynak atamaları yapılır. Özetle, proje başlamadan önce planlama aşamasında faaliyetlerin kaynak tipleri ve bu kaynak tiplerinin sayısı bellidir. Örneğin, bir projenin ilk faaliyetini 3 Teknoloji A çalışanı, 2 Teknoloji B çalışanı gerçekleştirecektir.

Firmada kaynaklar projelere dedike çalışmaktadırlar. Firma yönetimi tarafından dedike çalışmanın projelere ve firmaya aidiyet duygusunu arttırdığı düşünülmektedir. Ayrıca çalışan farklı işlerde bölünmediğinden çalışandan alınan verime de arttırdığı düşünülmektedir.

Teknoloji A ve Teknoloji B kaynak tipleri çalışan sayısı yani kaynakların kapasiteleri firmada belirlidir. Problem ile ilgili doğrusal karar modelinde kaynakların kapasitesi de kısıtlar arasında yer alacaktır.

Proje başlamadan önce planlama aşamasında öncüllük ardılık ilişkileri belirlenir. Bir faaliyetin başlaması için ondan önce hangi faaliyetlerin tamamlanmış olması gerektiği, başlayan bir faaliyeti hangi faaliyetlerin takip etmesi gerektiği ya da paralel aktiviteler belirlenir. Projelerdeki öncüllük ilişkileri proje yöneticileri ve teknik yöneticiler tarafından belirlenmektedir. Proje yöneticileri, proje planlama ile ilgili faaliyetlere girdi sağlarken, teknik yöneticiler de analiz,tasarım gibi mühendislikle ilgili kısımlara görüş vermektedir.

Birinci bölümde belirtildiği gibi projeler arasındaki öncüllük ilişkileri faaliyet ağ diyagramları ile gösterilir. Tez çalışmasında projeler için çizilen faaliyet ağ EK 3, EK 4, EK 5, EK 6, EK 7 'de belirtilmiştir.

3.3. Doğrusal Karar Modeli

KKPÇ problemi için varsayımlar aşağıda sıralanmıştır.

- i. Faaliyet süreleri deterministiktir.
- ii. Faaliyetlerin birim zaman kaynak kullanımı sabittir.
- iii. Bir faaliyete atanan kaynak faaliyet süresince o faaliyet tarafından kullanılır.
- iv. Başlatılan faaliyetler kesintisiz bitirilmek zorundadır; ara verilemez.
- v. Faaliyetler iptal edilemez. Proje serimindeki her faaliyet gerçekleştirilmek zorundadır [24] .

Matematiksel model oluşturulurken indeksler, kümeler, parametreler, karar değişkenleri tanımlanarak karar modeli ve kısıtlar belirtilmiştir.

Kümelerin Tanımları:

P : Proje Sayısı

A_p : P Projesindeki Aktivite Sayısı

- $A_{First}=0$ Projeler serimindeki başlangıç düğümü(yapay)
- $A_{Last} = \max \{ A[1] , A[2] , \dots, A[P] \} + 1$ Projeler serimindeki bitiş düğümü(yapay)

C_r : r Kaynağının Kapasitesi

R : Kaynak Sayısı

T : Planlama Ufku

S_p : P Projesindeki Öncüllük Kümesi

- $S_p = \{ (i,j) \}$ p projesinde i faaliyetinden sonra j faaliyeti gelmelidir.

İndekslerin Tanımları:

$i,j= 0, \dots, A+1$ Aktivite indeksi

$t=0, \dots, T$ Zaman periyodu indeksi

$r=1, \dots, R$ Kaynak indeksi

$p=1, \dots, P$ Proje indeksi

Parametrelerin Tanımları:

C_{rt} : t zamanında r kaynağının kapasitesi

u_{rpi} : p projesinde i aktivitesini gerçekleştirmek için r kaynağının tüketim miktarı

d_{pi} : p projesinde i aktivitesinin süresi

Karar Değişkenleri:

$PTZ \geq 0$ (PTZ: Proje Tamamlanma Zamanı)

$$x_{ipt} = \begin{cases} 1, & p \text{ projesindeki } i. \text{ Faaliyeti } t \text{ zamanda başlarsa} \\ 0, & p \text{ projesindeki } i. \text{ Faaliyet } t \text{ zamanda başlamaz ise} \end{cases}$$

Doğrusal Karar Modeli

Beş farklı kısıt altında amaç fonksiyonu anlatılmıştır.

$$tX_{A_{Last},p,t} \leq \text{ProjeTamamlanmaZamanı} \quad p=1, \dots, P ; t=0, \dots, T \quad (3.1)$$

$$\sum_{t=0}^T x_{A_{Last},p,t} = 1 ; p=1, \dots, P \quad (3.2)$$

$$\sum_{t=0}^T x_{i,p,t} = 1, i=1, \dots, A_p ; p=1, \dots, P \quad (3.3)$$

$$\sum_{t=0}^T tx_{j,p,t} - \sum_{t=0}^T tx_{i,p,t} \geq d_{p,i} \quad \forall (i,j) \in S_p ; p=1, \dots, P \quad (3.4)$$

$$\sum_{p=1}^P \sum_{i=1}^{A_p} u_{r,p,i} \left(\sum_{z=\max\{t-d_{p,i}+1,0\}}^t x_{i,p,z} \right) \leq C_r \quad r=1, \dots, R; t=0, \dots, T \quad (3.5)$$

kısıtları altında

$$\text{enk ProjeTamamlanmaZamanı} \quad (3.6)$$

3.6 numaralı amaç fonksiyonu projelerin tamamlanma zamanlarını enküçüklemektedir yani karar değişkeni olan, proje tamamlanma zamanının en

küçük değeri aranmaktadır. 3.1 numaralı kısıt projeler serimine eklenen yapay faaliyetin başlama zamanı projelerin tamamlanma zamanından küçük veya eşit olacağını; 3.2 numaralı kısıt ise projeler serimine eklenen yapay faaliyetin, her projede yalnız bir zaman diliminde başlayacağını ifade eder. 3.3 numaralı kısıtta ise her projenin her faaliyeti yalnızca bir zaman aralığında başladığını, 3.4 numaralı kısıtta ise her projedeki faaliyetlerin öncüllük ardıllık ilişkisine uygun olarak başlayacağı anlatılmaktadır. 3.5 numaralı kısıt herhangi bir zaman aralığında, herhangi bir kaynağın bütün projelerin bütün faaliyetlerinin kaynak kullanım miktarları toplamı herhangi bir t zaman aralığında herhangi bir r kaynağının kapasitesinden küçük olacağını tanımlar.

3.4 Örnek Problem

Bölüm 3.2'de anlatılan gerçek hayat probleminin boyutu büyük olduğundan, çözüme daha kolay ulaşmak adına öncelikle daha küçük boyutlu örnek bir uygulama yapılmıştır. Örnek olarak ele alınacak problemin verileri bu bölümde detaylı olarak belirtilmiş, matematiksel modeli yazılmış ve OPL programlama dilinde kodlanıp, CPLEX çözümü elde edilmiştir.

Elde edilen sonuç için projelerin Gantt şemaları da çizilerek çözüm daha anlaşılır hale getirilmiştir. Grafiklerde kaynak kapasitesinin zamana göre değişimi de izlenmekte olup, toplam kullanılan kaynaklarında kaynak kapasitesini aşmadığı ayrıca gözlemlenmiştir.

3.4.1 Örnek problemin verileri

Problemin verileri; projelerin faaliyet sayıları, faaliyetlerin süreleri ve kaynak tiplerine göre kaynak kullanım miktarları Çizelge 3.1, Çizelge 3.2 ve Çizelge 3.3 'de belirtilmiştir.

1.kaynak tipinden firmada 6 kişi varken, ikinci kaynak tipinden 8 kişi bulunmaktadır. Yani kaynakların kapasiteleri 6 ve 8 'dir. 3 projenin toplam planlama ufku 50 günü geçmemelidir.

Çizelge 3.1 Birinci Proje Verileri

Proje Faaliyetleri	Faaliyet Süreleri	1.Kaynak Tipinin kullanım miktarı	2. Kaynak Tipinin Kullanım Miktarı	Öncüllük İlişkileri
1	3	2	1	-
2	5	3	2	-
3	1	3	2	-
4	3	1	0	1
5	2	1	0	2,3
6	4	2	1	4
7	5	3	4	2,5
8	6	1	2	5
9	4	1	2	5
10	4	1	2	6,7

Çizelge 3.2 İkinci Proje Verileri

Proje Faaliyetleri	Faaliyet Süreleri	1.Kaynak Tipinin kullanım miktarı	2. Kaynak Tipinin Kullanım Miktarı	Öncüllük İlişkileri
1	2	1	2	-
2	4	2	3	-
3	3	1	0	2
4	5	2	0	1
5	7	2	2	4,3
6	6	2	1	5

Çizelge 3.3 Üçüncü Proje Verileri

Proje Faaliyetleri	Faaliyet Süreleri	1.Kaynak Tipinin kullanım miktarı	2. Kaynak Tipinin Kullanım Miktarı	Öncüllük İlişkileri
1	2	1	2	-
2	4	2	3	1
3	6	3	3	1
4	8	1	3	1
5	3	1	2	2
6	5	1	2	3
7	7	2	3	4
8	9	3	3	5,6,7
9	3	1	1	8

3.4.2 Örnek uygulamanın çözümü

Matematiksel model Optimizasyon Programlama Dili (OPL) yardımıyla kodlanarak, karar modeli CPLEX çözücüsü ile çözdürülmüş ve sonuçları incelenmiştir. Karar modelinin OPL kodu EK-1, çözümü ise Ek-2 'de verilmiştir.

CPLEX'de matematiksel model çalıştırıldığında 3 projenin de toplam en kısa tamamlanma zamanı 35 bulunmaktadır. Ayrıca 3 proje içinde en kısa tamamlanma zamanlarına ulaşmak için gerekli olan x_{ipt} değerleri de çözümde belirtilmiştir.

Daha öncede bahsedildiği gibi x_{ipt} p projesindeki i. faaliyetin t zamanında başlama değerini göstermektedir.

Örneğin;

$x[1,3,0] = 1$ eşitsizliğinin anlamı; 3. Projedeki 1. Faaliyet t= 0 zamanında başlayacağı ifade etmektedir.

CPLEX çözümde bulunan x_{ipt} değerleri, aktivitelerin süreleri ve her bir kaynak çeşidinden kullanılan miktarlarına göre 3 projenin Gantt Şemaları çizilmiştir. Problemden 2 tip kaynak bulunduğundan , Çizelge 1 kaynak kapasitenin 6 olduğu birinci kaynak tipinin Gantt şemasını gösterirken, Çizelge 2 kaynak kapasitenin 8 olduğu ikinci kaynak tipinin gantt şemasını göstermektedir.

Şekil 3.1 ve Şekil 3.2 'de belirtilen grafikler de kutuların içerisinde yazan değerler p projedeki i. faaliyet değerlerini içermekte olup x_{ip} notasyonu ile gösterilmiştir. Grafik çizilirken CPLEX çözümündeki x_{ipt} değerleri kullanılmıştır.

Gantt şemasında x eksenini zamanı y eksenini ise kaynak tüketim miktarını göstermektedir. Çizelge 1'in nasıl çizildiği kısaca özetlenirse;

Birinci proje için grafiğin çizilmesi için öncelikle OPL çözümünde p=1 olan ve aşağıda belirtilen x_{ipt} değerleri kullanılacaktır.

$$x[3,1,0] = 1$$

$$x[2,1,1] = 1$$

$$x[1,1,4] = 1$$

$$x[4,1,7] = 1$$

$$x[5,1,17] = 1$$

$$x[6,1,19] = 1$$

$$x[7,1,23] = 1$$

$$x[8,1,28] = 1$$

$$x[9,1,31] = 1$$

$$x[10,1,31] = 1$$

$$x[11,1,35] = 1$$

$x[3,1,0] = 1$; birinci projede üçüncü faaliyet $t= 0$ zaman diliminden başlayacaktır. Birinci projedeki üçüncü faaliyetin süresi bir gün, birinci tip kaynak kullanım miktarı da üç olduğundan grafikte x ekseninde bir gün ilerletip, y ekseninde ise yüksekliği üç olarak gösterilmiştir.

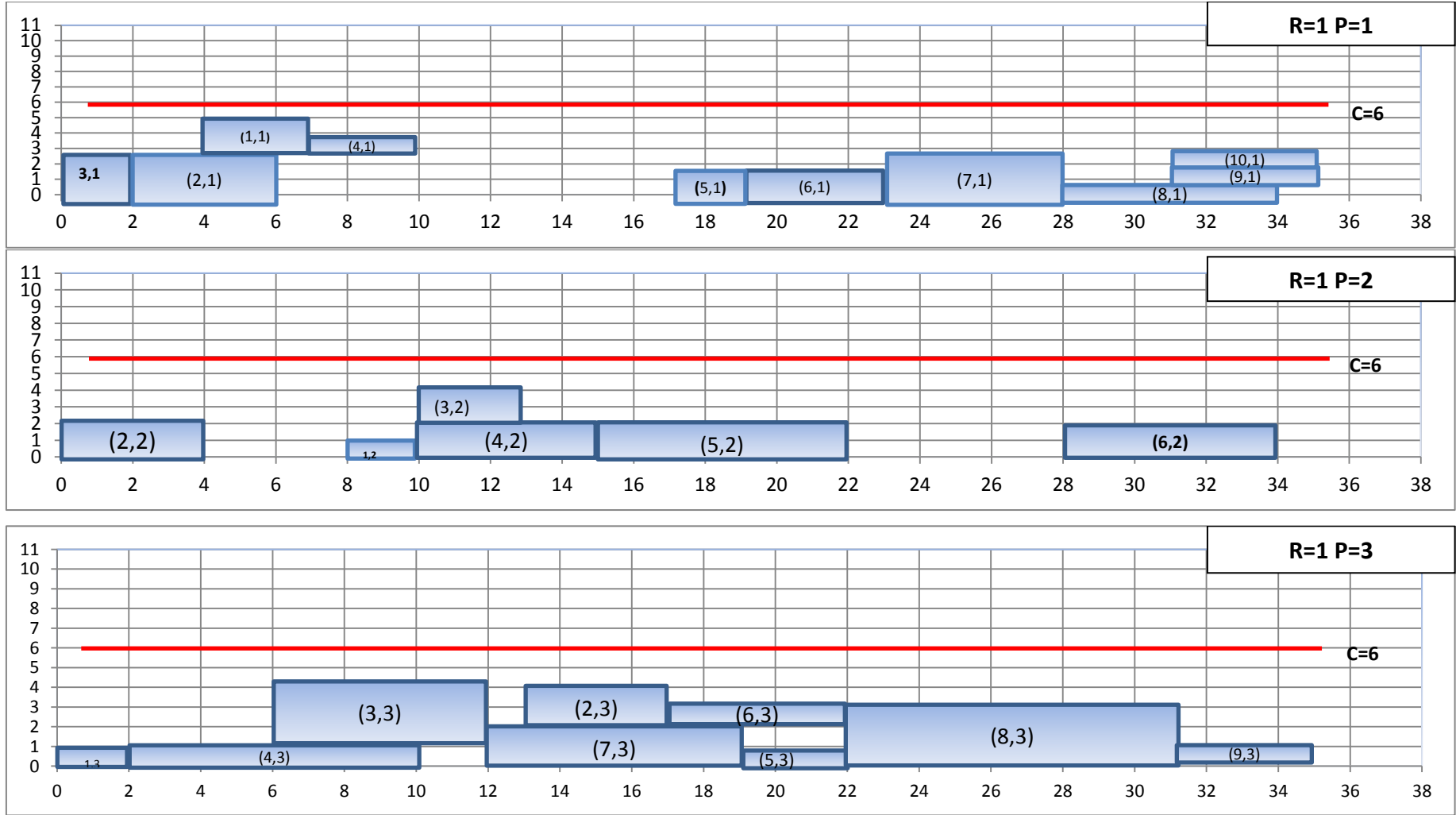
$x[2,1,1] = 1$; birinci projedeki ikinci faaliyeti $t= 1$ zaman diliminden başlayacaktır. Birinci projedeki ikinci faaliyetin süresi beş gün, birinci tip kaynak kullanım miktarı da üç olduğundan grafikte x eksenini beş gün ilerletip, y ekseninde ise yüksekliği üç olarak gösterilmiştir.

