

**Proje Çizelgelemede
Kısıtlı Kaynakları ve Devresel Değişkenlikleri
Dikkate Alan Bir Algoritma Önerisi**

Cem Tuğrul Atasever

YÜKSEK LİSANS TEZİ

Endüstri Mühendisliği Anabilim Dalı

Ekim 2009

**A Project Scheduling Algorithm
Considering Resource Constraints
And Seasonal Effects**

Cem Tuğrul Atasever

MASTER OF SCIENCE THESIS

Department of Industrial Engineering

October 2009

**Proje Çizelgelemede
Kısıtlı Kaynakları Ve Devresel Değişkenlikleri
Dikkate Alan Bir Algoritma Önerisi**

Cem Tuğrul Atasever

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Endüstri Mühendisliği Anabilim Dalı
Yöneylem Araştırması Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Yrd. Doç. Dr. Servet HASGÜL

Ekim, 2009

ONAY

Cem Tuğrul Atasever' in YÜKSEK LİSANS tezi olarak hazırladığı “**Proje Çizelgelemede Kısıtlı Kaynakları ve Devresel Değişkenlikleri Dikkate Alan Bir Algoritma Önerisi**” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğinin ilgili maddeleri uyarınca değerlendirilerek kabul edilmiştir.

Danışman: Yrd. Doç. Dr. Servet HASGÜL

Yüksek Lisans Tez Savunma Jürisi:

Üye : Prof. Dr. Nihat YÜZÜGÜLLÜ

Üye : Doç. Dr. Nuray GİRGINER

Üye : Yrd. Doç. Dr. Aydın SİPAHİOĞLU

Üye : Yrd. Doç. Dr. R. Aykut ARAPOĞLU

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve sayılı kararıyla onaylanmıştır.

Prof. Dr. Nimetullah BURNAK

Enstitü Müdürü

ÖZET

Kaynak kısıtlı proje çizelgeleme problemi, öncelik kısıtları ve sınırlı kaynaklar altında faaliyetlerin en az sürede tamamlanmasını amaçlamaktadır. Kaynak maliyeti ve kapasitelerin devresel olarak değişiklik gösterdiği durumlarda devresel değişiklikler altında kaynak kısıtlı bir çizelgeleme yapmak gerekecektir. Kaynak maliyetleri, devresel değişkenlik gösterdiğinde kaynak çizelgeleme yapılırken, kaynak maliyetinin yüksek olduğu devrelerden kaçınılarak toplam maliyet en küçüklenmeye çalışılacaktır.

Bu çalışmada kısıtlı kaynak ve devresel değişkenli proje çizelgeme için bir algoritma tasarlanarak, tek ve çok kaynak kısıtı ile tekli ve çoklu projeler üzerinde algoritmanın çalışması test problemleri üzerinde denenmiş ve sonuçları incelenmiştir.

Proje yöneticilerine karar desteği sağlayacak şekilde geliştirilen algoritmanın kullanıldığı bir yazılım hazırlanmıştır. Yapı sektöründe bir uygulama projesi bu yazılım ile çizelgelenmiş ve elde edilen sonuçlar değerlendirilmiştir.

Anahtar Kelimeler: Kaynak kısıtlı proje çizelgeleme, kaynak dengeleme, devresellik

SUMMARY

Resource-constrained project scheduling problem (RCPS), which aims at scheduling a set of activities at minimal duration subject to precedence constraints and limited resource availabilities. Resource costs and resource capacities may change seasonally. In this situation, resource constrained scheduling has to be done by considering the seasonal variations. While the resource costs are under seasonal variations and scheduling the resources, the total cost is aimed to be minimized by staying away from the periods which have high resource costs

In this study, an algorithm is designed for resource constrained project scheduling with seasonal variation, experiments are done on the test problems which have one / multiple resources with one project and multiple projects and the results are evaluated.

As such by providing a decision support for project managers, software is prepared that makes the developed algorithm. A real life problem is scheduled by using this software on the construction field and derived solutions are evaluated.

Key words: Resource constrained project scheduling, resource leveling, seasonality

TEŐEKKÜR

Çalıőmamn her aőamasında yakın ilgi ve desteęini gördüğüm; çalıőmalarımın yönlendirilmesi ve sonuçlandırılmasında büyük emeęi geçen tez danıőmanım Sayın Yrd. Doç. Dr. Servet HASGÜL'e çalıőmalarım sırasında beni maddi açıdan destekleyen Türkiye Bilimsel ve Teknik Araőtırma Kurumu Bilim İnsanı Destekleme Daire Başkanlığı'na (TÜBİTAK-BİDEB) teőekkür etmeyi bir borç bilirim.

İÇİNDEKİLER

	<u>Sayfa</u>
ÖZET	v
SUMMARY.....	vi
TEŞEKKÜR.....	vii
ŞEKİLLER DİZİNİ	xii
ÇİZELGELER DİZİNİ	xiii
1. GİRİŞ.....	1
2. PROJE VE PROJE YÖNETİMİ KAVRAMI.....	3
2.1 Proje Kavramı.....	3
2.2 Proje Yönetimi Kavramı.....	6
2.2.1 Proje yönetiminin aşamaları	7
2.2.2 Proje yönetim teknikleri ve tarihsel gelişimi	10
2.3 Projenin Çizelgelenmesi	12
2.3.1 Çizelgeleme teknikleri	13
2.3.2 Gantt şeması.....	15
3. KAYNAK ÇİZELGELEME	19
3.1 Kaynakların Sınıflandırılması.....	19
3.2 Öncüllük İlişkileri.....	20
3.3 Kaynak Kısıtlı Proje Çizelgeleme	21
3.3.1 Kaynak kısıtlı proje çizelgeleme probleminin tanımı.....	21

3.3.2	KKPÇP'nin serim üzerinde gösterimi	22
3.3.3	Faaliyet - kaynak ilişkileri	22
3.3.4	Değişik amaç fonksiyonları	23
3.4	Tek Modlu Kaynak Kullanımlı Projelerde Proje Süresinin Enküçüklenmesi Problemi.....	23
3.4.1	Problemin tanımı ve matematiksel programlama gösterimi	24
3.4.2	Kesin çözüm yöntemleri	25
3.4.3	Sezgisel yöntemler	26
3.5	Çok Modlu Kaynak Kullanımlı Projelerde Proje Süresinin En Küçüklenmesi Problemi.....	33
3.5.1	Problemin tanımı ve matematiksel programlama gösterimi	33
3.5.2	Kesin çözüm yöntemleri	34
3.5.3	Meta-sezgisel yöntemler	35
3.6	Kaynak Hareketli Çizelgeleme	35
3.6.1	Kaynak birleştirilmesi.....	36
3.6.2	Kaynak biriktirme	36
3.6.3	Kaynak dağılımı.....	37
3.6.4	Kaynak düzleştirme	38
3.7	Çoklu Proje Yönetimi	38
3.7.1	Kaynak kısıtlı birden çok projenin çizelgelenmesi.....	39
3.7.2	Kaynak kısıtlı çoklu proje çizelgeleme için kurallar	42

3.7.3	Çoklu proje çizelgeleme	44
4.	KISITLI KAYNAK VE DEVRESEL DEĞİŞKENLİ PROJE ÇİZELGELEME İÇİN BİR ALGORİTMA ÖNERİSİ.....	50
4.1	Devresel Değişiklikler Altında Proje Çizelgeleme.....	50
4.2	Önerilen Algoritmanın Tanıtımı	52
4.3	Algoritmanın Örnek Problemler Üzerinde Çalıştırılması.....	56
4.3.1	Tek kaynak kısıtının olması durumunda çözüm	56
4.3.2	Kısıtlı çok kaynağın olması durumunda çözüm	60
4.3.3	Çoklu projelerin çözümü	63
4.4	Önerilen Algoritmanın Literatürde Yer Alan Test Problemleri İle Denenmesi	67
4.4.1	Projelerin kaynak maliyetlerinin dikkate alınmadan çizelgenmesi ...	69
4.4.2	Çoklu projelerin kaynak maliyetlerinin dikkate alınmadan çizelgenmesi.....	72
4.4.3	Projelerin devresel kaynak maliyetlerinin dikkate alınarak çizelgenmesi.....	72
4.4.4	Çoklu projelerin devresel kaynak maliyetlerinin dikkate alınarak çizelgenmesi.....	77
4.4.5	Test sonuçlarının değerlendirilmesi.....	77
4.5	Geliştirilen Karar Destek Sistemi	78
4.6	Geliştirilen Algoritmanın Yapı Sektörüne Uygulanması	84
5.	SONUÇ VE ÖNERİLER.....	87
	KAYNAKLAR DİZİNİ.....	89

EKLER..... 97

ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 3.1 Çoklu Proje Çizelgeleme Yaklaşımı.....	45
Şekil 4.1 Algoritma Akış Diyagramı	55
Şekil 4.2 Örnek Problem Serim Diyagramı	56
Şekil 4.3 Birikimli Toplam Maliyetler	61
Şekil 4.4 Toplam Birikimli Kaynak Maliyetleri.....	63
Şekil 4.5 Proje 2 Serim Diyagramı	63
Şekil 4.6 Çoklu Projelerin Birleştirilmesi.....	65
Şekil 4.7 Serim Diyagramı Oluşturulması Ekran Görünümü	79
Şekil 4.8 Faaliyet İlişkileri Formu	80
Şekil 4.9 Kaynak Bilgileri Formu.....	80
Şekil 4.10 Devre Bilgileri Formu	81
Şekil 4.11 Devresel Kaynak Maliyetleri Formu	81
Şekil 4.12 Faaliyet Bilgileri Formu	82
Şekil 4.13 Çizelge Formu	83
Şekil 4.14 Kaynak Kullanım Grafiği	83
Şekil 4.15 Uygulama Projesi Birikimli Toplam Maliyetler.....	86

ÇİZELGELER DİZİNİ

	<u>Sayfa</u>
Çizelge 4.1 Örnek Proje Faaliyet Süreleri ve Kaynak Gereksinimleri	57
Çizelge 4.2 Örnek Proje İçin CPM ile Elde Edilen Çözüm.....	57
Çizelge 4.3 Örnek Proje Tek Kaynak ve Sabit Maliyet ile Çözüm	58
Çizelge 4.4 Devresel Kaynak Maliyet ve Kapasiteleri	58
Çizelge 4.5 Örnek Proje Tek Kaynak ve Devresel Maliyet ile Çözüm	59
Çizelge 4.6 Örnek Proje Tek Kaynak ve Devresel Maliyet ile Çözüm	60
Çizelge 4.7 Örnek Proje Kaynak Gereksinimleri	61
Çizelge 4.8 Örnek Proje Devresel Kaynak Maliyet ve Kapasiteleri.....	62
Çizelge 4.9 Örnek Projenin Çok Kaynak Kısıtı Altında Çözümü	62
Çizelge 4.10 Proje 2 Faaliyet Süreleri ve Kaynak Gereksinimleri	64
Çizelge 4.11 Örnek Proje 2 Çözümü	64
Çizelge 4.12 Çoklu Projelerin CPM ile Çizelgelenmesi.....	66
Çizelge 4.13 Çoklu Projelerin Çok Kaynak Kısıtı ile Çözülmesi.....	67
Çizelge 4.14 PSPLIB Veri Setleri.....	68
Çizelge 4.15 Öncelik Kuralları	68
Çizelge 4.16 En iyi Sonuca Ulaşma Oranları	70
Çizelge 4.17 En İyi Sonuca Yakınlık.....	70
Çizelge 4.18 Öncelik Kurallarının Karşılaştırılması.....	71
Çizelge 4.19 J30 Problemleri için Ortalama Sapma ve Çalışma Süresi Değerleri	71

Çizelge 4.20 Çoklu Projelerin Kaynak Maliyetsiz Çizelgelenmesi.....	73
Çizelge 4.21 Devresel Maliyetler İçin Veri Setleri.....	74
Çizelge 4.22 Devresel Maliyetler ile Yapılan Çözüm	76
Çizelge 4.23 Devresel Maliyeler ile Yapılan Çözüm –Çoklu Proje.....	77
Çizelge 4.24 Uygulama Projesi Devreleri	85
Çizelge 4.25 Uygulama Projesi Devresel Kaynak Maliyet ve Kapasitleri	85

1. GİRİŞ

Günümüz ekonomik koşulları altında kaynakların, sürenin ve bütçenin sınırsız olarak kullanılabilirdiği bir ortam öngörmek mümkün değildir. Kaynakların kıt, zamanın çok değerli olduğu bu koşullarda, kaynakların en akılcı biçimde kullanımını sağlayacak bir yönetim anlayışının gerekliliği açıktır. Zamanın ve paranın en iyi şekilde değerlendirilmesi, kısıtlı olan malzeme, işgücü, makine-ekipman gibi kaynakların en uygun biçimde kullanılmasını sağlamak amacıyla, bir projenin başlangıcından bitimine kadar olan sürecin planlanması ve programlanması gerekmektedir. Planlama ve programlama süreçleri proje yönetiminin temelini oluşturmaktadır.

Çizelgeleme; yapılacak olan işin zamana yayılması ile oluşan iş programını kontrol ederek, karar vericiye sunan bir karar destek mekanizmasıdır. Proje planlama, programlama için gerekli olan tüm verilerin toplanmasıyla başlar. Belirli bir zaman diliminde yapılacağı belirtilen iş miktarları, ilgili analizlerdeki değerlerin hesaplanan miktarlarla çarpımı ile oluşturulacak işgücü programı, makine ekipman donanımı, malzeme programı ve maliyet programı haline gelmektedir.

Kaynak dengeleme problemi, kaynak istemlerinin tepe değerlerini aşağıya çekerek çeşitli dönemlerdeki kaynak ihtiyaçlarını mümkün olduğunca aynı seviyeye getirmeyi amaçlayan bir problemdir. Proje değerlendirme ve gözden geçirme tekniği (PERT) ve Kritik Yol Yöntemi (CPM) serimlerinde kaynak dengeleme sorunlarına çözüm arayan pek çok bulgusal yöntem geliştirilmiştir. Küçük çaplı projelerdeki kaynak dengeleme problemlerinde kolaylıkla uygulanan bu yöntemler, büyük çaplı projelerde her zaman istenen sonucu vermeyebilir. Büyük çaplı projeler için kaynak dengelemesinin oldukça karmaşık olmasına karşılık, yüzlerce faaliyeti ve onlarca kaynağı dikkate alıp çözüm geliştiren bilgisayar yazılımları mevcuttur.

Kaynakların maliyeti sabit olabileceği gibi devresel olarak değişebilir. Örneğin yapı projelerinde belirli sıcaklıkların altında ısıtma kabloları kullanıldığından, beton dökme işleminin maliyetleri fazla olabilmektedir. Aynı şekilde işçilik maliyeti de kullanılan ekipman ve giysiler nedeniyle artmaktadır. Bu nedenle birçok kaynak maliyeti kalemi yapı projelerinde mevsimsel olarak değişmektedir. Benzer şekilde

malzeme ve ekipman fiyatları da devresel olarak deęişkenlik gösterebilir. Kaynak kapasiteleri projenin ve faaliyetin koşullarına baęlı olarak devresel deęişiklikler gösterebilir. Kaynak maliyetinin deęişken olduęu ortamda faaliyetler maliyetlerin düşük olduęu devrelerde yapılarak proje maliyetleri düşürölmek istenecektir. Fakat bu durumda da proje süresi uzayabilir.. Proje süresinin uzamasıyla oluşun gecikme ile yine proje maliyetinin artması gözlenebilir. Kaynakların düşük maliyetli dönemlere kaydırılması eęer proje süresinin uzamasına neden oluyorsa, gecikmenin oluşturacaęı maliyet ile kıyaslanması ve deęerlendirilmesi gerekecektir. Mevcut kaynak çizelgeleme algoritmaları devresel deęişkenleri göz ardı etmektedir. Devresel deęişkenlięin olduęu durumlarda pratikte uygulanabilir ve etkin çözümler elde etmek için devresel deęişkenleri dikkate alan bir algoritma tasarlanabilir.

Bu çalışmada kısıtlı kaynak ve devresel deęişkenli proje çizelgeleme için bir algoritma tasarlanmıştır. Kaynak maliyet ve kapasitelerinin devresel olarak belirlenebildięi ve önerilen algoritma ile çözümler üretebilen bir karar destek sistemi tasarlanmıştır.

Çalışmanın ikinci kısmında, Proje ve Proje Yönetim kavramlarının gelişimiyle ilgili bilgi verilmiştir. Proje Yönetim Teknikleri ve uygulama alanlarından bahsedilmiş Proje Çizelgeleme ve Proje Çizelgeleme Tekniklerine yer verilmiştir.

Çalışmanın üçüncü bölümünde, Kaynak Çizelgeleme, Kısıtlı Kaynakların Çizelgenmesi, Kısıtlı Kaynaklara ilişkin matematiksel modeller ve çözüm tekniklerinden bahsedilmiştir. Kaynak Hareketli Çizelgeleme ve Kaynak Dengeleme teknikleri ayrıntılı olarak anlatılmıştır. Çoklu Proje Yönetimi ve Çoklu Proje Çizelgeleme teknikleri hakkında bilgiler verilmiştir.