Bu işlemler iki kaynak tipi ve üç projenin bütün x_{ipt} değerleri için çizildiğinde Şekil 3.1 ve Şekil 3.2 'deki grafikler elde edilmiştir. İki kaynak tipi için de üç projenin en kısa tamamlanma zamanının 35 olduğu da açıkça görülmektedir.

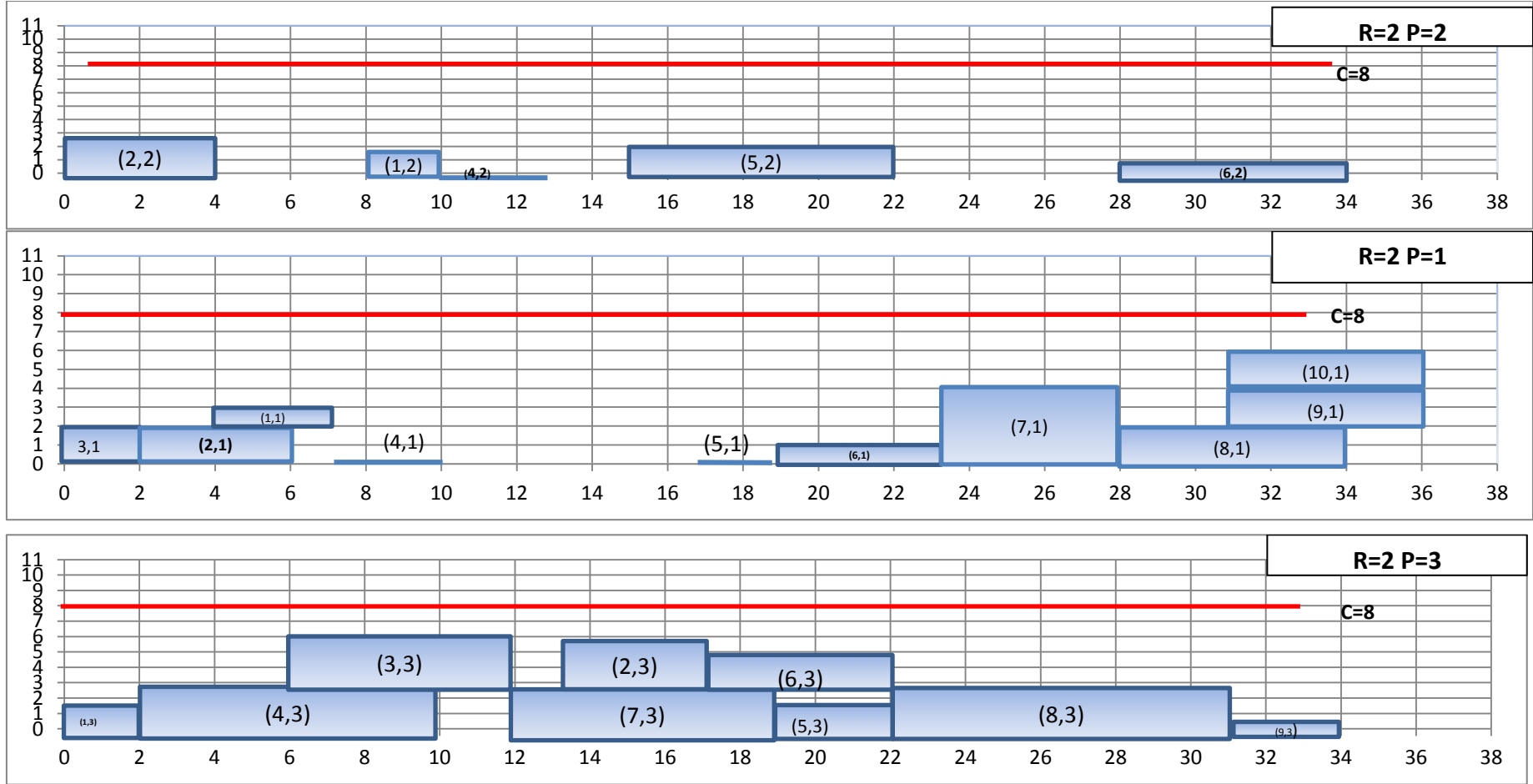
Ayrıca grafiklerde 1. kaynak tipi ve 2. kaynak tipi için 3 projenin ayrı ayrı tamamlanma zamanları görülmektedir. Örneğin, çizelge-1 incelendiğinde 1. kaynak tipi için 1. projenin tamamlanma süresi 35 gün, 2. Projenin tamamlanma süresi 34 gün ve 3. Projenin tamamlanma süresinin 35 gün olduğu okunmakta olup, 3 projenin de toplam en kısa tamamlanma zamanın da 35 gün olduğu sonucuna varılmaktadır.

Son olarak grafiklerden de anlaşıldığı gibi; 3 proje, 2 kaynak tipi için çizilen 6 grafikte kaynak kullanım miktarları kaynak kapasitelerini aşmamaktadır. 1. tip kaynak için çizilen Gantt şemasında kaynak kapasitesi 6'yı, 2. tip kaynak için çizilen Gantt şemasında kaynak kapasitesi 8'i geçmemiştir.

Sonuç olarak Gantt şemaları sayesinde modelin verilen bir problem verisi için olurlu bir çözüm bulduğu görülmüş, kurulan model ve kısıtlar daha anlaşılır ve yorumlaması kolay olmuştur.



Şekil 3.1 Birinci Kaynak Tipi 3 Proje Gantt Şeması



Şekil 3.2 İkinci Kaynak Tipi 3 Proje Gantt Şeması

4. GERÇEK HAYAT PROBLEMİ

Bölüm 3.4.1’de çok projeli kaynak kısıtlı proje çizelgeleme konusunda örnek bir uygulama üzerinden gidilmiş, matematiksel model OPL programında kodlanmış, çözümleri alınmış ve çözüm aynı zamanda Gantt şeması ile gösterilmiştir.

Bu bölümde ise Bölüm 3.2 ‘de detaylı olarak anlatılan problemin verileri ve çözümü verilecektir. Problemden özel bir yazılım firmasındaki genel müdür yardımcılığının altındaki 5 proje incelenmiştir. Firmanın gizliliğini sağlamak amacıyla projelerin ve projedeki faaliyetlerin adları paylaşılmamıştır. Proje isimleri harflerle sembolize edilmiştir.

Bu çalışmada; öncelik ilişkileri olan farklı faaliyetlere sahip, aynı yenilenebilir kaynak havuzundan beslenen birden çok projenin toplam proje süresi optimize edilerek en küçük toplam proje süresinin hesaplanması amaçlanmaktadır.

4.1 Problemin Verileri

Beş projenin zamanlama ufku (T) maksimum 700 gün olarak belirlenmiştir. Zamanlama ufku, projelerin toplam sözleşmesel sürelerinin üzerine yüzde yirmi gecikme payı eklenerek hesaplanmıştır. Literatürde zamanlama ufku, genellikle kritik yol metodu ile bulunmaktadır. T büyüdükçe problemi çözmek zorlaşır bu nedenle zamanlama ufkunun doğru tespit etmek problemlerin çözümü için kritik önem taşımaktadır. Problemden Teknoloji A (TA) ve Teknoloji B (TB) olmak üzere iki çeşit kaynak tipi bulunmaktadır. Bu kaynakların kapasiteleri ise TA için 50 kişi, TB içinde 45 kişi olarak belirlenmiştir. Problemin verileri şirketin kullandığı bir proje yönetim yazılımından alınmıştır. Çizelge 4.1, A projesinin; Çizelge 4.2, B projesinin; Çizelge 4.3, C projesinin; Çizelge 4.4, D projesinin; Çizelge 4.5 ise E projesinin faaliyet sayılarını, faaliyet sürelerini, faaliyetleri gerçekleştirmek için 1. kaynak tipinin kaynak kullanım miktarını ve 2. Kaynak tipinin kaynak kullanım miktarı verilerini göstermektedir.

Çizelge 4.1 A Projesinin Verileri

Proje Faaliyetleri	Faaliyet Süreleri (Gün)	1.Kaynak Tipinin kullanım miktarı (TA)	2. Kaynak Tipinin Kullanım Miktarı (TB)
1	1	3	2
2	50	4	3
3	60	5	4
4	35	4	3
5	10	1	1
6	45	6	5
7	15	2	1
8	4	1	1
9	2	4	3
10	1	1	1
11	60	5	2
12	35	4	2
13	10	1	1
14	45	6	4
15	15	2	3
16	10	3	5
17	1	2	1
18	1	2	1
19	5	3	2
20	1	4	3

Çizelge 4.2 B Projesinin Verileri

Proje Faaliyetleri	Faaliyet Süreleri (Gün)	1.Kaynak Tipinin kullanım miktarı (TA)	2. Kaynak Tipinin Kullanım Miktarı (TB)
1	1	2	3
2	40	3	2
3	40	6	4
4	45	5	4
5	35	1	2
6	25	4	2
7	40	3	5
8	20	2	1
9	25	4	3
10	30	3	0
11	35	6	4
12	20	2	1
13	50	2	0
14	50	2	0
15	30	5	4
16	60	2	1

17	90	6	3
18	65	4	2
19	70	2	1
20	80	3	4
21	20	5	2
22	3	4	1
23	1	1	0
24	90	3	2
25	90	3	1
26	70	3	4
27	70	3	0
28	70	3	1
29	25	3	1

Çizelge 4.3 C Projesinin Verileri

Proje Faaliyetleri	Faaliyet Süreleri (Gün)	1.Kaynak Tipinin kullanım miktarı (TA)	2. Kaynak Tipinin Kullanım Miktarı (TB)
1	1	1	2
2	80	3	0
3	80	5	4
4	70	5	3
5	65	6	2
6	45	4	0
7	55	5	4
8	50	4	2
9	30	3	2
10	5	1	3
11	7	3	0
12	5	2	1
13	4	1	1
14	10	2	1
15	20	4	3
16	15	2	3
17	5	2	4
18	10	5	2
19	15	4	3
20	50	1	0
21	10	3	2
22	80	4	3
23	80	4	1
24	80	4	5
25	80	3	2
26	80	2	1

Çizelge 4.4 D Projesinin Verileri

Proje Faaliyetleri	Faaliyet Süreleri (Gün)	1.Kaynak Tipinin kullanım miktarı (TA)	2. Kaynak Tipinin Kullanım Miktarı (TB)
1	1	1	2
2	50	3	0
3	45	3	2
4	93	6	5
5	20	2	1
6	50	3	2
7	45	3	0
8	95	6	4
9	20	2	4
10	45	4	2
11	50	2	3
12	5	3	2
13	7	2	1
14	13	1	0
15	7	2	1
16	20	1	2
17	30	4	5
18	15	2	0
19	1	1	1
20	60	3	2

Çizelge 4.5 E Projesinin Verileri

Proje Faaliyetleri	Faaliyet Süreleri	1.Kaynak Tipinin kullanım miktarı (TA)	2. Kaynak Tipinin Kullanım Miktarı (TB)
1	1	1	1
2	80	5	4
3	75	4	2
4	80	3	2
5	80	2	1
6	80	1	2
7	80	0	1
8	80	2	1
9	80	4	3
10	80	2	0
11	5	4	2
12	1	1	0
13	30	4	2
14	30	2	1
15	5	4	3
16	1	1	1

17	15	2	4
18	15	4	2
19	5	4	3
20	4	1	5
21	4	1	0
22	7	2	0
23	10	4	3
24	5	3	2
25	60	2	0
26	15	2	1
27	5	2	4
28	10	3	7
29	15	6	0
30	20	5	3
31	8	3	2
32	2	0	1
33	1	1	2
34	1	1	2
35	50	2	1

Projelerdeki öncüllük ilişkilerini göstermek için ise A projesinin faaliyet ağ diyagramı EK 3, B projesinin faaliyet ağ diyagramı EK 4, C projesinin faaliyet ağ diyagramı EK 5, D projesinin faaliyet ağ diyagramı EK 6, E projesinin faaliyet ağ diyagramı ise EK 7' de belirtilmiştir.

5 proje için toplam A faaliyet bulunmaktadır. A+1, son faaliyeti ifade eden yapay bir faaliyettir. Dolayısıyla bütün projeler için de son faaliyet A+1 olarak olarak tanımlanır. Tanımlanan A+1 faaliyeti bütün projelerin en son aktivitesinden sonra başlamaktadır. A+1 faaliyetinin başlaması, bütün faaliyetlerin bittiği anlamına gelmektedir. Dolayısıyla her projenin son faaliyeti A+1'in öncülü olarak oluşturulur.

Faaliyet ağ diyagramları çizilirken de beş projenin de son aktivitesi yapay olan A+1 faaliyetidir. 5 proje için en son faaliyet E projesindeki 36 numaralı faaliyet olduğundan faaliyet ağ diyagramlarında da bütün projelerin son faaliyetlerinin E-36 olduğu gözlemlenme olup ve kesikli çizgi ile gösterilmiştir.

4.2 Problemin Çözümü

Beş proje için faaliyet sayıları, faaliyetlerin süreleri, öncüllük ilişkileri, kaynak tipi sayısı, kaynak kapasitesi, faaliyetleri gerçekleştirmek için hangi kaynak tipinden kaç kişi kullanacağı verileri çalışma yapılan firmadan alınmıştır. Bu veriler

doğrultusunda ÇPKKPÇP için Bölüm 3.3.'de verilen doğrusal karar modeli, Optimizasyon Programlama Dilinde (OPL) kodlanmıştır. OPL kodu EK 8'de, CPLEX çözümü ise EK 9 'da belirtilmiştir.