Çalışmanın dördüncü bölümünde ise, önerilen algoritma ayrıntılı olarak açıklanmış ve örnek problemlerle gösterilmiştir. Literatürdeki test problemleri ile elde edilen sonuçlar deęerlendirilmiştir. Geliştirilen karar destek sisteminin veri giriş ekranları ve kullanımı tanıtılmıştır. Bir yapı firmasında kaynak kısıtları altında, kaynak maliyetlerinin devresel olarak deęiştii bir proje, hazırlanan karar destek sistemi ile çizelgenmiş ve sonuçlar deęerlendirilmiştir.

2. PROJE VE PROJE YÖNETİMİ KAVRAMI

2.1 Proje Kavramı

Proje kavramının çeşitli tanımları vardır. Örneğin Juran projeyi çözümlü planlanıp, çizelgelenmiş bir sorun olarak tanımlamıştır. Bu tanım bize projenin var olan bir problemi ortadan kaldırmaya yönelik çabaların toplamı olduğu sonucuna götürür. Problemin doğru tanımlanması projenin başarısını olumlu etkiler. Archibald tarafından yapılmış bir başka tanım “Proje yeni bir ürün, yeni bir bina, tesis, yeni bir sistem veya özel bazı sonuçların elde edilmesi için gerçekleştirilmesi gereken toplam süreçtir” der. General Electric firması proje kavramını özel bir amacın elde edilmesi için, sınırlı bir zaman aralığı için tanımlanmış faaliyetler olarak tanımlamış ve projenin bir defaya mahsus olarak yapıldığını vurgulamıştır (Wermter,1996).

Bir projede, insan ve insan dışı kaynaklar, başlangıcı ve sonu belirli bir amaç için bir organizasyon çerçevesinde bir araya getirilir. Proje sonunda ise bu kaynaklar başka yerlere tahsis edilir (Gitmez, 1998).

Bütün bu tanımlardan çıkarılacak sonuç, projenin bir ihtiyaca cevap verdiği, bir amaca yönelik olduğu, yalnızca bir tek defalık yapıldığı, başlama ve bitiş tarihlerinin belli olduğu, bir organizasyon yapısı içinde gerçekleştiği ve kaynak tükettiğidir (Gitmez, 1998).

Alman standardı DIN 69901’a göre proje kavramı açıklanırken “bir defalık olmasına” ve “benzerinin olmamasına” dikkat çekilmekte ve süre, hedef, kaynak (işgücü, sermaye, finansal kaynaklar v.b.) gibi kısıtları bulunan ve kendine özgü organizasyon özelliklerine sahip işlerin ve görevlerin proje olarak nitelendirilebileceği belirtilmektedir (Koçel, 1993).

Proje, genellikle üç yıldan daha az süreli olan ve çeşitli örgütsel birimler tarafından yerine getirilen, birbiriyle ilişkili işlerden oluşan, iyi tanımlanmış bir amacı, belirli bir zaman çizelgesi ve bütçesi olan karmaşık bir çabadır (Barutçugil, 1983).

Bir iş grubunu “proje” haline getiren ana özellikler olarak şunlar belirtilebilir (Koçel, 1993);

- İş grubu içinde yer alan faaliyetlerin belirli başlangıç ve bitiş noktalarının bulunması,
- İş grubunun ve bunun içindeki faaliyetlerin çoğu kez bir defaya mahsus olmak üzere yapılması, yani geçici olması (iş grubundaki faaliyetlerin bir defa yapılması ile amacın gerçekleşmesi)
- İş grubu içindeki bu faaliyetlerin her zaman tam olarak bilinmemesi, belirsizliklerin olması,
- İşlerin zamanında bitirilmemesinin işletme açısından önemli kayıplara neden olması,
- İş grubu içinde yer alan her işin bitiş zamanı, maliyeti ve kalitesi açısından belirli standartların olması,
- İşlerin ve faaliyetlerin gerçekleşmesi sırasında, işlerle ve standartlarla ilgili sürekli değişikliklerin olması,
- Faaliyetlerin gerektirdiği kaynakların miktarının zaman içinde çan eğrisi şeklinde dağılması,
- İşletmelerin ele aldıkları iş gruplarının yapı projelerinin çoğu kez birbirinden farklı olması,
- Projeyi gerçekleştiren işletme ile projenin sahibi arasında, çoğu kez hukuki nitelikte bir kontratın bulunması,

Eğer bir işletmenin yaptığı işler yukarıdaki özellikleri taşıyorsa, bu işler “Proje” adı altında ele alınabilir.

İşletmelerin ele aldığı ve yürüttüğü çok çeşitli projeler bulunmaktadır. Bunlar genellikle (Barutçugil,1988);

1. Mal veya hizmet üretimi için sözleşme ile yürütülen ticari projeler,
2. Yeni ürün ve teknoloji üzerinde yapılan araştırma-geliştirme ve mühendislik projeleri,
3. İnşaat ve sabit sermaye yatırım projeleri,
4. Bilgi sistemleri ve yönetim projeleri
5. Büyük bakım projeleri olmak üzere beş ana başlık altında toplanırlar.

Ticari Projeler: Mühendislik uygulamaları gerektiren ve belirli bir müşteri talebini veya iç gereksinimini karşılamak amacıyla üstlenilen her türlü sözleşme veya programlardır. Bunlar; teknik, pazarlama, imalat ve diğer yönlerden oldukça karmaşık yapıdadırlar. Teknik risk söz konusudur. Birden fazla sayıda bağımsız örgütün katkısını gerektirir.

Araştırma – Geliştirme ve Mühendislik Projeleri: Yeni ürün veya üretim teknolojisi geliştirmek veya mevcut ürün veya teknolojilerde önemli değişiklikler yapmak amacıyla başlatılan veya üstlenilen program veya projelerdir. İleri düzeyde yenilik, belirsizlik ve risk özelliklerini içerir.

İnşaat ve Sabit Sermaye Yatırım Projeleri: Genellikle arazi, bina makine-teçhizat satın alımları, kiralanmaları ve inşa edilmeleri amacıyla yapılan harcamaları veya mevcut sabit tesislerin büyük ölçüde değiştirilmeleri ve yeniden düzenlemeleri için gereken olağan dışı çabaları kapsar.

Bilgi Sistemleri ve Yönetim Projeleri: İşin yürütülmesini ve örgütlenmesini, yeniden düzenlenen ve işletme içindeki haberleşmeyi, denetimi ve yönetimi etkinleştirme amacıyla ele alınan her türlü projelerdir.

Büyük Bakım Projeleri: Özellikle süreç endüstrisinde devresel olarak yapılan koruyucu bakım ve yenileme program ve projeleridir.

2.2 Proje Yönetimi Kavramı

Proje yönetiminin tarihsel gelişimi incelendiğinde bugünkü anlamda ilk Proje Yönetimi uygulamalarının II. Dünya Savaşı yıllarında başladığı görülür.

Savaş yıllarının beraberinde getirdiği kısıtlayıcı ve zorlayıcı etkiler ve özellikle zaman faktörünün baskısı, o yıllarda A.B.D.'de bazı ulusal ve askeri projelerin gerçekleştirilmesinde savunma gerekleri çerçevesinde o zamana kadar denenmemiş ve hâlihazırda mevcut organizasyon, planlama, izleme ve kontrol yöntemleri ile çözümü mümkün olmayan karmaşık işlem ve süreçlerin bir arada yönetimi sorunu ortaya çıkarmıştır.

1941 yılında başlatılan “Manhattan Engineering Project” adlı proje ile ilk atom bombasının geliştirilmesi sırasında zaman faktörünün kısıtlayıcı etkisi altında çok sayıda karmaşık sistem görevlerinin gerçekleştirilmesi için üniversitelerden, endüstriden, silahlı kuvvetlerden ve kamu sektöründen birçok bilim adamı, uzman, mühendis, asker ve hükümet yetkililerinin çalışmalarının projenin toplam hedefi doğrultusunda planlanarak organize edilmesi ve uyum sorunu ile karşı karşıya kalınmıştır.

İlk proje uygulaması bu ve bunu izleyen diğer bazı askeri amaçlı ve uzay-havacılık projelerinde görülmüştür. Bu süreçte önemli kilometre taşlarından biri de II. Dünya Savaşı sonrası A.B.D.'de başlatılan askeri amaçlı “Polaris Program” ile NASA'nın “Apollo” projeleri olmuştur.

ABD'de geliştirilmiş olan proje yönetimi uygulamaları savaş sonrası yıllarda Batı Avrupa ülkelerinde de kullanılmaya ve hızlı bir şekilde yayılmaya başlamıştır.

Örneğin daha sonraları Avrupa Uzay Ajansı (ESA) adı altında birleşmiş olan European Space Vehicle Launcher Development Organization (ELDO) ve European Space Research Organization (ESRO) ile NATO, ABD'de geliştirilmiş olan bu uygulamaların kısmen ya da tamamen Avrupa'ya aktarılmasında taşıyıcı rolünü üstlenmişlerdir.

İlk olarak ortaya çıkışı ve geliştirilmesi askeri amaçlı ve uzay-havacılık projeleri ile olan proje yönetimi teknikleri kısa bir süre içinde endüstriyel üretim, araştırma ve geliştirme, inşaat, tarım ve hizmet sektörlerine yayılmıştır (Ünle, 2007).

Yirminci yüzyılın ikinci yarısında önem kazanan ve başta gelişmiş ülkeler olmak üzere pek çok ülkede ve her düzeyde kullanımı hızla artan proje yönetimi şu şekilde açıklanabilir:

Proje Yönetimi Enstitüsü (2000)'e göre proje yönetimi; projenin gereksinimlerini karşılamak için proje etkinliklerine bilginin, becerilerin, araçların ve tekniklerin uygulanması olarak tanımlanabilir (Ives, 2005).

Proje yönetimi; proje olarak tanımlanan işlerin, bu tanımdaki zaman, performans ve kaynak kriterleri ile kısıtları uyarınca önceden belirlenmiş olan hedefe ulaşmak için sürdürülen planlama, organizasyon, yönetim, kaynakların tahsisi ve kullanımı, uygulama, izleme, kontrol ve değerlendirme faaliyetlerinin bütünüdür (Peşkircioğlu, 1989).

2.2.1 Proje yönetiminin aşamaları

Yönetimin amacı, etkinliği ve verimliliği sağlamaktır. Bunu sağlamak için yönetim sürecinin 5 evreden oluştuğu kabul edilir. Bunlar şöyle verilmektedir (Tosun, 1982);

- i.** Planlama Evresi
- ii.** Örgütlenme(Organizasyon) Evresi
- iii.** Yürütme ve Yönlendirme (Emir-Komuta) Evresi
- iv.** Eşgüdüm (Koordinasyon) Evresi
- v.** Kontrol ve Düzeltme Evresi

Planlama, bir amacı gerçekleştirmek için en iyi davranış biçimini belirleme, geliştirme sürecidir. Bu süreç sonunda neyin, ne zaman, neden, nasıl ve kim tarafından

yapılacağı kararlaştırılır, programlanır. Örgütlenme ise belirlenen amaca ulaşmak için nasıl bir organizasyon yapısı kurulması gerektiğinin belirlendiği süreçtir. Bu aşamada kişilerin yetki ve sorumlulukları da tanımlanır. Örgütlenme aslında planlama ile birlikte gerçekleştirilen bir süreçtir.

Yürütme ve yönlendirme, işlerin nasıl yaptırılacağına belirlenmesidir. Aslında yöneticilik bir işi başkasına isteterek yaptırmak biçiminde de tanımlanmaktadır.

Kontrol, gerçekleşenin planlanana uygun olup olmadığının gözlenmesi, düzeltme ise arada fark varsa bunun giderilmesi için yapılacakların belirlenmesidir. Ancak kontrolün yapılabilmesi için önceden standartların belirlenmesi, kontrolün ne zaman, kimin tarafından ve nasıl yapılacağına belirlenmiş olması gerekir.

Projenin belirli kısıtlar uyarınca, planlanan değer ve hedeflere en uygun şekilde yürütülmesi için gerekli kararları zamanında alarak uygulamaya koyabilecek yetki ve sorumluluktaki bir kişi, proje yöneticisi olarak görev yapmalıdır (Turan, 1993). Proje yöneticisinin karşılaştığı en önemli sorun, projenin istenen zamanda, planlanan bütçeyle ve istenen kalite düzeyini yakalayarak tamamlamaktır. Projenin bitiş süresine uymak için kaliteden feragat edilebilir veya daha çok kaynak kullanmak gerekeceği için maliyet artar. Kalitenin arttırılmaya çalışılması da planlanan bütçenin aşılması veya süresinin uzaması tehlikesi ortaya çıkar. Maliyetin planlanan seviyeyi geçmemesi için gösterilecek çaba, hem süre hem kalitede hedeflenen sapmaya yol açabilir. Kısaca zaman, maliyet ve kalite birbiriyle çelişen 3 amaçtır ve proje yöneticisi bunları belli bir hedefi sağlamak için dengelemeye çalışır.

Amerikan Proje Yönetim Enstitüsü, proje yöneticisinin altı fonksiyonunu şöyle tanımlamıştır (Thomsett, 1996);

1. Hedefleri ve yapılacak işleri yeteri kadar ayrıntılı bir şekilde, anlamayı ve düzeltici faaliyetlere girişmeye imkân vermek için belirleyip, proje ölçeğini yönetmek.

2. Projeye dahil olan insan kaynaklarını yönetmek.

3. Projeye dahil olan çeşitli taraflar arasında bilgi akışını sağlamak ve projeyi çizelge dahilinde yürütmek için yeterli bilgiye sahip olmak, iletişimi yönetmek.

4. Zamanı, planlama ve çizelgeleme ile yönetmek.

5. Projenin sonuçlarının tatmin edici olması için kaliteyi yönetmek.

6. Projenin en düşük maliyetle ve bütçe içinde bitirilmesi için maliyetleri yönetmek.

Bu görevleri yerine getirmek için proje yöneticisinin bazı özellikleri olmalıdır; Kendisi yapılan işi genel olarak bilmelidir, münazara ve ikna kabiliyeti yüksek olmalıdır (Thomsett, 1996).

Proje yöneticisi genelde yüksek baskı ve stres altında olacaktır. Sık değişen önceliklerle ve çıkarları zıt insanlarla, bütçeyi aşan maliyetlerle, çizelge ve yapılan işle başa çıkmak zorunda kalacaktır (Trevor, 1998).

Bunların üstesinden gelmek için, işi sevmeye, teknik bilgi ve beceriye, analitik zekaya ve sabır, insanları dengeleme yeteneklerine ve liderlik özelliklerine sahip olmak zorundadır. Kısaca yönetici için şu özellikleri sıralayabiliriz (Trevor, 1998);

1. İletişim yetenekleri
2. Tartışma ve ikna etme yetenekleri
3. Liderlik yetenekleri (Hedef belirleme ve astlarını organizasyon istikamette sevk etme)
4. İnsanlarla kuvveti ilişkiler kurma ve bunları yürütebilme yeteneği
5. Pazarlama, mukavele yapma, müşterilerle ilişki yeteneği
6. Kaynak yönetimi
7. Teknik bilgi ve beceriler
8. Bütçeleme ve maliyet yönetimi yetenekleri
9. Çizelgeleme ve zaman yönetimi becerileri.

2.2.2 Proje yönetim teknikleri ve tarihsel gelişimi

Bir projede mevcut işlerin şekilsel gösteriminde en basit yöntem 1918 yılında Henry L. Gantt tarafından geliştirilen Gantt şemasıdır. Gantt şemasının en önemli özelliği projeyi faaliyet boş zamanları olarak ifade ederken her faaliyetin zaman grafiğini ve faaliyetler arası ilişkileri sergilemede kolaylık sağlayan esnekliğidir.

Modern proje yönetimi teknikleri, 19. Yüzyılın sonlarında karmaşıklaşan iş yaşamı ile birlikte şekillenen ve gelişen yönetim ilkelerinin evrimleşmesi ile elde edilmiştir. Özellikle o yıllarda gerçekleştirilen büyük ölçekli devlet projeleri proje yönetimi tekniklerinin gelişmesinde itici güç olmuştur.

Proje yönetimi ile ilgili ilk bilimsel çalışma 1900'lü yılların başında Frederick Taylor (1856-1915) tarafından gerçekleştirilmiştir. Taylor, yönetim tekniklerinin bilimsel olarak analiz edilebileceğini ve geliştirilebileceğini göstererek yönetim anlayışında yeni bir sayfa açmıştır. Taylor'un çalışmalarından önce verimliliği artırmanın tek yolu işçilerin daha uzun saatler boyunca daha sıkı çalıştırılmasıydı. Taylor, iş süreçlerinin bileşenlerini tek tek analiz ederek, iş planlamasını daha verimli hale getirmiştir.

Bilimsel yönetim akımının öncülerinden biri olan Henry L. Gantt, bugün yaygın biçimde uygulanan proje izleme ve değerlendirme yöntemlerinden olan PERT ve CPM'in esasını oluşturan "Gantt Şemaları"nı geliştirmiştir. Bu grafikler sayesinde proje takvimini oluşturmada büyük kolaylıklar sağlanmış, bilgisayarın da devreye girmesiyle birlikte, proje mühendislerinin işi iyice kolaylaşmış ve iş takibi kolayca yapılır olmuştur.