Problemin çözümü incelendiğinde optimum sonuç olarak 456 bulunmuştur. Optimum sonuç beş projenin toplam en küçük tamamlanma zamanını vermektedir. Beş projenin en küçük tamamlama zamanına ulaşması için olması gereken x_{ipt} (i. Faaliyet p. proje t zamanda başlaması) değerleri de EK- 9'da belirtilen CPLEX çözümünde verilmiştir.

4.3 Model Sayesinde Yapılabilecek Analizler

OPL modelinin çeşitli kısıtlar altında çalıştırılması sonucu elde edilen sonuca göre beş projenin en küçük toplam tamamlanma süresi 456 'dır. Ayrıca beş projeyi en kısa sürede bitirmek için projelerdeki faaliyetleri hangi zaman diliminde başlatılması gerektiği konusunda da model bilgi vermektedir.

Problem için kullanılan veri seti (proje sayısı, projelerdeki aktivite sayısı, aktivitelerin süreleri, öncüllük ilişkileri, kaynak tipi sayısı, faaliyetleri gerçekleştirmek için kullanılan kaynak sayısı ve kaynak kapasiteleri) çalışma yapılan firmadan alınmıştır. Bu veriler doğrultusunda, kısıtlar belirlenmiş ve bu kısıtlar altında da çok projeli kaynak kısıtlı proje çizelgeleme modeli çalıştırılarak, optimum sonuç yani projelerin toplam en kısa tamamlanma süresi bulunmuştur. Model aynı zamanda optimum sonuca ulaşmak için projelerdeki faaliyetleri hangi t zamanında başlatılması gerektiği konusunda da bilgi vermektedir.

Bu durumda firma çalışanları proje sayısı, projelerdeki aktivite sayısı, aktivitelerin süreleri, öncüllük ilişkileri, kaynak tipi sayısı, faaliyetleri gerçekleştirmek için kullanılan kaynak sayısı ve kaynak kapasitelerini değiştirerek projelerin en kısa sürede tamamlanması konusunda analizler yapıp sonuçlarını inceleyebilir ve yönetime raporlayabilirler.

Model çözümü sayesinde yapabilecek analizden bahsedilirse;

- Firma çalışanları, proje planlamasının doğru yapıp yapılmadığı hakkında fikir sahibi olabilirler. Projelerin en kısa tamamlanma süresi toplamı, projelerin

toplam sözleşmesel tamamlanma sürelerini geçmeyecek şekilde planlama yapılması gerekmektedir.

- Beş proje, iki kaynak tipi için çok projeli kaynak kısıtlı proje çizelgeleme matematiksel modeli çalıştırıldığında en kısa tamamlanma süresinin yani optimum sonucun 200 gün çıktığı varsayılın. Fakat sözleşmeye göre projelerin toplam süreleri 180 gündür ve 180 günden fazla çalışılan her gün için firma günlük projelerin sözleşme bedelinin binde 5'i kadar cezaya girmektedir. Ayrıca firma sektörde itibar da kaybetmek istemeyip, müşteri memnuniyetini yüksek tutmak istemektedir. Bu nedenle cezaya girmemek için önerilen matematiksel modeldeki değişkenleri değiştirerek projelerin en kısa tamamlanma zamanını hesaplayarak analizler yapılabilir.
 - ✓ Projedeki kaynak sayısını arttırabilir.
 - ✓ Kaynak kapasitelerini arttırabilir.
 - ✓ Paralel yapıya getirebilecek faaliyetlerin öncüllük ilişkilerini değiştirebilir.
 - ✓ Faaliyetlerin sürelerini kısaltarak, o faaliyet için kullanılan kaynak sayıları değiştirebilir.
 - ✓ Kaynak tipi sayıları değiştirebilir.
 - ✓ Kaynak çeşidi sayısı arttırılabilir.
- Proje sayısını arttırarak elindeki kaynaklarla projelerin en kısa toplam kaç gün de tamamlanacağını hesaplanabilir. Böylece firma yetkilileri personel alıp almamaya karar verebilir.
- Firmanın elindeki 5 proje incelenerek 5 projenin en kısa tamamlanma süresi hesaplanmıştır. Fakat firmada teklif aşamasında çalıştığı 3 proje daha olduğunu varsayılın ve bu 3 projede firmanın elindeki 5 proje bitmeden başlamaktadır. Firmaya, genel müdürün kesin talimatı üzerine yeni personel alınamamaktadır. Bu durumda elindeki mevcut kaynakları kullanarak 8 projenin en kısa tamamlanma zamanı bulanabilir. Bulunan sonuç sayesinde teklif aşamasındaki projelerin şartnamede yazan proje sürelerine uyup uymayacağı kontrol edilir. Kaynak sıkıntısından dolayı yeni gelecek projelere teklif verilemeyecekse bu durum genel müdüre açıklanarak özel izinle kaynak alınmasını sağlanabilir.

- Firmanın stratejik kararı üzerine 5 projenin de toplam 2 ay erken bitirilmesi genel m¼d¼r tarafından talep edilmiř olsun. Bu durumda ka kaynak daha alınarak 5 projeyi 2 ay erken bitirebilirim analizi yapılabilir.

Sonuç olarak, model sayesinde firmanın ihtiyalarına g¼re eřitli olasılıklar denenerek, projelerin en kısa tamamlanma s¼resinin ne olacađı hakkında ¼nceden bilgi edinilmektedir.

Model öz¼m¼n¼n firmaya sađladıđı avantajlar dıřında, modelin de firmaya sađladıđı faydalar bulunmaktadır. Uygulama yapılan tez alıřmasında dođrusal karar modeli alıřtırılarak, öz¼m alınmıř ve sonuları incelenmiřtir. Fakat model alıřtırıldıđında sonu alınamaz ise firmanın elindeki kaynaklarla projeleri tamamlamayacađı yorumu ıkarılabilir. Ayrıca, kurulan modelin proje y¼netimine olumlu katkısı sayesinde m¼řteri memnuniyeti de artmaktadır.

5. ARAYÜZ GELİŞTİRİLMESİ

Modelin kullanılıp firmaya yarar sağlaması için firma çalışanları tarafından etkin kullanılmalıdır. Bu etkinliği arttırmak için firma çalışanlarına eğitim verilebilir. Eğitim verilmesinin yanı sıra OPL programının doğru kullanılıp, verilerin doğru girilmesi ve doğru analiz edilmesi için firmada endüstri mühendisinin çalıştırılması ve yok ise yeni personel alımına gidilmesi önerilmektedir. Fakat yeni personel alımı, çalışma yapılan firma için ek maliyet getireceğinden firma yeni personel alımına sıcak bakmayabilir.

Firmada endüstri mühendisi olsa bile önerilen modelin kullanılması ve analizler yapabilmesi için çalışanlar zamanlarının büyük bir kısmını verilerin OPL programına girilmesi ve kodun hazırlanması için harcaması gerekmektedir. Firma, çalışanlarından daha farklı işlerde faydalanmak isteyebilir ve fayda-maliyet analizi yapıldığında zamanlarını OPL programında harcamalarını tercih etmeyebilir.

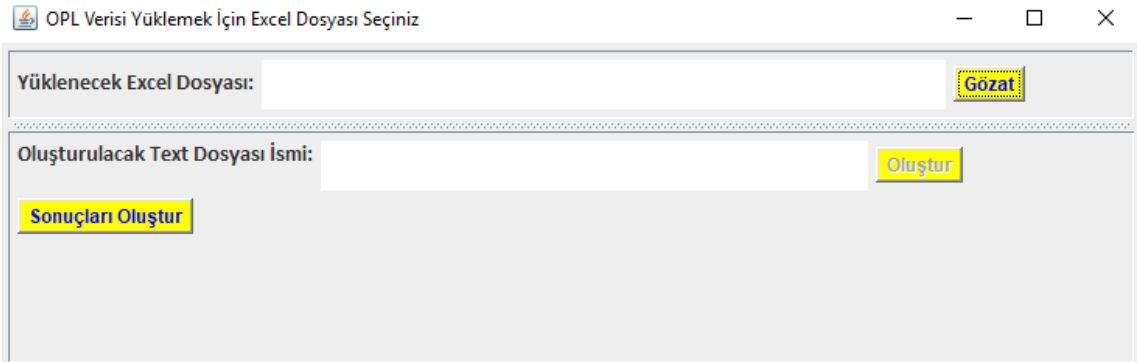
Firma çalışanları genellikle yeniliklere ve değişime çok çabuk ayak uydurmadıklarından firmalar için önerilen yöntem kullanıcı dostu olmalı ve çalışanları zora sokmamalıdır. Firma çalışanları önerilen yöntemden sonuç almak için hazırlık aşamasında vakitlerinin büyük bir kısmını harcayorsa ve bu durum mevcut işlerini aksatarak çalışanı zora sokuyorsa çalışanlar önerilen yöntemin kullanılmasında direnç göstermektedirler.

Firma çalışanlarının oluşabilecek direncini kırmak ve çalışma yapılan firmada kullanımı yaygınlaştırmak için tez çalışmasında JAVA uygulaması ile yazılım geliştirilerek ara yüz tasarlanmıştır.

Geliştirilen yazılım sayesinde, firmada excel bilgisine sahip ve yönetim kadrosunun yetki verdiği bütün kullanıcılar projeler ile ilgili durumları değerlendirerek sonuçları analiz edebileceklerdir.

Yazılım Java programlama dili kullanarak bir masaüstü uygulaması olarak tasarlanmış ve yazılım geliştirme aracı olarak Eclipse Oxygen kullanılmıştır.

Yazılım işlev olarak belli bir formatta hazırlanan Excel dosyasını girdi olarak alıp bu Excel dosyasındaki veriyi okuyup, okunan veriyi OPL programına girdi olacak olan koda çevirerek txt uzantılı dosya formatında yazmaktadır. Sonrasında OPL yardımıyla oluşan kodu CPLEX çözücüsü ile çözdürmek için hazırlanan run.bat dosyası çalıştırılır. Şekil 5.1 'de ekran görüntüsü alınan, tasarlanan ara yüzde, sonuçları oluştur butonuna basıldığında ise kullanıcılar projelerin en kısa tamamlanma zamanını ve en kısa tamamlanma zamanına ulaşmak için hangi projedeki hangi faaliyetin hangi t zamanında başlaması gerektiği bilgisine Şekil 5.2 'de belirtilen metin bilgisi olarak ulaşabilirler.



Şekil 5.1 Yazılım Sayesinde Tasarlanan Ara Yüz

```
Dosya Düzen Biçim Görünüm Yardım
Projelerin En Erken Tamamlanma Zamanı: 35
1.Proje:
1. Faaliyet 4. Zaman aralığında başlayacaktır.
2. Faaliyet 1. Zaman aralığında başlayacaktır.
3. Faaliyet 0. Zaman aralığında başlayacaktır.
4. Faaliyet 7. Zaman aralığında başlayacaktır.
5. Faaliyet 17. Zaman aralığında başlayacaktır.
6. Faaliyet 19. Zaman aralığında başlayacaktır.
7. Faaliyet 23. Zaman aralığında başlayacaktır.
8. Faaliyet 28. Zaman aralığında başlayacaktır.
9. Faaliyet 31. Zaman aralığında başlayacaktır.
10. Faaliyet 31. Zaman aralığında başlayacaktır.
11. Faaliyet 35. Zaman aralığında başlayacaktır.
2.Proje:
1. Faaliyet 8. Zaman aralığında başlayacaktır.
2. Faaliyet 0. Zaman aralığında başlayacaktır.
3. Faaliyet 10. Zaman aralığında başlayacaktır.
4. Faaliyet 10. Zaman aralığında başlayacaktır.
5. Faaliyet 15. Zaman aralığında başlayacaktır.
6. Faaliyet 28. Zaman aralığında başlayacaktır.
11. Faaliyet 34. Zaman aralığında başlayacaktır.
3.Proje:
1. Faaliyet 0. Zaman aralığında başlayacaktır.
2. Faaliyet 13. Zaman aralığında başlayacaktır.
3. Faaliyet 6. Zaman aralığında başlayacaktır.
4. Faaliyet 2. Zaman aralığında başlayacaktır.
5. Faaliyet 19. Zaman aralığında başlayacaktır.
6. Faaliyet 17. Zaman aralığında başlayacaktır.
7. Faaliyet 12. Zaman aralığında başlayacaktır.
8. Faaliyet 22. Zaman aralığında başlayacaktır.
9. Faaliyet 31. Zaman aralığında başlayacaktır.
11. Faaliyet 34. Zaman aralığında başlayacaktır.
```

Şekil 5.2 Ara Yüz Çıktısı

Yazılımda OPL kodunun üretilmesi için girdi olacak Şekil 5.3 'de ekran görüntüsü verilen Excel dosyasının formatı kullanıcı ile paylaşılır. Kullanıcının dikkat etmesi gereken faaliyet sayısı, kaynak kapasitesi, faaliyet süreleri, faaliyetleri gerçekleştirmek için kullanılan kaynak sayısı verilerini girdikten sonra sonlarına yıldız koymaktır. Yıldız, okuma işlemi yapılırken ilgili dizinin sonuna geldiğinin kanıtıdır. Ayrıca Excelde kaynak kullanım miktarı kolonu doldurulurken öncelikle 1. projenin 1. kaynak tipindeki kullanım miktarı akabinde ise 1. projenin 2. kaynak tipi kullanım miktarı sıralaması ile girilir. Bütün projeler için bu işlem tekrarlanır ve veri girişi tamamlanınca da sonuna belirtildiği gibi yıldız konulur. Veri girişleri tamamlanınca kullanıcının sadece programı çalıştırarak sonuç alması yeterlidir. Kullanıcının veri girişi yapacağı Excel dosyası Şekil 5.3'de belirtilmiştir.

Proje Sayısı	Kaynak Tipi Sayısı	Zaman (Gün)	Aktivite Sayıları	Kaynak Kapasitesi	Faaliyet Süreleri	Kaynak Kullanım Miktarı	Öncüllük İlişkileri
3	2	50	10	6	3	2	1,0,1
			6	8	5	3	1,0,2
			9	*	1	3	1,0,3
			*		3	1	1,1,4
					2	1	1,2,7
					4	2	1,2,5
					5	3	1,3,5
					6	1	1,4,6
					4	1	1,5,7
					4	1	1,5,8
					2	1	1,5,9
					4	2	1,6,10
					3	1	1,7,10
					5	2	1,8,11
					7	2	1,9,11
					6	2	1,10,11
					2	1	2,0,1

Şekil 5.3 Veri Giriş Dosyası

Geliştirilen yazılım; proje sayısı, kaynak tipi sayısı, aktivite sayısından vb. bağımsız olarak çalışacak şekilde geliştirilmiştir. Kullanıcı n proje sayısı, n kaynak tipi ve n aktivite sayısı için programı çalıştırarak OPL kodunun üretilmesini sağlayabilir. Bu sayede kullanıcılar farklı senaryoları deneyerek hangi sonucun kendileri için daha verimli olacağı sonucuna ulaşabilmektedirler.