Proje kapsamındaki faaliyetlerin tamamlanma zamanlarını dikkate alarak projenin takibini kolaylaştıran tekniklerden olan PERT (ilk kez 1958 yılında Amerikan ordusunda Polaris denizaltı füzelerinin yapımı projesinde kullanıldı) ve CPM ile birlikte karmaşık projelerin en geç tamamlanma zamanlarının hesaplanabilmesi, projenin daha erken zamanda tamamlanması istendiğinde yeni düzenlemelerin yapılmasına imkan

vermesi proje yöneticilerinin, projenin akışı üzerindeki kontrollerinin artmasına yardımcı oldu.

İlk önce askeri alanda silah geliştirilmesi konusunda kullanılan bu teknikler değişen rekabetçi piyasaların etkisiyle endüstriyel projelerin de vazgeçilmez araçları haline geldi. Anılan yöntemler ülkemizde de birçok büyük projede kullanılmıştır. II. Fatih Sultan Mehmet Köprüsü ve Güney Doğu Anadolu Projesi CPM, Keban Barajı ve İstanbul Boğaziçi Köprüsü PERT' in uygulandığı projelere örnek gösterilebilir (www.bilgiyonetimi.org).

Proje yönetiminde esas gelişme 1950 ve 1960'lı yıllarda olmuştur. Özellikle Apollo uzay programı ile yeni planlama tekniklerine gereksinimler olduğu fark edilmiş ve serim modelleri, CPM, PERT faaliyetler arasındaki farklı öncelik ilişkileri, iş ayrımı şeması, matris organizasyon, kazanılmış değer, öncelik diyagramı, kaynakların dengelenmesi gibi yeni kavramlar ve teknikler türetilmiştir. Geliştirilen bu teknikler günümüzde de uygulanmaktadır.

1950'li yıllarda projeden tek başına sorumlu bir kişinin yani proje yöneticisinin olması gerekliliği fark edilmiştir. Hemen sonrasında proje ekipleri, matris organizasyon yapısı, kaynakların paylaşımı gibi kavramlar geliştirilmiştir. Bir projenin serim biçiminde gösterilebileceği ise 1956 yılında Flagle tarafından gösterilmiştir. Bu çalışma aynı zamanda PERT' in de öncüsü kabul edilmektedir (Burke, 1999).

Serim modelleri projenin görsel olarak kolaylıkla algılanmasını sağlar. Karmaşık öncelik sonralık ilişkilerini göstermekte çok başarılıdır. Faaliyetlerin ne kadar geciktirilebileceklerine ve bunun proje üzerinde ne gibi etkileri olacağına dair önceden detaylı bilgiler verir. Oluşabilecek darboğazların görülmesini sağlar. Serim modelleri ile birden fazla projenin aynı anda planlanması ve kontrolü ve farklı proje terminlerine göre, toplam proje maliyetleri hesaplanarak, arzu edilen planın seçilmesi mümkündür.

Hatta farklı projelerin aynı serimde izlenip ortak kaynakların kullanımı konusunda plan yapmak da mümkündür. Serim modelinin en büyük yararı ise faaliyetlerden bazılarını çıkarma, yeni faaliyet ekleme gibi güncellemeleri projeye başarılı bir şekilde yansıtılmasına ve yeni duruma ait planların kolayca türetilmesine

olanak vermesidir. Böylece projenin günü gününe izlenerek aksayan noktalara müdahale etme şansı artar.

İlk geliştirilen serim modeli CPM' dir. 1957 yılında Remington Rand şirketinden J. E. Kelly ve Du Pont şirketinden M. R. Walker tarafından kimyasal ürün üretecek fabrikaların kurulması ve bakımı faaliyetlerinin çizelgelenmesi amacıyla geliştirilmiştir (Sht ve diğerleri, 1994).

Matris organizasyon yapısı ilk kez 1960 yılında kullanılmış, 1963 yılında proje hayat döngüsü ve kazanılmış değer analizi kavramları tanımlanmıştır.

Bilgisayarların gelişimi de proje yönetimini olumlu etkilemiştir. İlk proje yönetimi yazılımı 1983 yılında geliştirilen Harvard Project Manager' dır. Günümüzde de varlığını sürdüren bu yazılımın yanı sıra, Primavera ve Microsoft Project yazılımları da profesyonel programlardır.

2.3 Projenin Çizelgelenmesi

Çizelgeleme; belirli kriterlere uyulması koşulu ile bir ya da daha fazla amacı iyileştirecek şekilde, gerçekleşmesi gereken görevlerin sıralarını ve mevcut kaynakların zaman içindeki kullanımını göstermektir. Çizelgelenin bir diğer tanımı da, belirli kriterlere ve kısıtlara uymak koşulu ile gerçekleşmesi gereken görevlerin daha az süre ve kaynak kullanarak problemin çözümünü göstermektir (Biroğul, 2005). Bu nedenle çizelgeleme; proje/üretim planlama ve yönetimin karar aşamalarında konu edilen çalışmanın etkinlik ve verimliliğini belirleyen önemli bir fonksiyondur. Proje çizelgeleme; mevcut kısıtlı kaynakları kullanarak, belirli amaçlar çerçevesinde yerine getirilmesi gereken görevlerin gerçekleşmesi için program yapılmasıdır. Bu açıdan proje ne kadar iyi çizelgelenebilirse, yönetici ve uygulayıcıların etkinlik ve verimliliği o oranda artmaktadır.

Proje çizelgeleme; doğası gereği yalın bir çalışma konusu olmayıp, çizelgelemeyi etkileyen pek çok unsur içermektedir. Örneğin; faaliyetlerin öncüllük ilişkileri, faaliyet süreleri, faaliyetin gerçekleşmesi için gereken kaynaklar ve miktarları gibi unsurlar sayılabilir. Çizelgelemeyi etkileyen temel unsur, süreç planlama

aşamasıdır. Süreç planlama aşaması, faaliyetlerin mevcut imkânlar çerçevesinde sıralanması ya da düzenlenmesi aşamasıdır (Paksoy, 2007).

2.3.1 Çizelgeleme teknikleri

Projede yer alan faaliyetlerin çizelgelenmesinde en yaygın olarak üç teknik kullanılmaktadır. Bu teknikler Gantt Şeması, Kritik Yol Metodu ve Proje Değerlendirme ve Geliştirme Tekniği metotlarıdır. Bunlardan CPM ve PERT şebeke mantığını kullanmaktadır. Bu bölümde kısaca bu teknikler açıklanacaktır (Moder ve Phillips,1970).

2.3.1.1 Kritik yol metodu

Kritik yol metodunun temel amacı faaliyetler arası öncelik ilişkilerini dikkate alarak projenin en erken tamamlanma zamanına ulaşmaktır. Bu sebeple her bir faaliyete ilişkin en erken başlama, en erken tamamlanma, en geç başlama ve en geç tamamlanma zamanları hesaplanır. Kritik yol yönteminde Bitiş-Başlangıç (Finish-to-Start) ilişkisi kullanılmaktadır. Kritik yol analizinde sürelerin hesaplanması ileriye ve geriye doğru hesaplamalarla yapılmaktadır.

İleriye doğru hesaplamada en erken başlama (ES_j) ve en erken tamamlanma zamanı (EF_j) her bir faaliyet için hesaplanmaktadır. Daha sonra bu hesaplamalar sonunda projenin tamamlanma süresi ortaya çıkarılmaktadır. Projenin tamamlanma zamanı en son faaliyetin en erken tamamlanma süresine eşitlenmektedir. Bu kurala sıfır proje boşluğu kuralı denilmektedir.

P_j : j faaliyetinin öncülleri

d_j : j faaliyetinin süresi olmak üzere;

$$ES_j = \max \{ EF_i \mid i \in P_j \} \quad j = 2 \dots \dots n \quad (2.1)$$

$$EF_j = ES_j + d_j \quad j = 2 \dots \dots n \quad (2.2)$$

Daha sonraki aşamada her bir faaliyete ilişkin en geç tamamlanma zamanı LF_j ve en geç başlama zamanı LS_j hesaplanmaktadır. Bu hesaplama ise $LF_n = LS_n$ ise Proje tamamlanma süresi olarak alınır. Bu durumda;

$$LF_j = \max \{LS_h \mid h \in P_j\} \quad j = n-1 \dots\dots 1 \quad (2.3)$$

$$LS_j = LF_j - d_j \quad j = n-1 \dots\dots 1 \quad (2.4)$$

Proje tamamlanma süresini hesaplamak üzere bir doğrusal model yazılmak istenirse, ST_j , j faaliyetinin başlangıç zamanını göstermek üzere (Elmaghraby vd., 1995; Kamburowsky vd., 2000);

Model:

$$ST_i + d_i \leq ST_j \quad j = 2 \dots\dots n \text{ ve tüm } i \in P_j \quad (2.5)$$

$$ST_1 = 0 \quad (2.6)$$

kısıtları altında,

$$\min (ST) = ST_n \quad (2.7)$$

şeklinde yazılabilir.

Burada amaç fonksiyonu (2.7) başlangıç zamanlarını ST_n en küçükler. Böylece proje tamamlanma süresini de en küçüklemiş olur. Kısıt (2.5) öncelik ilişkilerinin gerçekleşmesinin sağlar. En son olarak da (2.6) kısıtı ile planlama sürecinin başlangıcı daima sıfıra eşitlenerek projenin başlanması sağlanmış olur. Projedeki faaliyetlerin başlangıç zamanlarıyla ilgili olarak ileriye doğru hesaplama sonucunda elde edilen ES_j ile ST_j $j= 2 \dots\dots n-1$ birebir örtüşmeyebilir.

2.3.1.2 Bolluk zamanlarının hesaplanması

Bir projenin olası en erken tamamlanma zamanı o projedeki en uzun süren faaliyetlere bağlıdır. Proje süresinin elde edildiği bu faaliyetler kritik faaliyetler olarak adlandırılırlar. Oluşan kritik faaliyetler dizisine kritik yol denir.

Kritik yol üzerinde olmayan faaliyetler ise bazı bolluk zaman değerlerine sahiptirler. Bu bolluk zamanlarına göre projelerdeki faaliyetlerin bir kısmı belirli sınırlar dahilinde uzatılabilir ya da ertelenebilir. Bolluk ya da aylak zamanlar olarak ifade edilen bu zamanların hesaplanması projenin planlanması ve kontrolü açısından önem arz etmektedir. Projenin planlanması bakımından düşünüldüğünde kritik olmayan faaliyetlerin bolluk zamanlarına göre yeniden düzenlenmesi bütçe ve kaynak kısıtlarının asılmasına yardımcı olabilmektedir. Bununla ötesinde çoklu projelerde ortak kaynak kullanımı durumu söz konusu olabilmektedir. Ortak kaynak kullanımı durumlarında bir projedeki bolluktan istifade edilerek diğer projeye bazı kaynakların aktarılması darboğazların asılmasına yardımcı olabilmektedir. Projenin kontrolü bakımından kritik ve kritik olmaya yakın yolların etkili takibini kolaylaştırmaktadır (Paksoy, 2007).

2.3.2 Gantt şeması

Gantt şeması; yatay eksene zaman, düşey eksene faaliyetlerin sıralanmasıyla çizilir. Çoğu kez faaliyetlerin başlangıçtan bitişe doğru sıralanmış hali kullanılır. Kaynak dengeleme gibi özel durumlarda farklı sıralamalarda kullanılabilir. Şemada ilk faaliyet sıfıncı andan başlayarak süresi kadar yatay çubukla ifade edilecek şekilde çizilir. İzleyen faaliyetlerde öncülük ilişkisine bağlı olarak konumlandırılır. Bütün faaliyetlerin çizimi tamamlandığında, son faaliyetin bitiş zamanı projenin tamamlanma zamanını, başlangıçtan bitişe kadar kesintisiz olarak süren faaliyetler dizisi de kritik faaliyetleri gösterir. Diğerleri de geciktirilebilir faaliyetler olarak isimlendirilir. Şemadan bu faaliyetlerin ne kadar geciktirilebilecekleri, öncülük ilişkilerine dikkat edilerek hesaplanır (Ünle, 2007).

2.3.2.1 PERT Yöntemi

PERT 1958 yılında birleşik devletler donanması tarafından tam bir silah sisteminin gelişimini programlamada bir araç olarak bulunmuştur (Cottrell, 1999). PERT, İngilizce, Project Evaluation and Review Technique kelimelerinin baş harflerinden oluşmuş bir sözcük olup, Proje Değerlendirme ve Gözden Geçirme Tekniği anlamına gelmektedir. PERT'in tarihsel gelişimi incelendiğinde, Gantt şemalarının PERT'e temel teşkil ettiği görülmektedir. Ancak Gantt şeması, yapılacak işin safhaları arasındaki ilişkiyi bir dereceye kadar gösterdiğinden, bir takım eksiklikleri olan bir

yöntemdir (Levin ve Kirkpatrick, 1973). Bu nedenle PERT'e Gantt şemasının daha gelişmiş bir biçimi de denilebilir.

PERT başlangıçta bir projenin zaman unsurlarına uyarlanmış ve belirli bir tarihte bir projenin tamamlanabilme ihtimalini belirlemede yardımcı olacak olasılıksal zaman hesaplarını kullanmıştır (Lu ve Li, 2003). PERT tıpkı CPM programlaması gibi, uygulama zamanının analizi için mantık diyagramları kullanır. PERT şemaları ok üzerinde faaliyet diyagramları olarak çizilir. PERT programlama olay üzerine odaklanır. Programlamacının en muhtemel proje süresini ve belirli bir zamanda projenin ya da projenin herhangi bir kısmının tamamlanacağı ihtimalini hesap etmesine olanak verir. Bu sebeple PERT, olasılıksal bir metodu göz önünde bulundurur. Çünkü PERT hesaplama sürecine olasılıkları da katar (Callahan, 1992).

PERT, sabit faaliyet süreleriyle ilgilenmektense, her faaliyetin süresini bazı olasılık dağılımları kullanarak tespit etmektedir. Bu yöntemde, üç süre tahmini yapmayı sağlayacak istatistiksel yöntemler kullanılarak, projedeki belirsizliklerle başa çıkılmaya çalışılır. Bu süreler, en iyimser (a), en kötümser (b) ve en olası (m) sürelerdir.

Faaliyetler arasındaki öncelik ilişkileri belirlenerek, her faaliyet için 3'lü süre tahmini yapıldıktan sonra, şebeke diyagramı çizilir ve her faaliyetin en erken ve en geç başlama ve bitiş süreleri ve gecikme süreleri bulunur. Bu teknikte her bir faaliyetin beklenen süresi Formül 2.8'de gösterildiği gibi hesaplanmaktadır (İpeköz, 1994).

$$\mu = (a+4m+b)/6 \quad (2.8)$$

$$\text{Varyansı bulmak için : } \sigma^2 = [(b-a)/6]^2 \quad (2.9)$$

$$\text{Standart sapmayı bulmak için ise: } \sigma = (b-a)/6 \quad (2.10)$$

Varyansı ve standart sapmayı hesaplamak için Formül 2.9 ve Formül 2.10 kullanılmaktadır (Krajewski ve Thompson, 1981). PERT'te faaliyetlerin beklenen süresinin hesaplanmasında Beta dağılımı kullanılmaktadır.

Ayrıca PERT sayesinde, (2.11) formülü ve standart normal dağılım tablosu kullanılarak, projenin farklı tarihlerde tamamlanma olasılıklarını hesaplamak mümkündür.

$$z = (T - T_c) / \sigma_c \quad (2.11)$$

Burada $P(z)$, projenin belli bir sürede tamamlanma olasılığını, T , projenin programlanan süresini, T_c , projenin beklenen bitirilme süresini, σ_c ise standart sapmayı göstermektedir (Yamak, 1994). PERT'te önemli olan projenin ne kadar sürede bitirilebileceğini tespit etmek iken, PERT'in farklı bir biçimi olan PERT/Maliyet de ise önemli olan, projenin maliyetiyle ilgili bilgileri ortaya koymaktır. Bu teknikte, her faaliyet için, maliyetin en küçük olması durumunda süre tahminleri ve sürenin en küçük olması durumunda maliyet tahminleri yapılır.

PERT in başlıca avantajları aşağıdaki gibi sıralanabilir:

- PERT in temel kullanımı daha önce yapılmamış projeler içindir.
- PERT zaman ve maliyet hesaplarına zemin sağlar.
- PERT aynı zamanda diğer belirleyici metotlardan daha fazla bilgi erişimine izin verir.
- PERT belirli bir zaman aralığında kapsamlı bir projenin tamamlanma başarısının ya da belirlenmiş tarihlerin kesin kilometre taşlarına ulaşma olasılığını değerlendirme fırsatı verir (Callahan, 1992).

Fakat, PERT yöntemine karşı, 1960'lardan bu yana kaynaklarda birçok eleştiri ve düzenleme ortaya çıkmıştır. PERT le ilgili başlıca problem ve kısıtlar ise aşağıdaki gibidir;

- PERT zaman alabilecek çoklu zaman yaklaşımları gerektirir (Callahan, 1992; Cottrell, 1999).
- Proje mühendisleri ve planlamacıları için bir faaliyetin iyimser, en olası ve olurlu süreçlerini hesap etmek zordur.
- Beta paylaşımlarının bütün proje faaliyetlerine uygulanabileceği farz edilir.