Geliştirilen yazılımın doğruluğunu test etmek adına gerçek hayat problemi olan ve Bölüm 4.1 'de tanımlanan problemin verileri (proje sayısı, projelerdeki faaliyet sayısı, faaliyetlerin öncüllük ilişkileri, kaynak çeşidi sayısı, kaynak kapasitesi, faaliyetleri gerçekleştirmek için hangi kaynak çeşidinden kaç kişinin kullanılabilceği, zaman periyodu sayısı) Ek 10 'da belirtilen excele girilmiştir. Yazılım çalıştırıldığında ise daha önceden problemin çözümü için yazılan ve EK-8'de belirtilen kodun aynısı üretilmiştir. Böylece geliştirilen yazılımla, OPL kodunun yazılması için vakit harcanmayıp sonuca gitmek için daha hızlı aksiyon alınacak, kullanıcılar hazırlık aşamasından çok çözüme ve sonuçları yorumlamaya odaklanacaklardır. Java Programlama dilinde yazılan kod ise EK-11'de belirtilmiştir.

6. SONUÇ ve ÖNERİLER

Tez kapsamında, amaç fonksiyonu projelerin toplam tamamlanma süresini en küçükleyen çok projeli ve kaynak kısıtlı (yenilenebilir kaynak) proje çizelgeleme problemi üzerinde çalışılmıştır. Yazılım firmasındaki projeler incelenmiş ve bu projelerin, en kısa tamamlanma zamanları ve projelerin en kısa sürede tamamlanması için projedeki faaliyetlerin hangi t zamanlarda başlaması gerektiği çözüm olarak verilmiştir.

Firmada kullanımı kolaylaştırmak içinde problemin verilerinin yazıldığı Excel dosyasını girdi olarak alıp bu Excel dosyasındaki veriyi okuyup, okunan veriyi OPL'de kullanılan koda çeviren Java Uygulaması ile bir yazılım geliştirilmiştir.

Sonuç olarak, tez çalışmasındaki amaç kaynak kısıtlı projelerin çizelgenerek en kısa tamamlanma sürelerinin bulunması olarak tanımlanabilir.

Anlatılan tez çalışmasında kaynakların kapasiteleri birim zamanda sabit alınarak hesaplamalar yapılmıştır. Her faaliyet için kaynak kapasitesinin zamana bağlı değişimi göz önünde bulundurulmamıştır. İleriki çalışmalarda bu durum da göz önünde bulundurularak çalışma detaylandırılabilir.

Tez çalışmasındaki model $T=0$ zaman diliminden başlayarak projelerin çizelgelemesini sağlamaktadır. Beş projenin başladığı ve $T=10$ zaman diliminde iki tane yeni proje alındığı ve kaynak kapasitelerinin de değiştiği durumlar göz önünde bulundurularak, doğrusal karar modeli detaylandırılabilir. JAVA uygulaması kullanılarak geliştirilen ara yüz genişletilerek ileriki dönemlerde firmaların daha aktif kullanacağı proje takip sistemine dönüştürülebilir.

Ek olarak tez çalışmasında projelerin maliyetleri hesaba katılmamıştır. Üst yönetimin verdiği bazı stratejik kararlarda oluşabilecek maliyetler dikkate alınmamaktadır. Firmanın zamanında projeleri teslim etmesi için oluşabilecek ekstra maliyetler göz ardı edilir. Örneğin, gecikmede olan bir proje olduğu varsayılın; firma genel müdürü, projede gecikme yaşanmaması için oluşabilecek bütün ek maliyetleri kabul edip kaynak alınması ya da bazı iş paketlerinin alt yükleniciye verilmesi yönünde karar verebilir. Bu sayede firma güvenilirliğini sağlayarak sektörde de itibarını sarsmamış olur. Özetle, maliyetlerin dikkate

alınmadığı çok projeli aynı kaynak havuzundan beslenen kaynak kısıtlı projelerin çizelgelemesinde firmaya önerilen model kullanılabilir.

Maliyetlerin göz önünde bulundurulması gereken durumlarda ise amaç fonksiyonu en az maliyet ve en kısa sürede projelerin tamamlanmasını sağlayan bir model geliştirilebilir. Örneğin, yöneticiler projelerin belirlenen zaman periyodunda tamamlanması için kaynak sayısını arttırdıklarında projelerde oluşacak ek maliyetleri görerek karar vermek isteyebilirler.

Son olarak; tez çalışması kapsamında TA ve TB olmak üzere 2 çeşit yenilenebilir kaynak tipi bulunmakta olup kaynaklar projelere dedike çalışmaktadırlar. Başka bir tez çalışmasında ise kaynakların dedike çalışmadığı yani matris organizasyon yapısını yansıtan bir model kurulabilir. Matris organizasyon yapısında bir kaynak birden fazla projelere gün içerisinde çalışabilmektedir. Örneğin, TA kaynak tipi bir günde yüzde 20 A projesi, yüzde 30 B projesi, yüzde 10 C projesi, yüzde 40 D projesinde çalışabilmektedir. Matris organizasyon yapısı ile yönetilen firmalar için de matematiksel model kurularak sonuçları incelenebilir. Ayrıca boş kaynakların tespiti için model değiştirilebilir.

KAYNAKLAR LİSTESİ

- [1] Albayrak, B., Proje Yönetimi ve Proje Danışmanlığı, Birinci Baskı, İstanbul: Beta Yayınevi, s.306, 2001.
- [2] Bala, H., A Comparative Study Of Computational Procedures For Resource Constrained Project Scheduling Problem, Bilkent Üniversitesi Yüksek Lisans Tezi, Ankara,1991.
- [3] Balkaya, H., Kaynak Kısıtlı Proje Çizelgeleme Problemlerinin Genetik Algoritma Yaklaşımıyla Optimizasyonu, Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü Ekonometri Ana Bilim Dalı Yüksek Lisans Tezi, İzmir, 2011.
- [4] Bouleimen, K., ve Lecocq H., A new efficient simulated annealing algorithm for the resource-constrained Project scheduling problem and its multiple mode version, ELSEVIER European Journal of Operational Research,149:268-281,2003.
- [5] Browning, T., ve Yassine, A., Resource-constrained multi-project scheduling: Priority rule performance revised, ELSEVIER Int. J. Production
- [6] Dean, B.V., ve Denzler, D.R., Watkins, J.J., Multiproject staff scheduling with variable resource constraint, IEEE Transactions on Engineering Management, 39, 59-72., 1992.
- [7] Demeulemeester, E. and Herroelen, W., New Benchmark Results for the Resource-Constrained Project Scheduling Problem, Management Science, 43:1485-1492, 1997.
- [8] Doğruer, M., Proje Yönetimi, İstanbul: Açılım Kitap,2007.
- [9] Dorndorf U., Pesch E. ve Phan-Huy T., A Time-Oriented Branch-and Bound Algorithm for Resource-Constrained Project Scheduling with Generalised Precedence Constraints, Management Science, 46:1365-1384, 2000.
- [10] Erdal, M., Kısıtlı Kaynak Koşullarında Yapı Projelerinin Genetik Algoritma ile Programlanması, Gazi Üniversitesi Fen Bilimleri Enstitüsü Yapı Eğitimi Bölümü Doktora Tezi, Ankara, 2007.
- [11] Talbot, F. B., Resource Constrained Project Scheduling with Time-Resource Tradeoffs, The Nonpreemptive Case, Management Science, 28 (10), 1197-1210, 1982.
- [12] Kılıç, M., Multiobjective Genetic Algorithm Approached to Project Scheduling Under Risk, Sabancı Üniversitesi Mühendislik ve Doğa Bilimleri Fakültesi Yüksek Lisans Tezi, İstanbul,2003.
- [13] Kone, O., Artigues, C., Lopez, P., ve Mongeau, M., Event-based Mixed Integer Linear Programming Models for resource-constrained project scheduling problems, HAL archives-ouvertes, 2009.

- [14] Lawrence, S.R., Morton, T.E., Resource-constrained multi-project scheduling with tardy costs: Comparing myopic, bottleneck and resource pricing heuristics, *European Journal of Operational Research*, 64, 168-187, 1993.
- [15] Lova, A., ve Tormos, P., Analysis of scheduling schemes and heuristic rules performance in resource-constrained multiproject scheduling, *Annals of Operations Research*, 2000.
- [16] Özdemir, G., Kısıtlı Kaynaklarla Proje Çizelgelemesi Problemlerinde Kullanılan Genetik Algoritma Metodları ve Bunların Karşılaştırılması, Ankara Üniversitesi Sosyal Bilimler Enstitüsü Ana Bilim Dalı Yüksek Lisans Tezi, Ankara, 2006.
- [17] Pritsker A.A.B., Watters L.J., ve Wolfe P.M., Multiproject scheduling with limited resources: a zero-one programming approach, *Management Science* 16:93–107, 1969.
- [18] Reyck B.D. ve Herroelen W., A Branch-and-Bound Procedure for the Resource-Constrained Project Scheduling Problem with Generalized Precedence Relations, *Econ Papers*, 1996.
- [19] Slowinski, R., Multiobjective network scheduling with efficient use of renewable and nonrenewable resources, *European Journal of Operational Research*, 7, 265-273, 1981.
- [20] Slowinski, R., Multiobjective project scheduling under multiple category resource constraints, 151-167, 1989.
- [21] Şerifoğlu, S., ve Ulusoy, Ş., Kaynak Kısıtlı proje çizelgelemede indirgenmiş nakit akışı maksimizasyonu için genetik algoritma yaklaşımı, *Endüstri Mühendisliği Dergisi*, 2000.
- [22] Tekel, S. , *Modern Proje Yönetimi Kimin İcadı*, 2014.
- [23] Tsubakitani, S., ve Deckro, R.F., A heuristic for multi-project scheduling with limited resources in the housing industry, *European Journal of Operational Research*, 49, 80-91, 1990.
- [24] Ulusoy, G., *Proje Planlamada Kaynak Kısıtlı Çizelgeleme*, Mühendislik ve Doğa Bilimleri Fakültesi, Sabancı Üniversitesi, İstanbul, 2002.

EKLER LİSTESİ

	<u>Sayfa</u>
EK 1 Örnek Uygulama Opl Kodu.....	47
EK 2 Örnek Uygulama Cplex Çözümü	50
EK 3 A Projesi Faaliyet Ağ Diyagramı	53
EK 4 B Projesi Faaliyet Ağ Diyagramı	54
EK 5 C Projesi Faaliyet Ağ Diyagramı	55
EK 6 D Projesi Faaliyet Ağ Diyagramı	56
EK 7 E Projesi Faaliyet Ağ Diyagramı	57
EK 8 Gerçek Hayat Probleminin OPL Kodu	58
EK 9 Gerçek Hayat Probleminin Cplex Çözümü	67
EK 10 Gerçek Hayat Probleminin Verileri için Hazırlanan Dosya.....	71
EK 11 Java Kodu.....	76

EK 1 Örnek Uygulama Opl Kodu

```
using CPLEX;

int P = 3;
int R = 2;
int T = 50;
int A[1..P] = [10,6,9];
int C[1..R] = [6,8];

int d[1..P][0..(max(p in 1..P) A[p])+1] = [
    [0,3,5,1,3,2,4,5,6,4,4,0], //P=1 d[1][0] = d[1][(max(p in 1..P) A[p])] = 0
    [0,2,4,3,5,7,6,0,0,0,0,0], //P=2 d[2][0] = d[2][(max(p in 1..P) A[p])] = 0
    [0,2,4,6,8,3,5,7,9,3,0,0] //P=3 d[3][0] = d[3][(max(p in 1..P) A[p])] = 0
];

int u[1..R][1..P][0..(max(p in 1..P) A[p])+1] = [
    [//R=1.....
    [0,2,3,3,1,1,2,3,1,1,1,0], //P=1 r[1][1][0] = r[1][1][(max(p in 1..P) A[p])] = 0
    [0,1,2,1,2,2,2,0,0,0,0,0], //P=2 r[1][2][0] = r[1][2][(max(p in 1..P) A[p])] = 0
    [0,1,2,3,1,1,1,2,3,1,0,0] //P=3 r[1][3][0] = r[1][3][(max(p in 1..P) A[p])] = 0
    ],
    [//R=2.....
    [0,1,2,2,0,0,1,4,2,2,2,0], //P=1 r[2][1][0] = r[2][1][(max(p in 1..P) A[p])] = 0
    [0,2,3,0,0,2,1,0,0,0,0,0], //P=2 r[2][2][0] = r[2][2][(max(p in 1..P) A[p])] = 0
    [0,2,3,3,3,2,2,3,3,1,0,0] //P=3 r[2][3][0] = r[2][3][(max(p in 1..P) A[p])] = 0
    ]
];

tuple PredSucc(int p;int i;int j;);
{PredSucc} S = { <1,0,1>,
                <1,0,2>,
                <1,0,3>,
                <1,1,4>,
                <1,2,7>,
                <1,2,5>,
                <1,3,5>,
                <1,4,6>,
                <1,5,7>,
                <1,5,8>,
                <1,5,9>,
                <1,6,10>,
                <1,7,10>,
                <1,8,11>,
                <1,9,11>,
                <1,10,11>};
```

```

    <2,0,1>,
    <2,0,2>,
    <2,1,4>,
    <2,2,3>,
    <2,4,5>,
    <2,3,5>,
    <2,5,6>,
    <2,6,11>,

    <3,0,1>,
    <3,1,2>,
    <3,1,3>,
    <3,1,4>,
    <3,2,5>,
    <3,3,6>,
    <3,4,7>,
    <3,5,8>,
    <3,6,8>,
    <3,7,8>,
    <3,8,9>,
    <3,9,11>
};

```

```

dvar boolean x[0..(max(p in 1..P) A[p])+1][1..P][0..T];
dvar float+ ProjectCompletionTime;

minimize
    ProjectCompletionTime;

subject to
{

forall(p in 1..P, t in 0..T)
    t*x[(max(p in 1..P) A[p])+1][p][t] <= ProjectCompletionTime;

forall(p in 1..P)
    sum(t in 0..T)
        x[(max(p in 1..P) A[p])+1,p,t] == 1;

forall(p in 1..P,i in 1..A[p])
    sum(t in 0..T)
        x[i,p,t] == 1;

forall(s in S)
    sum(t in 0..T)t*x[s.j,s.p,t]-
    sum(t in 0..T)t*x[s.i,s.p,t] >= d[s.p,s.i];

forall(t in 0..T,r in 1..R)
    sum(p in 1..P, i in 1..A[p])u[r,p,i]*

```

```

    sum(z in max1((t-d[p,i]+1),0)..t)x[i,p,z] <= C[r];
}

execute PrintSolution
{
  writeln("Solution");
  writeln("-----");
  for(var t=0;t<=T;t++)
    for(var i=0;i<=I;i++)
      for(var p=1;p<=P;p++)
        if(x[i][p][t]==1)
          writeln( "X[" , i , "," , p , "," , t , "]" = " , x[i][p][t] );
}

```

EK 2 Örnek Uygulama Cplex Çözümü

Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\pelin>cd C:\Users\pelin\Desktop\PELİN TEZ EKLER

C:\Users\pelin\Desktop\PELİN TEZ EKLER>oplrn EK_1.txt

<<< setup

<<< generate

Tried aggregator 1 time.
MIP Presolve eliminated 29 rows and 581 columns.
Reduced MIP has 290 rows, 1001 columns, and 10336 nonzeros.
Reduced MIP has 1000 binaries, 0 generals, 0 SOSs, and 0 indicators.
Presolve time = 0.20 sec. (27.30 ticks)
Probing fixed 118 vars, tightened 0 bounds.
Probing time = 0.02 sec. (3.70 ticks)
Cover probing fixed 2 vars, tightened 98 bounds.
Tried aggregator 1 time.
MIP Presolve eliminated 54 rows and 136 columns.
MIP Presolve modified 1711 coefficients.
Reduced MIP has 236 rows, 865 columns, and 8976 nonzeros.
Reduced MIP has 864 binaries, 1 generals, 0 SOSs, and 0 indicators.
Presolve time = 0.02 sec. (5.87 ticks)
Probing time = 0.00 sec. (2.79 ticks)
Cover probing fixed 1 vars, tightened 0 bounds.
Tried aggregator 1 time.
MIP Presolve eliminated 6 rows and 8 columns.
MIP Presolve modified 156 coefficients.
Reduced MIP has 230 rows, 857 columns, and 8913 nonzeros.
Reduced MIP has 856 binaries, 1 generals, 0 SOSs, and 0 indicators.
Presolve time = 0.02 sec. (6.46 ticks)
Probing time = 0.00 sec. (2.77 ticks)
Cover probing fixed 0 vars, tightened 1 bounds.
Clique table members: 5517.
MIP emphasis: balance optimality and feasibility.
MIP search method: dynamic search.
Parallel mode: deterministic, using up to 4 threads.