- PERT, proje tamamlama zamanı olasılıklarını hesaplamada sadece kritik yolu göz önünde bulundurur. Kritikleşmede önemli bir olasılık olan yakın kritik yolları yok sayar (Cottrell, 1999).

3. KAYNAK ÇİZELGELEME

Bir projede her faaliyetin gerçekleştirilmesi için çeşitli kaynaklar kullanılır. Bunlar, işgücü, malzeme, donanım ve paradır. Ancak para çok özel bir kaynak olduğu için ayrı bir başlıkta incelenir. Proje yönetiminde kaynak çizelgelemesi denince işgücünün, malzemenin ve donanımın kullanımıyla ve bunların çizelgenmesiyle ilgili sorunlar anlaşılır.

Çizelgelemeden kasıt, proje boyunca hangi zaman diliminde, hangi kaynaktan ne kadar ihtiyaç olacağını önceden belirlenmesi, eldeki olanakların projeyi zamanında bitirmeye yetip yetmeyeceğinin analiz edilmesidir. Buna göre kaynaklar yetmeyecekse veya kaynakların aynı zaman diliminde farklı yerlerde kullanımını gerektiren, pratikte uygulanması mümkün olmayan sıkıntılı durumlar ortaya çıkıyorsa, önceden önlem alınarak kaynak miktarı arttırılabilir veya bu mümkün değilse eldeki olanaklarla projenin geciktirilmeden veya mümkün olan en az gecikmeyle nasıl tamamlanabileceği belirlenmeye çalışılır. Sonuç olarak proje faaliyetleri için yeni çizelge hazırlanarak, yeni durum için hangi zaman diliminde hangi kaynaklara, ne kadar gerek duyulacağını belirlenmesi çalışması yapılır.

3.1 Kaynakların Sınıflandırılması

Kısıtlı kaynakların mevcudiyeti, kaynakların kullanımında belirli amaç işlevini veya işlevlerini eniyileyecek kaynak tahsislerini önemli kılar. Kaynaklar faaliyetlerin gerçekleştirilmesi için kullanılırlar. Faaliyetler burada geniş anlamda tanımlanabilir; örneğin, yeni bir katın betonunun atılması, bir frezenin bir parçanın yüzeyini belirli bir ölçüye indirmesi, bir işlemcinin bir alt programı işlemesi gibi. Kaynakları çeşitli şekillerde sınıflandırmak mümkündür. Zaman bazına göre sınıflandırmada; yenilenebilir, yenilenemez ve çift yönden kısıtlı kaynaklar şeklinde bir sınıflandırma yapılır.

Yenilenebilir kaynakların bir zaman birimi içindeki toplam kullanım miktarı kısıtlı olmakla birlikte kullanım ile tüketilmezler. Belirli bir faaliyet çerçevesinde kullanıldıktan sonra faaliyet sona erince tekrar kullanıma hazır duruma gelirler.

Yenilenemez kaynaklar kullanımları esnasında tüketilirler. Çeşitli faaliyetlerde kullanılan tüketim malzemeleri yenilenemez kaynakların tipik örnekleridir. Yenilenemez kaynakların proje süresi boyunca kullanılabilir toplam miktarı üzerinde kısıt vardır. Bir kaynağın hem bir zaman birimi içinde kullanım miktarı hem de proje süresi boyunca toplam tüketimi üzerinde kısıt olması durumunda bu kaynak çift yönden kısıtlı kaynak olarak nitelendirilir. Örneğin; para çift yönden kısıtlı bir kaynak olabilir; zaman birimi içindeki harcama kısıtlı olabildiği gibi proje süresince toplam harcama miktarı da kısıtlı olabilir.

Kısmi yenilenebilir kaynaklar modellemede belirli zaman dilimleri için farklı kaynak kullanımı üst sınırı tanımlama olanağı vermektedir. Buna göre, örneğin, vardiya düzenini problem gösterimine dâhil etmek mümkün olmaktadır. Bu şekilde, daha gerçekçi problem tanımlarına olanak tanınmaktadır (Drexl vd., 1999). Diğer bir farklı kaynak kullanımı durumu da bir faaliyetin uygulama süresi boyunca değişken kaynak kullanımı talebi olmasıdır (Sprecher, 1994).

Diğer bir kaynak sınıflandırma da kaynakların bölünebilirliğine dayandırılır. Adet olarak ifade edilen, yani bölünebilen kaynaklara, örneğin beş adet iş makinesi gibi, ayrık kaynaklar denir. Buna mukabil, örneğin elektrik enerjisi gibi bölünemeyen kaynaklara da sürekli kaynaklar olarak tanımlanır.

3.2 Öncüllük İlişkileri

Çizelgeleme, sıralamadan farklı olarak faaliyetlerin hangi sırada gerçekleştirilecekleri bilgisi dışında bu faaliyetlerin ne zaman başlayıp ne zaman biteceği bilgisini de içerir. Çizelgelenen faaliyetlerinin kullandığı kaynaklar kısıtlı olabildiği gibi faaliyetler arasında öncüllük ilişkileri de olabilir. Öncüllük ilişkileri teknolojik bir gereksinimi yansıttığı gibi, tamamen yönetimsel bir kararın sonucu da olabilirler. Öncüllük ilişkileri, zaman bazında faaliyetler arasındaki başlangıç – bitiş ilişkilerini belirler. Bir faaliyet, öncül faaliyetlerinin hepsi bitmeden başlayamaz.

Genellikle kabul edilen bir faaliyetin bütün öncülleri tamamlanınca hemen başlayabileceğidir. Ancak uygulamada farklı durumlar söz konusu olabilmektedir. Örneğin, eşzamanlı tasarım faaliyetlerinde proje süresini kısaltmak için bazı faaliyetlerin kısmen paralel yürütülmesi istenir. Bir faaliyet sonucu elde edilen bozulabilir bir malzemeyi kullanacak olan bir faaliyetin malzeme hazır olduktan sonra belirli bir süre içinde başlaması istenebilir. Bu değişik durumları modelleyebilmek için geliştirilmiş öncüllük ilişkileri tanımlanmıştır. Bunlar; Başlangıç-Bitiş (SF), Başlangıç–Başlangıç (SS), Bitiş Bitiş (FF) ve Bitiş–Başlangıç (FS) tipi öncüllük ilişkileridir. Bütün bu öncüllük ilişkileri için geçerli olmak üzere, en az bekleme süresi ve en çok bekleme süresi tanımlanmıştır. Örneğin, bir faaliyetin başlaması için diğer bir faaliyetin en azından belirli bir süre önce başlamış olması isteniyorsa, bu bir FS ilişkisidir ve en az bekleme süresi ile modellenir. Bir faaliyetin başlaması için diğer faaliyetin başlamasından sonra belirli bir süreden fazla geçmemiş olması gerekiyorsa, bu bir SS ilişkisidir ve en çok bekleme süresi ile modellenir.

3.3 Kaynak Kısıtlı Proje Çizelgeleme

Bu bölüme kaynak kısıtlı çizelgeleme probleminin tanımı verilerek başlanacaktır. Ardından KKPÇP'nin Serim Üzerinde Gösterimi, Faaliyet Kaynak İlişkileri ve Değişik Amaç İşlevleri bölümleri anlatılarak devam edilecektir.

3.3.1 Kaynak kısıtlı proje çizelgeleme probleminin tanımı

Kaynak kısıtlı proje çizelgeleme problemi (KKPÇP), kısıtlı kaynaklar kullanılarak bir projeyi oluşturan faaliyetlerin, öncüllük ilişkilerini ihlal etmeden amaç işlevini eniyeleyecek biçimde çizelgelenmesidir.

Problemin tanımına esas teşkil eden varsayımları aşağıdaki gibi sıralayabiliriz:

- 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 faaliyeti gerçekleştirilmek zorundadır.

Kaynak kısıtlı proje çizelgeleme probleminin proje yönetimi için ifade ettiği pratik değerin yanında, teorik açıdan da araştırmacılar açısından bir çekiciliği vardır. Proje süresinin en küçüklendiği kaynak kısıtlı proje çizelgeleme probleminin *NP-zor* bir problem olduğu Blazewicz vd. (1983) tarafından gösterilmiştir.