Root relaxation solution time = 0.00 sec. (2.78 ticks)

Nodes		Objective	IInf	Best Integer	Cuts/		Gap
Node	Left				Best Bound	ItCnt	
	0	29.0000	56		29.0000	70	
*	0+			50.0000	29.0000		42.00%
	0	29.0000	55	50.0000	Cuts: 48	127	42.00%
	0	29.0000	68	50.0000	Cuts: 103	240	42.00%
	0	29.0000	39	50.0000	Cuts: 10	299	42.00%
	0	29.0000	51	50.0000	Cuts: 82	370	42.00%
*	0+			42.0000	29.0000		30.95%
	0	29.0000	27	42.0000	29.0000	370	30.95%

Elapsed time = 7.19 sec. (3502.07 ticks, tree = 0.01 MB, solutions = 8)

3769	40	34.0000	36	35.0000	34.0000	165327	2.86%
3773	38	34.0000	30	35.0000	34.0000	165594	2.86%
3775	38	34.0000	65	35.0000	34.0000	166057	2.86%
3777	36	infeasible		35.0000	34.0000	166341	2.86%
3779	36	34.0000	64	35.0000	34.0000	166700	2.86%
3780	35	infeasible		35.0000	34.0000	166872	2.86%
3781	36	34.0000	32	35.0000	34.0000	166900	2.86%
3782	35	cutoff		35.0000	34.0000	167270	2.86%
3783	36	34.0000	63	35.0000	34.0000	167271	2.86%
3784	37	34.0000	37	35.0000	34.0000	167377	2.86%

Elapsed time = 9.30 sec. (4466.32 ticks, tree = 0.01 MB, solutions = 8)

GUB cover cuts applied: 50

Clique cuts applied: 24

Cover cuts applied: 11

Implied bound cuts applied: 4

Flow cuts applied: 1

Mixed integer rounding cuts applied: 17

Zero-half cuts applied: 8

Lift and project cuts applied: 1

Gomory fractional cuts applied: 1

Root node processing (before b&c):

Real time = 0.84 sec. (226.88 ticks)

Parallel b&c, 4 threads:

Real time = 10.00 sec. (4939.63 ticks)

Sync time (average) = 2.17 sec.

Wait time (average) = 2.28 sec.

Total (root+branch&cut) = 10.84 sec. (5166.51 ticks)

```
<<< solve
```

```
OBJECTIVE: 35
```

```
Solution
```

```
-----
```

```
X[1,3,0] = 1
```

```
X[2,2,0] = 1
```

```
X[3,1,0] = 1
```

```
X[2,1,1] = 1
```

```
X[4,3,2] = 1
```

```
X[1,1,4] = 1
```

```
X[3,3,6] = 1
```

```
X[4,1,7] = 1
```

```
X[1,2,8] = 1
```

```
X[3,2,10] = 1
```

```
X[4,2,10] = 1
```

```
X[7,3,12] = 1
```

```
X[2,3,13] = 1
```

```
X[5,2,15] = 1
```

```
X[5,1,17] = 1
```

```
X[6,3,17] = 1
```

```
X[5,3,19] = 1
```

```
X[6,1,19] = 1
```

```
X[8,3,22] = 1
```

```
X[7,1,23] = 1
```

```
X[6,2,28] = 1
```

```
X[8,1,28] = 1
```

```
X[9,1,31] = 1
```

```
X[9,3,31] = 1
```

```
X[10,1,31] = 1
```

```
X[11,2,34] = 1
```

```
X[11,3,34] = 1
```

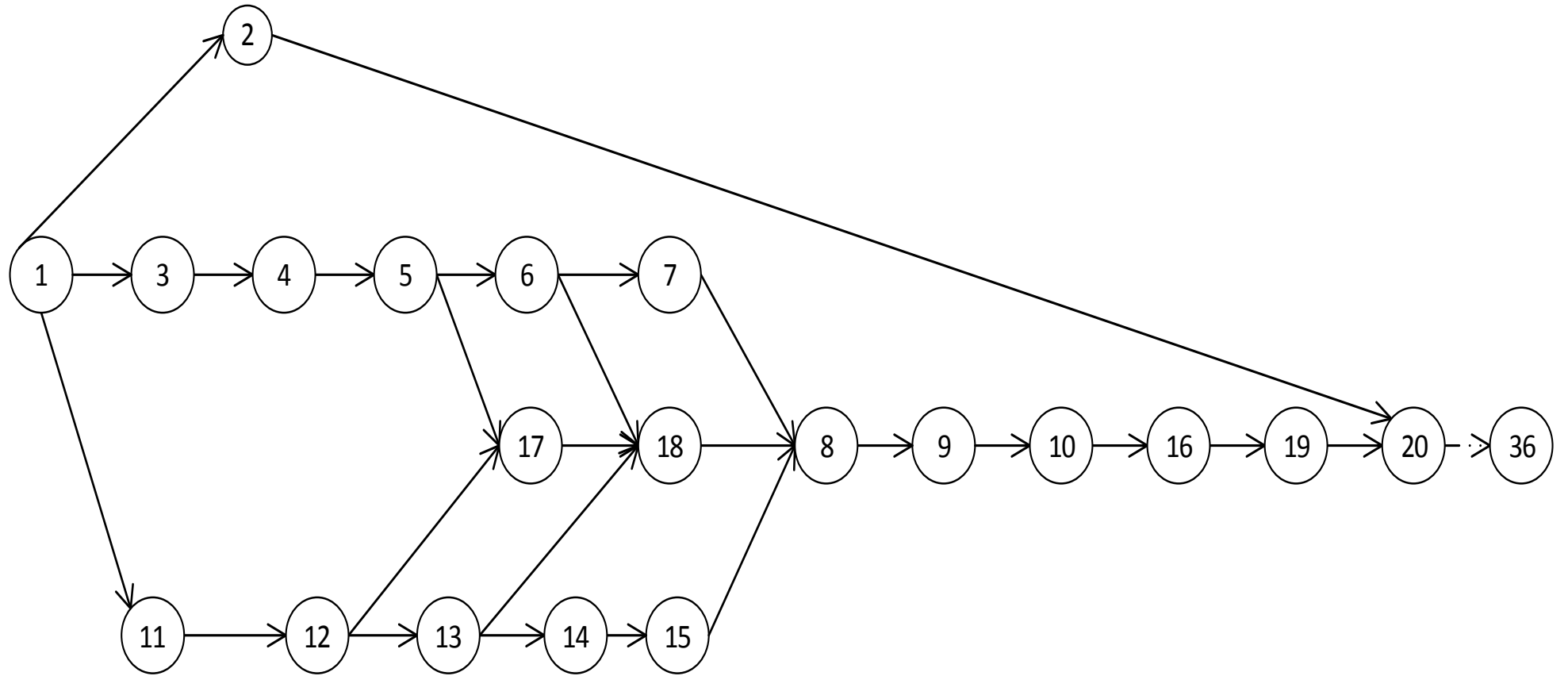
```
X[11,1,35] = 1
```

```
<<< post process
```

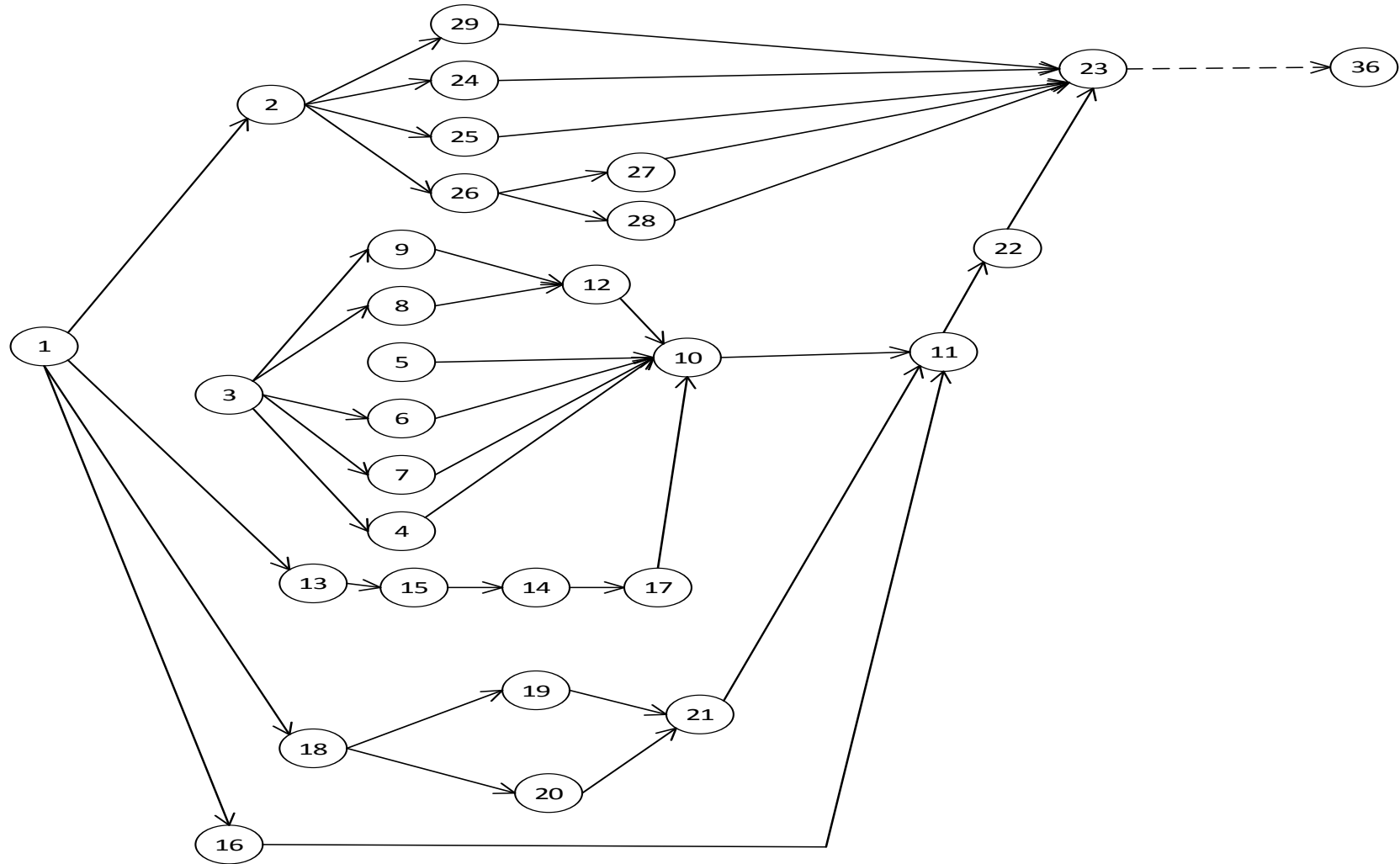
```
C:\Users\pelin\Desktop\PELİN TEZ EKLER>
```

```
<<< done
```

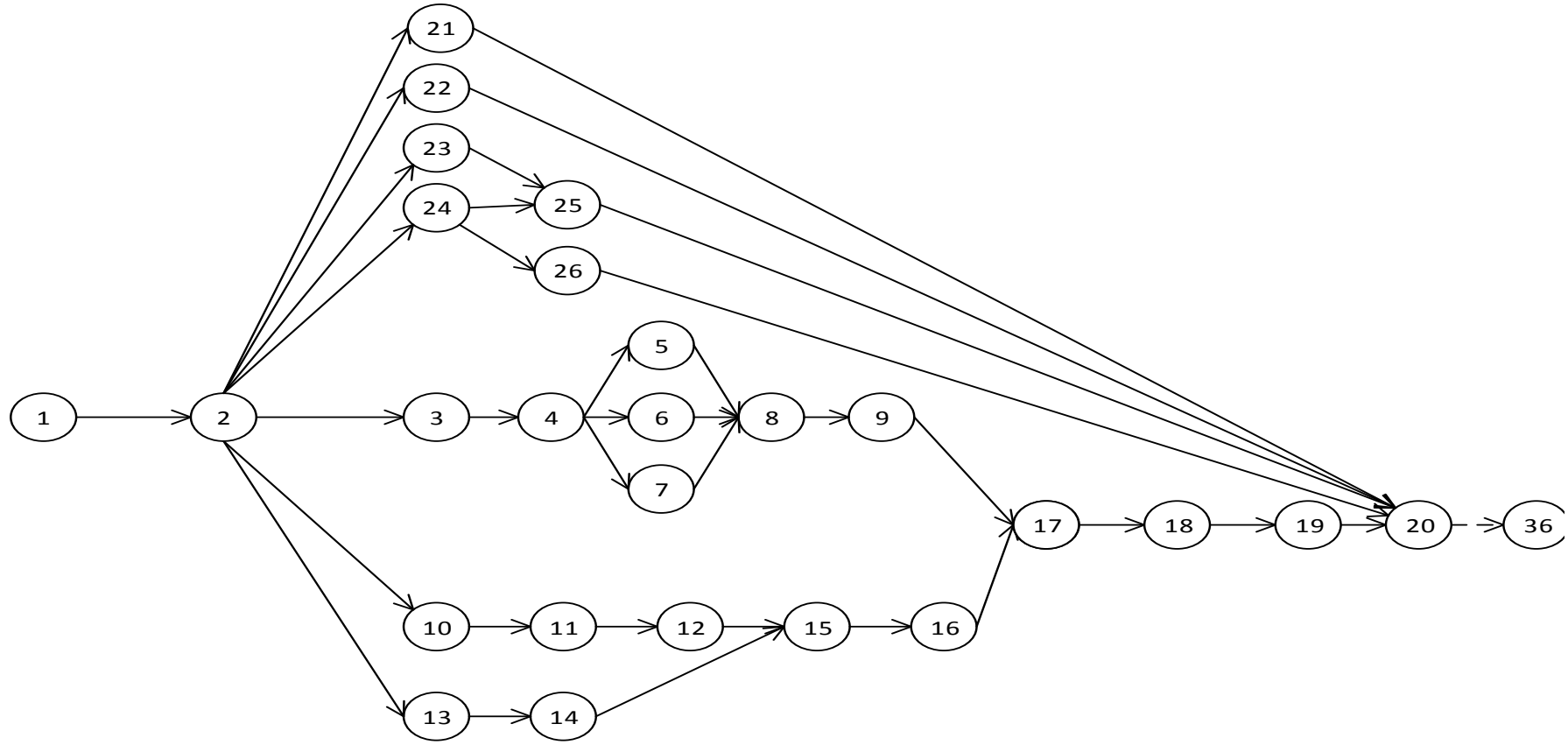

EK 3 A Projesi Faaliyet Ağ Diyagramı



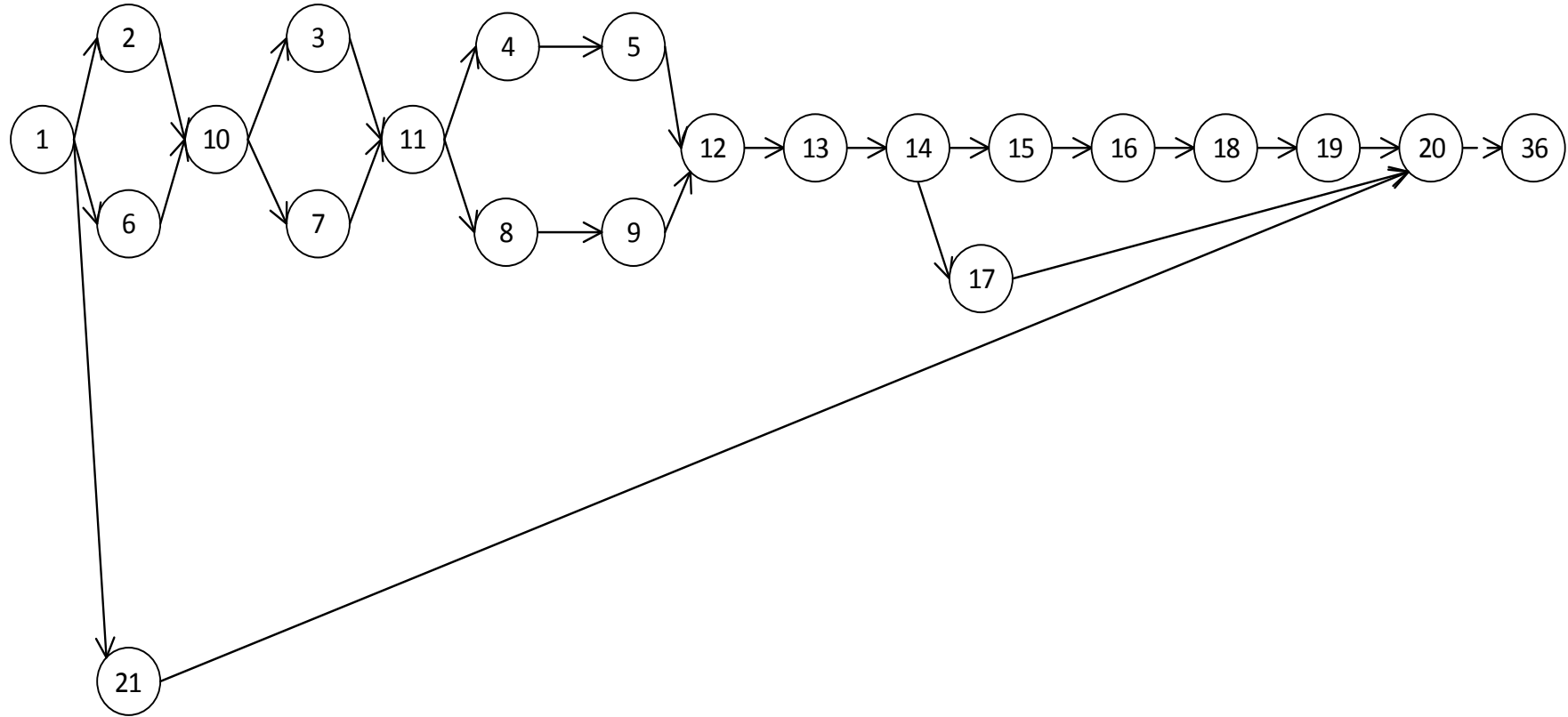
EK 4 B Projesi Faaliyet Ağ Diyagramı



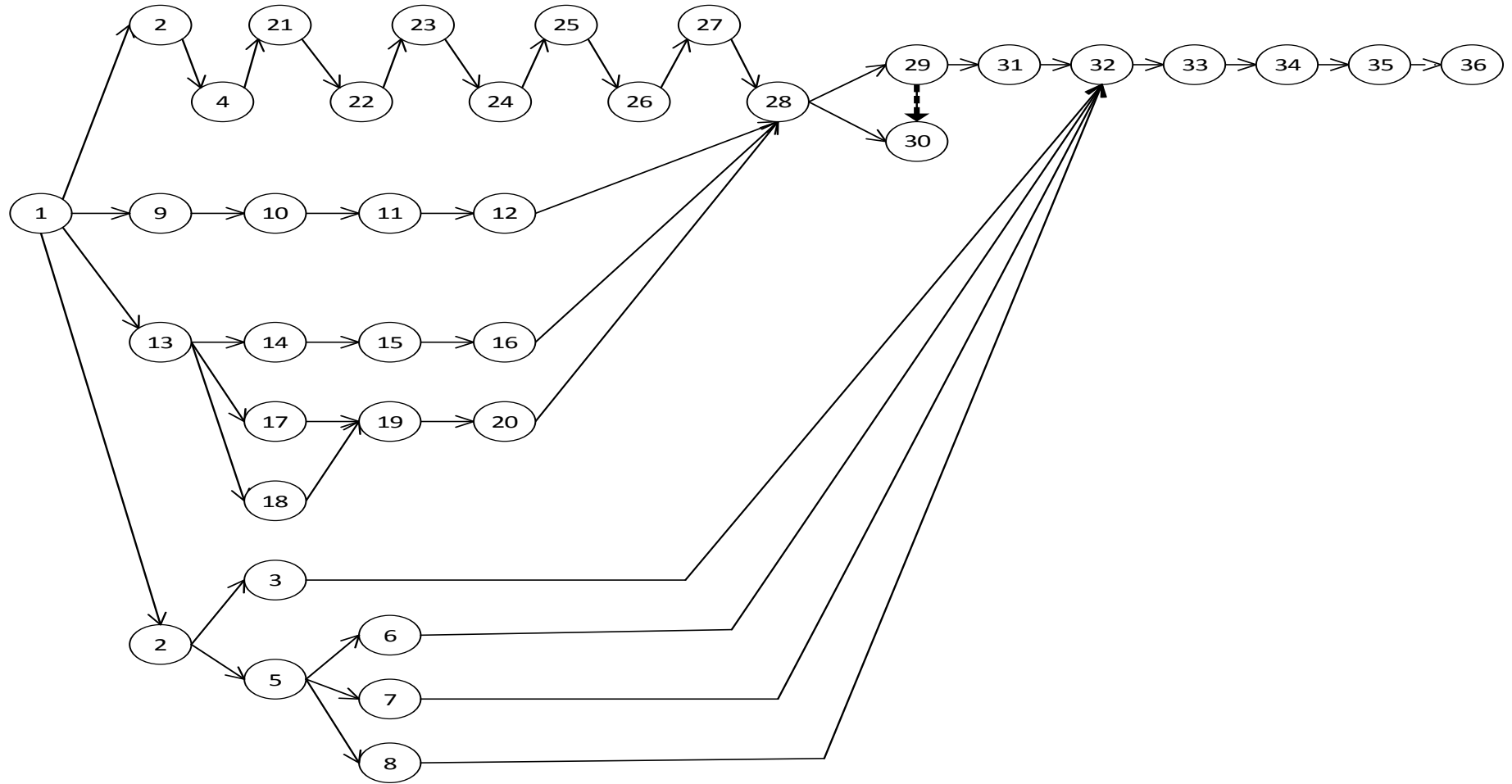
EK 5 C Projesi Faaliyet Ağ Diyagramı



EK 6 D Projesi Faaliyet Ağ Diyagramı



EK 7 E Projesi Faaliyet Ağ Diyagramı




```
{PredSucc} S = { <1,0,1>,
                  <1,1,2>,
                  <1,1,3>,
                  <1,1,4>,
                  <1,1,11>,
                  <1,2,20>,
                  <1,3,4>,
                  <1,4,5>,
                  <1,5,6>,
                  <1,5,17>,
                  <1,6,7>,
                  <1,6,18>,
                  <1,17,18>,
                  <1,18,8>,
                  <1,7,8>,
                  <1,11,12>,
                  <1,2,20>,
                  <1,3,4>,
                  <1,4,5>,
                  <1,5,6>,
                  <1,5,17>,
                  <1,6,7>,
                  <1,6,18>,
                  <1,17,18>,
                  <1,18,8>,
                  <1,7,8>,
                  <1,11,12>,
                  <1,12,17>,
                  <1,12,13>,
                  <1,13,14>,
                  <1,13,18>,
                  <1,14,15>,
                  <1,15,8>,
```

<1, 8, 9>,
<1, 9, 10>,
<1, 10, 16>,
<1, 16, 19>,
<1, 19, 20>,
<1, 20, 36>,

<2, 0, 1>,
<2, 1, 2>,
<2, 1, 3>,
<2, 2, 29>,
<2, 2, 24>,
<2, 2, 25>,
<2, 5, 26>,
<2, 26, 27>,
<2, 26, 28>,
<2, 29, 23>,
<2, 24, 23>,
<2, 25, 23>,
<2, 27, 23>,
<2, 28, 23>,
<2, 3, 9>,
<2, 3, 8>,
<2, 9, 12>,
<2, 8, 12>,
<2, 12, 10>,
<2, 3, 5>,
<2, 3, 6>,
<2, 3, 7>,
<2, 3, 4>,
<2, 5, 10>,
<2, 6, 10>,
<2, 7, 10>,

<2,4,10>,
<2,1,13>,
<2,13,15>,
<2,15,14>,
<2,14,17>,
<2,17,10>,
<2,10,11>,
<2,11,22>,
<2,22,23>,
<2,1,18>,
<2,18,19>,
<2,18,20>,
<2,19,21>,
<2,20,21>,
<2,21,11>,
<2,1,16>,
<2,16,11>,
<2,23,36>,

<3,0,1>,
<3,1,2>,
<3,2,24>,
<3,24,26>,
<3,24,25>,
<3,2,22>,
<3,2,23>,
<3,2,21>,
<3,26,20>,
<3,25,20>,
<3,22,20>,
<3,23,20>,
<3,21,20>,
<3,2,3>,

<3, 3, 4>,
<3, 4, 5>,
<3, 4, 6>,
<3, 4, 7>,
<3, 5, 8>,
<3, 6, 8>,
<3, 7, 8>,
<3, 8, 9>,
<3, 9, 17>,
<3, 2, 10>,
<3, 10, 11>,
<3, 11, 12>,
<3, 12, 15>,
<3, 15, 16>,
<3, 2, 13>,
<3, 13, 14>,
<3, 14, 15>,
<3, 15, 16>,
<3, 16, 17>,
<3, 17, 18>,
<3, 18, 19>,
<3, 19, 20>,
<3, 20, 36>,

<4, 0, 1>,
<4, 1, 2>,
<4, 1, 6>,
<4, 2, 10>,
<4, 6, 10>,
<4, 10, 3>,
<4, 10, 7>,
<4, 3, 11>,
<4, 7, 11>,

<4, 11, 4>,
<4, 4, 5>,
<4, 11, 8>,
<4, 8, 9>,
<4, 5, 12>,
<4, 9, 12>,
<4, 12, 13>,
<4, 13, 14>,
<4, 14, 15>,
<4, 14, 17>,
<4, 17, 20>,
<4, 15, 16>,
<4, 16, 18>,
<4, 18, 19>,
<4, 19, 20>,
<4, 1, 21>,
<4, 21, 20>,
<4, 20, 36>,

<5, 0, 1>,
<5, 1, 2>,
<5, 2, 4>,
<5, 4, 21>,
<5, 21, 22>,
<5, 23, 24>,
<5, 24, 25>,
<5, 25, 26>,
<5, 26, 27>,
<5, 27, 28>,
<5, 28, 29>,
<5, 28, 30>,
<5, 29, 30>,
<5, 29, 31>,

<5, 31, 32>,
<5, 32, 33>,
<5, 33, 34>,
<5, 34, 35>,
<5, 1, 9>,
<5, 9, 10>,
<5, 10, 11>,
<5, 11, 12>,
<5, 12, 28>,
<5, 1, 13>,
<5, 13, 14>,
<5, 14, 15>,
<5, 15, 16>,
<5, 16, 28>,
<5, 13, 17>,
<5, 13, 18>,
<5, 17, 19>,
<5, 18, 19>,
<5, 19, 20>,
<5, 20, 28>,
<5, 1, 2>,
<5, 2, 3>,
<5, 3, 32>,
<5, 2, 5>,
<5, 5, 6>,
<5, 5, 7>,
<5, 5, 8>,
<5, 6, 32>,
<5, 7, 32>,
<5, 8, 32>,
<5, 35, 36>

};

```

dvar boolean x[0..(max(p in 1..P) A[p])+1][1..P][0..T];
dvar float+ ProjectCompletionTime;

minimize
  ProjectCompletionTime;

subject to
{

forall(p in 1..P, t in 0..T)
  t*x[(max(p in 1..P) A[p])+1][p][t] <= ProjectCompletionTime;

forall(p in 1..P)
  sum(t in 0..T)
    x[(max(p in 1..P) A[p])+1,p,t] == 1;

forall(p in 1..P,i in 1..A[p])
  sum(t in 0..T)
    x[i,p,t] == 1;

forall(s in S)
  sum(t in 0..T)t*x[s.j,s.p,t]-
  sum(t in 0..T)t*x[s.i,s.p,t] >= d[s.p,s.i];

forall(t in 0..T,r in 1..R)
  sum(p in 1..P, i in 1..A[p])u[r,p,i]*
  sum(z in max(1-(t-d[p,i]+1),0)..t)x[i,p,z] <= C[r];
}

execute PrintSolution
{
  writeln("Solution");
  writeln("-----");
}

```

```
for(var t=0;t<=T;t++)
  for(var i=0;i<=11;i++)
    for(var p=1;p<=P;p++)
      if(x[i][p][t]==1)
        writeln( "X[" , i , "," , p , "," , t , "] = " , x[i][p][t] );
```

EK 9 Gerçek Hayat Probleminin Cplex Çözümü

```
6638  324      cutoff          490.0000    456.0000  718217    6.94%
 6639  325      487.0000    155      490.0000    456.0000  718220    6.94%
 6640  326      456.0000    143      490.0000    456.0000  718221    6.94%
 6641  325      infeasible          490.0000    456.0000  718225    6.94%
Elapsed time = 4156.83 sec. (2477440.37 ticks, tree = 0.03 MB, solutions = 23)
* 9216+  82          456.0000    456.0000          0.00%
 9258  83      456.0000    106      456.0000    456.0000  724055    0.00%
 9259  82      infeasible          456.0000    456.0000  724058    0.00%
 9260  83      456.0000    106      456.0000    456.0000  724059    0.00%
 9261  82      infeasible          456.0000    456.0000  724062    0.00%
 9262  83      456.0000    106      456.0000    456.0000  724063    0.00%
 9263  82      infeasible          456.0000    456.0000  724066    0.00%
 9264  83      456.0000    106      456.0000    456.0000  724067    0.00%
 9265  82      infeasible          456.0000    456.0000  724070    0.00%
 9266  83      456.0000    106      456.0000    456.0000  724071    0.00%
 9267  82      infeasible          456.0000    456.0000  724074    0.00%
Elapsed time = 4168.19 sec. (2481648.26 ticks, tree = 0.03 MB, solutions = 23)
```

GUB cover cuts applied: 16

Clique cuts applied: 72

Cover cuts applied: 78

Flow cuts applied: 7

Mixed integer rounding cuts applied: 75

Zero-half cuts applied: 32

Gomory fractional cuts applied: 64

Root node processing (before b&c):

Real time = 358.74 sec. (188565.12 ticks)

Parallel b&c, 2 threads:

Real time = 3809.93 sec. (2293623.08 ticks)

```
Sync time (average) = 635.62 sec.  
Wait time (average) = 635.97 sec.  
-----  
Total (root+branch&cut) = 4168.68 sec. (2482188.20 ticks)
```

```
<<< solve
```

```
OBJECTIVE: 456
```

```
Solution
```

```
-----
```

```
X[1,1,0] = 1  
X[1,2,0] = 1  
X[1,3,0] = 1  
X[1,4,0] = 1  
X[1,5,0] = 1  
X[2,2,1] = 1  
X[2,3,1] = 1  
X[2,4,1] = 1  
X[2,5,1] = 1  
X[3,2,1] = 1  
X[6,4,1] = 1  
X[9,5,1] = 1  
X[11,1,4] = 1  
X[5,2,41] = 1  
X[8,2,41] = 1  
X[9,2,41] = 1  
X[10,4,52] = 1  
X[7,2,58] = 1  
X[3,1,73] = 1  
X[3,3,81] = 1  
X[5,5,81] = 1  
X[10,3,81] = 1
```


X[10,5,81] = 1
X[3,5,82] = 1
X[11,3,86] = 1
X[7,4,99] = 1
X[3,4,103] = 1
X[4,2,137] = 1
X[4,5,144] = 1
X[11,4,150] = 1
X[4,3,161] = 1
X[6,5,161] = 1
X[7,5,161] = 1
X[8,5,161] = 1
X[11,5,189] = 1
X[4,4,200] = 1
X[8,4,200] = 1
X[5,3,231] = 1
X[6,3,231] = 1
X[7,3,232] = 1
X[4,1,289] = 1
X[5,4,293] = 1
X[8,3,296] = 1
X[9,4,296] = 1
X[5,1,332] = 1
X[9,3,346] = 1
X[6,1,351] = 1
X[6,2,353] = 1
X[10,2,378] = 1
X[2,1,399] = 1
X[7,1,402] = 1
X[11,2,413] = 1
X[8,1,418] = 1
X[9,1,422] = 1
X[10,1,425] = 1

<<< post process

<<< done

C:\Users\Onur\Desktop\pelin tez>

EK 10 Gerçek Hayat Probleminin Verileri için Hazırlanan Dosya

Zaman (Gün)	Aktivite sayıları	Kaynak Kapasitesi	Aktivite Süreleri	Kaynak Kullanım Miktarı	Öncüllük İlişkileri
700	20	50	1	3	1,0,1
	29	45	50	4	1,1,2
	26	*	60	5	1,1,3
	20		35	4	1,1,11
	35		10	1	1,2,20
	*		45	6	1,3,4
			15	2	1,4,5
			4	1	1,5,6
			2	4	1,5,17
			1	1	1,6,7
			60	5	1,6,18
			35	4	1,17,18
			10	1	1,18,8
			45	6	1,7,8
			15	2	1,11,12
			10	3	1,2,20
			1	2	1,12,17
			1	2	1,12,13
			5	3	1,13,14
			1	4	1,13,18
			1	2	1,14,15
			40	3	1,15,8
			40	6	1,8,9
			45	5	1,9,10
			35	1	1,10,16
			25	4	1,16,19
			40	3	1,19,20
			20	2	1,20,36
			25	4	2,0,1
			30	3	2,1,2
			35	6	2,1,3
			20	2	2,2,29
			50	2	2,2,24
			50	2	2,2,25
			30	5	2,5,26
			60	2	2,26,27
			90	6	2,26,28
			65	4	2,29,23
			70	2	2,24,23
			80	3	2,25,23
			20	5	2,27,23

Zaman (Gün)	Aktivite sayıları	Kaynak Kapasitesi	Aktivite Süreleri	Kaynak Kullanım Miktarı	Öncüllük İlişkileri
			3	4	2,28,23
			1	1	2,3,9
			90	3	2,3,8
			90	3	2,9,12
			70	3	2,8,12
			70	3	2,12,10
			70	3	2,3,5
			25	3	2,3,6
			1	1	2,3,7
			3	80	2,3,4
			80	5	2,5,10
			70	5	2,6,10
			65	6	2,7,10
			45	4	2,4,10
			55	5	2,1,13
			50	4	2,13,15
			30	3	2,15,14
			5	1	2,14,17
			7	3	2,17,10
			5	2	2,10,11
			4	1	2,11,22
			10	2	2,22,23
			20	4	2,1,18
			15	2	2,18,19
			5	2	2,18,20
			10	5	2,19,21
			15	4	2,20,21
			50	1	2,21,11
			10	3	2,1,16
			80	4	2,16,11
			80	4	2,23,36
			80	4	3,0,1
			80	3	3,1,2
			80	2	3,2,24
			1	1	3,24,26
			50	3	3,24,25
			45	3	3,2,22
			93	6	3,2,23
			20	2	3,2,21
			50	3	3,26,20
			45	3	3,25,20
			95	6	3,22,20
			20	2	3,23,20

Zaman (Gün)	Aktivite sayıları	Kaynak Kapasitesi	Aktivite Süreleri	Kaynak Kullanım Miktarı	Öncüllük İlişkileri
			45	4	3,21,20
			50	2	3,2,3
			5	3	3,3,4
			7	2	3,4,5
			13	1	3,4,6
			7	2	3,4,7
			20	1	3,5,8
			30	4	3,6,8
			15	2	3,7,8
			1	1	3,8,9
			60	3	3,9,17
			1	1	3,2,10
			80	5	3,10,11
			75	4	3,11,12
			80	3	3,12,15
			80	2	3,15,16
			80	1	3,2,13
			80	0	3,13,14
			80	2	3,14,15
			80	4	3,15,16
			80	2	3,16,17
			5	4	3,17,18
			1	1	3,18,19
			30	4	3,19,20
			30	2	3,20,36
			5	4	4,0,1
			1	1	4,1,2
			15	2	4,1,6
			15	4	4,2,10
			5	4	4,6,10
			4	1	4,10,3
			4	1	4,10,7
			7	2	4,3,11
			10	4	4,7,11
			5	3	4,11,4
			60	2	4,4,5
			15	2	4,11,8
			5	2	4,8,9
			10	3	4,5,12
			15	6	4,9,12
			20	5	4,12,13
			8	3	4,13,14
			2	0	4,14,15

Zaman (Gün)	Aktivite sayıları	Kaynak Kapasitesi	Aktivite Süreleri	Kaynak Kullanım Miktarı	Öncüllük İlişkileri
			1	1	4,14,17
			1	1	4,17,20
			50	2	4,15,16
			*	2	4,16,18
				3	4,18,19
				4	4,19,20
				3	4,1,21
				1	4,21,20
				5	4,20,36
				1	5,0,1
				1	5,1,2
				3	5,2,4
				1	5,4,21
				2	5,21,22
				2	5,23,24
				1	5,24,25
				4	5,25,26
				3	5,26,27
				5	5,27,28
				1	5,28,29
				1	5,28,30
				2	5,29,30
				3	5,29,31
				3	5,31,32
				2	5,32,33
				4	5,33,34
				4	5,34,35
				2	5,1,9
				2	5,9,10
				5	5,10,11
				1	5,11,12
				3	5,12,28
				0	5,1,13
				4	5,13,14
				1	5,14,15
				0	5,15,16
				0	5,16,28
				4	5,13,17
				1	5,13,18
				3	5,17,19
				2	5,18,19
				1	5,19,20
				4	5,20,28

Zaman (Gün)	Aktivite sayıları	Kaynak Kapasitesi	Aktivite Süreleri	Kaynak Kullanım Miktarı	Öncüllük İlişkileri
				2	5,2,3
				1	5,3,32
				0	5,2,5
				2	5,5,6
				1	5,5,7
				4	5,5,8
				0	5,6,32
				1	5,7,32
				1	5,8,32
				2	5,35,36
				0	*

EK 11 Java Kodu

READEXCEL . JAVA

```
import java.io.File;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.io.IOException;
import jxl.Cell;
import jxl.CellType;
import jxl.Sheet;
import jxl.Workbook;
import jxl.read.biff.BiffException;
import jxl.read.biff.CellValue;
public class ReadExcel {
    private String inputFile, outFileName;
    private Integer projeSayisi, kaynakTipiSayisi, projeSuresi, maxAktivitesiOlanProje;
    private List<Integer> aktiviteSayisi, kaynakKapasite, faaliyetSureleri, kaynakAktiviteListesi;
    private List<String> tupleList;

    public void setInputFile(String inputFile) {
        this.inputFile = inputFile;
    }

    public void read(String fileName) throws IOException {
        File inputWorkbook = new File(inputFile);
        Workbook w;
        try {
            w = Workbook.getWorkbook(inputWorkbook);
            // Excel dosyasındaki ilk sheet alınır.
            Sheet sheet = w.getSheet(0);

            Cell cell = sheet.getCell(0, 1);
            CellType type = cell.getType();

            //Proje Sayısı okuma kısmı.

```



```

if (type == CellType.NUMBER) {
    projeSayisi = Integer.valueOf(cell.getContents());
    System.out.println("Proje Sayısı " + cell.getContents());
    //Proje Sayısı bulunursa ikinci kolona devam edilip Kaynak Tipi Sayısı okunmaya devam edilir.
}

cell = sheet.getCell(1, 1);

//Kaynak Tipi Sayısı okuma kısmı.
if(type == CellType.NUMBER){
    kaynakTipiSayisi = Integer.valueOf(cell.getContents());
    System.out.println("Kaynak Tipi Sayısı " + cell.getContents());
    // Kaynak Tipi Sayısı bulunursa üçüncü kolonadevam edilip Proje Süresi okunmaya devam edilir.
}

cell = sheet.getCell(2, 1);

//Proje Süresi okuma kısmı.
if(type == CellType.NUMBER){
    projeSuresi = Integer.valueOf(cell.getContents());
    System.out.println("Proje Süresi " + cell.getContents());
    // Proje Süresi bulunursa 4. kolona devam edilip Aktivite Sayıları okunmaya devam edilir.
}

cell = sheet.getCell(3, 1);
aktiviteSayisi = new ArrayList<>();

//Proje bazında aktivite sayısı dizini okuma kısmı.
if(type == CellType.NUMBER){
    int i = 1;    // Aktivite listesinin listesi indexi;
    while(type == CellType.NUMBER) {    //Okunan veri sayısal değer olduğu sürece okumaya devam edilir.
        if(cell.getContents().equals("")) {
            break; //Eğer exceldeki okuma sonunu işaret eden * karakterine gelinirse okumayı
        }
        aktiviteSayisi.add(Integer.valueOf(cell.getContents()));
        System.out.println("Aktivite " + i + "= " + cell.getContents());
        i++;    // İndex bir arttırılır.
        cell = sheet.getCell(3, i);    // Bir sonraki hücre okunur.
    }
}

cell = sheet.getCell(4, 1);
kaynakKapasite = new ArrayList<>();

```

bitirilir.

```

//Kaynak tipi bazında kapasite bilgilerinin okuması kısmı.
if(type == CellType.NUMBER){
    int i = 1;        // Kaynak Kapasite listesi indexi;
    while(type == CellType.NUMBER) { //Okunan veri sayısal değer olduğu sürece okumaya devam edilir.
        if(cell.getContents().equals("")) {
            break;    //Eğer exceldeki okuma sonunu işaret eden * karakterine gelinirse okumayı bitiren komut.
        }
        kaynakKapasite.add(Integer.valueOf(cell.getContents()));
        System.out.println("Kaynak Tipi " + i + " kapasitesi = " + cell.getContents());
        i++; // İndex bir arttırılır.
        cell = sheet.getCell(4, i); // Bir sonraki hücre okunur.
    }
}

cell = sheet.getCell(5, 1);
faaliyetSureleri = new ArrayList<>();
//Faaliyet Sürelerinin okunması kısmı. En çok faaliyet içeren projeninki kadar kayıt oluşturulacak.
maxAktivitesiOlanProje = maksimumAktiviteSayisiOlanProjeyiBul(aktiviteSayisi);
//En çok aktivitesi olan projeyi bulan metod.
System.out.println("En çok aktivitesi olan proje " + (maxAktivitesiOlanProje+1) + ". projedir.
    Aktivite sayısı = " + aktiviteSayisi.get(maxAktivitesiOlanProje));

int excelIndex=1; //Excel kolonundaki index.

while(!cell.getContents().equals("")) { //Okunan veri sayısal değer olduğu sürece okumaya devam edilir.

    for(int i=0;i<projeSayisi;i++) {

        faaliyetSureleri.add(0); //İlk aktivite için 0

        for(int j=0;j<aktiviteSayisi.get(i);j++) {
            faaliyetSureleri.add(Integer.valueOf(cell.getContents()));
            excelIndex++;
            cell = sheet.getCell(5, excelIndex);
        }

        //Eğer aktivite sayısı az ise kalan alanlara 0 basılması kısmı
        for(int k=0;k<aktiviteSayisi.get(maxAktivitesiOlanProje)-aktiviteSayisi.get(i); k++) {
            faaliyetSureleri.add(0);
        }
        faaliyetSureleri.add(0); //Son aktivite için 0
    }
}

```

```

}

for(int i=0;i<faaliyetSureleri.size();i++) {
    System.out.println("Faaliyet Süresi = " + faaliyetSureleri.get(i));
}

cell = sheet.getCell(6, 1);
kaynakAktiviteListesi = new ArrayList<>();

excelIndex=1;           //Excel kolonundaki index.

while(!cell.getContents().equals("")) { //Okunan veri sayısal değer olduğu sürece okumaya devam edilir.
    for(int kaynakTipiSayisiForLoop=0;kaynakTipiSayisiForLoop<kaynakTipiSayisi;kaynakTipiSayisiForLoop++){
        for(int i=0;i<projeSayisi;i++) {
            kaynakAktiviteListesi.add(0);           //İlk aktivite için 0
            for(int j=0;j<aktiviteSayisi.get(i);j++) {
                kaynakAktiviteListesi.add(Integer.valueOf(cell.getContents()));
                excelIndex++;
                cell = sheet.getCell(6, excelIndex);
            }

            //Eğer aktivite sayısı az ise kalan alanlara 0 basılması kısmı
            for(int k=0;k<aktiviteSayisi.get(maxAktivitesiOlanProje)-aktiviteSayisi.get(i); k++) {
                kaynakAktiviteListesi.add(0);
            }
            kaynakAktiviteListesi.add(0);           //Son aktivite için 0
        }
    }
}

for(int i=0;i<kaynakAktiviteListesi.size();i++) {
    System.out.println("Kaynak Aktivitesi = " + kaynakAktiviteListesi.get(i));
}

cell = sheet.getCell(7, 1);
tupleList = new ArrayList<>();

excelIndex=1;           //Excel kolonundaki index.

while(!cell.getContents().equals("")) { //Okunan veri tuple değer olduğu sürece okumaya devam edilir.

```

```

        tupleList.add(cell.getContents());
        excelIndex++;
        cell = sheet.getCell(7, excelIndex);
    }

    for(int i=0;i<tupleList.size();i++) {
        System.out.println("Tupel = " + tupleList.get(i));
    }

} catch (BiffException e) {
    e.printStackTrace();
}

fileName = fileName + ".txt";

PrintWriter outputStream = new PrintWriter(fileName);

outputStream.println("using CPLEX;");
outputStream.println();
outputStream.println("int P = " + projeSayisi.toString() + ";");
outputStream.println("int R = " + kaynakTipiSayisi.toString() + ";");
outputStream.println("int T = " + projeSuresi.toString() + ";");

outputStream.print("int A[1..P] = [");
for(int index=0; index < aktiviteSayisi.size(); index++) {
    outputStream.print(aktiviteSayisi.get(index).toString());
    if(!(index==aktiviteSayisi.size()-1)){
        outputStream.print(",");
    }
}
outputStream.println("];");

outputStream.print("int C[1..R] = [");
for(int index=0; index < kaynakKapasite.size(); index++) {
    outputStream.print(kaynakKapasite.get(index).toString());
    if(!(index==kaynakKapasite.size()-1)){
        outputStream.print(",");
    }
}
outputStream.println("];");
outputStream.println();

```

```

outputStream.println("int d[1..P][0..(max(p in 1..P) A[p])+1] = [");

for(int projeSayForLoop=0;projeSayForLoop<projeSayisi;projeSayForLoop++) {
    outputStream.print("\t\t\t\t\t [");
    for(int indexForEachAction=0;indexForEachAction<aktiviteSayisi.get(maxAkivitesiOlanProje)+2
;indexForEachAction++) {
        outputStream.print(faaliyetSureleri.get(projeSayForLoop * 12 + indexForEachAction).toString());
        if(!(indexForEachAction ==aktiviteSayisi.get(maxAkivitesiOlanProje)+1)){
            outputStream.print(",");
        }
    }
    outputStream.print("]");
    if(!(projeSayForLoop == projeSayisi-1))
        outputStream.println(",");
}
outputStream.println();
outputStream.println("\t\t\t\t\t ];");
outputStream.println();

outputStream.println("int u[1..R][1..P][0..(max(p in 1..P) A[p])+1] = [");

int kaynakAktiviteIndex = 0;

for(int kt = 0; kt<kaynakTipiSayisi; kt++) {
    outputStream.println("\t\t\t\t\t [");
    for(int projeSayForLoop=0;projeSayForLoop<projeSayisi;projeSayForLoop++) {
        outputStream.print("\t\t\t\t\t [");
        for(int indexForEachSourceType=0;indexForEachSourceType<aktiviteSayisi.get(maxAkivitesiOlanProje)+2
;indexForEachSourceType++) {
            System.out.println("Proje Sayisi="

"+projeSayForLoop+"kapasite="+kaynakAktiviteListesi.get(kaynakAktiviteIndex));
            outputStream.print(kaynakAktiviteListesi.get(kaynakAktiviteIndex).toString());
            kaynakAktiviteIndex++;
            if(!(indexForEachSourceType == (aktiviteSayisi.get(maxAkivitesiOlanProje))+1)){
                outputStream.print(",");
            }
        }
        outputStream.print("]");
        if(!(projeSayForLoop == projeSayisi-1))
            outputStream.println(",");
    }
}

```



```

" sum(t in 0..T)t*x[s.j,s.p,t]-\r\n" +
" sum(t in 0..T)t*x[s.i,s.p,t] >= d[s.p,s.i];\r\n" +
"\r\n" +
" forall(t in 0..T,r in 1..R)\r\n" +
" sum(p in 1..P, i in 1..A[p])u[r,p,i]*\r\n" +
" sum(z in maxl((t-d[p,i]+1),0)..t)x[i,p,z] <= C[r];\r\n" +
"}\r\n" +
"\r\n" +
"execute PrintSolution\r\n" +
"{\r\n" +
" writeln(\"Solution\");\r\n" +
" writeln(\"-----\");\r\n" +
" for(var t=0;t<=T;t++)\r\n" +
"   for(var i=0;i<=11;i++)\r\n" +
"     for(var p=1;p<=P;p++)\r\n" +
"       if(x[i][p][t]==1)\r\n" +
"         writeln( \"X[\" , i , \",\" , p , \",\" , t , \"] = \" , x[i][p][t] );\r\n" +
"}");

```

```

outputStream.flush();
outputStream.close();

```

```

}

```

```

public int maksimumAktiviteSayisiOlanProjeyiBul(List<Integer> aktiviteList) {
    int projeIndex=0;
    for(int i=0;i<aktiviteList.size();i++) {
        if(aktiviteList.get(projeIndex) < aktiviteList.get(i)) {
            projeIndex=i;
        }
    }
    return projeIndex;
}

```

```

public static void main(String[] args) throws IOException {
    ReadExcel test = new ReadExcel();
    test.setInputFile("C:\\Deniz\\importForOpl.xls");
}

```

```

public Integer getProjeSayisi() {
    return projeSayisi;
}

```

```

}

public void setProjeSayisi(Integer projeSayisi) {
    this.projeSayisi = projeSayisi;
}

public Integer getKaynakTipiSayisi() {
    return kaynakTipiSayisi;
}

public void setKaynakTipiSayisi(Integer kaynakTipiSayisi) {
    this.kaynakTipiSayisi = kaynakTipiSayisi;
}

public Integer getProjeSuresi() {
    return projeSuresi;
}

public void setProjeSuresi(Integer projeSuresi) {
    this.projeSuresi = projeSuresi;
}

public List<Integer> getAktiviteSayisi() {
    return aktiviteSayisi;
}

public void setAktiviteSayisi(List<Integer> aktiviteSayisi) {
    this.aktiviteSayisi = aktiviteSayisi;
}

public List<Integer> getKaynakKapasite() {
    return kaynakKapasite;
}

public void setKaynakKapasite(List<Integer> kaynakKapasite) {
    this.kaynakKapasite = kaynakKapasite;
}

public List<Integer> getFaaliyetSureleri() {
    return faaliyetSureleri;
}

public void setFaaliyetSureleri(List<Integer> faaliyetSureleri) {
    this.faaliyetSureleri = faaliyetSureleri;
}

```



```
public List<Integer> getKaynakAktiviteListesi() {
    return kaynakAktiviteListesi;
}

public void setKaynakAktiviteListesi(List<Integer> kaynakAktiviteListesi) {
    this.kaynakAktiviteListesi = kaynakAktiviteListesi;
}

public List<String> getTupleList() {
    return tupleList;
}

public void setTupleList(List<String> tupleList) {
    this.tupleList = tupleList;
}

public String getInputFile() {
    return inputFile;
}

public Integer getMaxAktivitesiOlanProje() {
    return maxAktivitesiOlanProje;
}

public void setMaxAktivitesiOlanProje(Integer maxAktivitesiOlanProje) {
    this.maxAktivitesiOlanProje = maxAktivitesiOlanProje;
}

public String getOutFileName() {
    return outFileName;
}

public void setOutFileName(String outFileName) {
    this.outFileName = outFileName;
}
```

JFRAMEFOROPL.JAVA

```
import java.awt.BorderLayout;
import java.awt.EventQueue;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.border.EmptyBorder;
import javax.swing.JButton;
import javax.swing.JFileChooser;
import java.awt.Color;
import java.awt.Font;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JTextArea;
import java.awt.Choice;
import java.awt.Button;
import java.io.File;
import java.io.IOException;
import java.awt.Desktop;
import javax.swing.JTextPane;
import java.awt.GridLayout;
import java.awt.TextField;
import javax.swing.BoxLayout;
```

```
import javax.swing.JTextField;
import javax.swing.JLabel;
import javax.swing.JOptionPane;
import java.awt.Panel;
import javax.swing.JSplitPane;
import java.awt.FlowLayout;

public class JFrameForOpl extends JFrame {
    private JPanel contentPane;
    private final Panel yuklePanel = new Panel();
    public File file;

    /**
     * Launch the application.
     */
    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    JFrameForOpl frame = new JFrameForOpl();
                    frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

```

    } catch (Exception e) {
        e.printStackTrace();
    }
}
});
}

/**
 * Create the frame.
 */
public JFrameForOpl() {
    setTitle("OPL Verisi Y\u00FCKlemek \u0130n Excel Dosyas\u0131 Se\u00E7iniz");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setBounds(100, 100, 750, 250);
    contentPane = new JPanel();
    contentPane.setBorder(new EmptyBorder(5, 5, 5, 5));
    setContentPane(contentPane);
    contentPane.setLayout(new BorderLayout(contentPane, BorderLayout.X_AXIS));
    JSplitPane splitPane = new JSplitPane();
    splitPane.setOrientation(JSplitPane.VERTICAL_SPLIT);
    contentPane.add(splitPane);
    FlowLayout fl_yuklePanel = (FlowLayout) yuklePanel.getLayout();

```

```
fl_yuklePanel.setAlignment(FlowLayout.LEFT);
splitPane.setLeftComponent(yuklePanel);
yuklePanel.setSize(250, 50);
```

```
JLabel inputLabel = new JLabel("Y\u00Fcklenecek Excel Dosyas\u0131:");
yuklePanel.add(inputLabel);
inputLabel.setFont(new Font("Calibri", Font.BOLD, 14));
```

```
JTextArea inputTextArea = new JTextArea();
inputTextArea.setForeground(Color.BLACK);
inputTextArea.setEnabled(false);
inputTextArea.setLineWrap(true);
inputTextArea.setColumns(40);
inputTextArea.setRows(2);
inputTextArea.setLineWrap(true);
```

```
yuklePanel.add(inputTextArea);
```

```
Button importButton = new Button("G\u00F6zet");
yuklePanel.add(importButton);
importButton.setFont(new Font("Calibri", Font.BOLD, 12));
importButton.setForeground(Color.BLUE);
```

```
importButton.setBackground(Color.YELLOW);  
importButton.setActionCommand("yukle");
```

```
Panel ciktiPanel = new Panel();  
FlowLayout flowLayout = (FlowLayout) ciktiPanel.getLayout();  
flowLayout.setAlignOnBaseline(true);  
flowLayout.setAlignment(FlowLayout.LEFT);  
splitPane.setRightComponent(ciktiPanel);  
ciktiPanel.setSize(250, 50);
```

```
JLabel outputLabel = new JLabel("Olu\u015Fturulacak Text Dosyas\u0131 \u0130smi:");  
outputLabel.setFont(new Font("Calibri", Font.BOLD, 14));  
ciktiPanel.add(outputLabel);
```

```
JTextArea outputTextArea = new JTextArea();  
outputTextArea.setRows(2);  
outputTextArea.setColumns(32);  
ciktiPanel.add(outputTextArea);
```

```
Button olusturButton = new Button("Olu\u015Ftur");  
olusturButton.setEnabled(false);
```

```

olusturButton.setActionCommand("olustur");
olusturButton.setFont(new Font("Calibri", Font.BOLD, 12));
olusturButton.setForeground(Color.BLUE);
olusturButton.setBackground(Color.YELLOW);
ciktiPanel.add(olusturButton);
importButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        JFileChooser fileChooser = new JFileChooser();
        fileChooser.setCurrentDirectory(new File(System.getenv("SystemDrive")));
        StringBuilder sb = new StringBuilder();
        pick_file(fileChooser);
    }
    public void pick_file(JFileChooser fileChooser) {
        if(fileChooser.showOpenDialog(null) == JFileChooser.APPROVE_OPTION) {
            file = fileChooser.getSelectedFile();
            inputTextArea.setText(file.getPath());
            olusturButton.setEnabled(true);
        }
    }
});

```

```
olusturButton.addActionListener(new ActionListener() {  
    @Override  
    public void actionPerformed(ActionEvent e) {  
        if(!outputTextArea.getText().isEmpty()) {  
            ReadExcel test = new ReadExcel();  
            test.setInputFile(file.getPath());  
            try {  
                test.read(outputTextArea.getText());  
            } catch (IOException ioe) {  
                // TODO Auto-generated catch block  
                ioe.printStackTrace();  
            }  
        } else {  
            JOptionPane.showMessageDialog(null, "Lütfen Çıktı Dosyasının İsmi Giriniz!!!");  
        }  
    }  
});  
}
```