3.3.2 KKÇP'nin serim üzerinde gösterimi

Kaynak kısıtlı proje çizelgeleme problemi bir serim (*network*) üzerinde tanımlanır. Bilindiği gibi bir serim, düğümler kümesi $\{V\}$, ayrıtlar kümesi $\{A\}$ ve düğümler ve ayrıtları ilişkilendiren ilişkiler kümesi $\{\Phi\}$ ile tanımlanır. İki tanımlama seçeneği vardır. Birinci uygulamada; faaliyetler ayrıtlar kümesinin, olaylar ise düğümler kümesinin elemanları olarak tanımlanır. İlişkiler kümesi, hangi faaliyetin hangi olaylar arasında yer aldığı bilgisini içerir. Bu tanımlama seçeneğine, *faaliyetlerin ayrıtlarda gösterimi* (FAG) denir. FAG serimin oluşturulmasında yapay faaliyet tanımına gerek duyulabilmektedir. Proje çizelgeleme problemlerinin serim üzerinde tanımlanmasında ilk kullanılan gösterim şekli FAG'dir. Diğer tanımlama seçeneği, faaliyetlerin düğümler üzerinde gösterildiği tanımlama şeklidir. Ayrıtlar öncüllük ilişkilerini gösterir. Bu tanımlama seçeneğine, *faaliyetlerin düğümlerde gösterimi* (FDG) denir. FDG yapay faaliyet tanımına gerek olmadığı için daha basit bir gösterimdir.

3.3.3 Faaliyet - kaynak ilişkileri

Bir faaliyetin gerçekleştirilme süresi ile kullanılan kaynaklar arasında bir ilişki vardır. Genel olarak, birim zaman içinde daha çok kaynak kullanımının maliyeti yükseltmesi, faaliyet süresini ise azaltması beklenir. Bu öngörüden hareketle, bu uygulama maliyet-süre ödünleşimi şeklinde modellenebilir. Kullanılan kaynağın ayrık veya sürekli olmasından bağımlı olarak maliyet-süre ödünleşimi ayrık veya sürekli bir işlev olarak ifade edilir. Ayrık işlev durumunda, işlevin her bir maliyet-süre çiftine karşı gelen noktası bir mod olarak nitelendirilir. Bir veya daha fazla sayıda faaliyeti birden fazla moda sahip proje çizelgeleme problemleri çok modlu problemler olarak nitelendirilirler.

Diğer bir ödünleşme türü kaynaklar arasındaki ödünleşimdir. Burada bir faaliyetin süresi sabittir ancak değişik kaynakların değişik kullanımları söz konusudur. Örneğin bir kanal açma faaliyetinin bir kazı makinesi, bir operatör ve iki düz işçi yerine aynı süre içinde on altı düz işçi tarafından yapılması gibi.

Proje yönetiminin uygulamalarında genellikle birden fazla projenin aynı kaynak havuzundan yararlanılarak yönetimi söz konusudur. Bu tür problemler, çok projeli (*multi-project*) çizelgeleme problemleri olarak nitelendirilirler.

3.3.4 Değişik amaç fonksiyonları

Kaynak kısıtlı proje çizelgeleme probleminde değişik amaç fonksiyonları kullanılmıştır. Proje yöneticisi açısından bakıldığında en önemli performans ölçütlerinden birisi projenin süresidir. Projenin elde mevcut kaynaklarla en az sürede bitirilmesi hedeflenir. Kaynak kısıtlarının olmadığı varsayılan ilk proje çizelgeleme gösterimlerinde kullanılan amaç fonksiyonu proje süresinin (C_{max}) en küçüklenmesidir. (Malcolm, vd., 1959; Kelley, 1961). KKPÇP' nde de en yaygın olarak kullanılan amaç proje süresinin en küçüklenmesidir (Özdamar ve Ulusoy, 1995). Projenin bir termini olması durumunda, termin bazlı başka amaç işlevleri de incelenmiştir. Bunlardan bir tanesi, gecikmenin en küçüklenmesidir. Bir diğeri de, tam-zamanında-üretim anlayışına yönelik olarak, erken bitirme ve gecikme toplamının en küçüklenmesidir. (Vanhoucke vd., 2000). Burada, erken bitirme ve gecikmenin ayrı ayrı etkileri olması durumunda, ağırlıklı toplam en küçüklenir.

Diğer sık kullanılan amaçlardan birisi de toplam maliyet veya toplam kârın en büyüklenmesidir (Herroelen vd., 1997). Bu amaçlar kullanılırken, nakit akışına paranın zaman değeri uygulanarak bugünkü değeri bulunur ve maliyet veya kâr, net bugünkü değer (NBD) olarak ifade edilir.

3.4 Tek Modlu Kaynak Kullanımlı Projelerde Proje Süresinin Enküçüklenmesi Problemi

Bu bölümde tek modlu kaynak kullanımlı projelerde proje süresinin enküçüklenmesi probleminin tanımı ve matematiksel gösterimi, kesin ve sezgisel çözüm

yöntemleriyle birlikte, öncelik kurallarına dayalı tek çözüm ve çok çözüm üreten sezgisel yöntemler anlatılmıştır. Son olarak da bu problem için kullanılan metasezgisellere değinilmiştir.

3.4.1 Problemin tanımı ve matematiksel programlama gösterimi

İncelenen problem, tek bir projenin yukarıda yapılan varsayımlar çerçevesinde tamamlanma süresinin enküçüklenmesi problemidir. Kaynak kullanımı tek modludur. KKPÇP'nin bu en basit halinin matematiksel programlama gösterimi aşağıda verilmiştir (Ulusoy,1996). Bu gösterimde proje seriminin oluşturulması için FDG kullanılmıştır.

Değişkenler:

$$X_{jt} = \begin{cases} 1, & \text{eğer } j \text{ faaliyeti } t \text{ dönemi sonunda bitiyorsa} \\ 0, & \text{diğer durumlarda} \end{cases} \quad (3.1)$$

Kısıtlar:

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

$$\sum_{t=EFT_i}^{LFT_i} tX_{jt} \leq \sum_{t=EFT_j}^{LFT_j} (t-d_j)X_{jt} \quad j=2, \dots, J, i \in P_j \quad (3.3)$$

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

Amaç Fonksiyonu:

$$\text{enk } Z = \sum_{t=EFT_j}^{LFT_j} tX_{jt} \quad (3.5)$$

Kullanılan sembollerin tanımları aşağıda verilmiştir:

t = zaman indisi ($t=1, \dots, T$),

j = faaliyet indisi ($j=1, \dots, J$),

R = yenilenebilir kaynaklar kümesi,

d_j = j faaliyetinin süresi,

P_j = j faaliyetinin öncüllerinin kümesi,

EFT_j = j faaliyetinin en erken bitiş zamanı,

LFT_j = j faaliyetinin en geç bitiş zamanı,

k_{jr} = j faaliyetinin r kaynağından birim zaman kullanım miktarı,

K_r = r yenilenebilir kaynağının birim zaman kullanım üst sınırı.

(3.5) probleminde, amaç işlevi bitiş faaliyetinin bitiş zamanını diğer bir deyişle, proje süresini enküçükleme problemidir. (3.2) kısıt kümesi her faaliyetin mutlaka çizelgelenmesini sağlar. Kısıt kümesi (3.3) ise, j faaliyeti ile bu faaliyetin öncülü olan i faaliyeti arasındaki öncüllük ilişkisinin yerine getirilmesi gereğini ifade eder. Birim zaman başına kaynak kısıtı ise kısıt kümesi (3.4) ile gösterilmiştir. (3.1) ise $\{0,1\}$ değişkeni x_{jt} 'nin tanımıdır. Değişken sayısını azaltabilmek amacı ile x_{jt} değişkeni $[EFT_j, LFT_j]$ zaman aralığında tanımlanmıştır. Bu zaman aralıklarının, eniyi çözümü dışlamadan en dar şekilde tanımlanması değişken adedinin düşük tutulmasına katkıda bulunacaktır.

3.4.2 Kesin çözüm yöntemleri

Proje süresinin enküçüklendiği kaynak kısıtlı proje çizelgeleme problemi *NP-zor* olarak tanımlanmış bir problemdir (Ulusoy ve Özdamar, 1996). Bu nedenle, kesin çözüm yöntemleri ancak belirli problem büyüklükleri için geçerli olabilmektedir. İlk

kesin çözüm yöntemleri arasında 0-1 tam sayı programlama yöntemleri üzerinde de durulmuştur (Pritsker vd., 1969; Patterson ve Huber,1974; Patterson ve Roth, 1976; Talbot ve Patterson, 1978). Çok sayıda çalışmada dal-sınır yöntemi kullanılmaktadır. Stinson vd. (1978) bu yöntemin ilk uygulama örneğini sunmuşlardır. Bir düğümdeki kısmi çizelgede devam etmekte olan faaliyetlerin en erken biteninin bitiş zamanı, o düğümden çıkacak düğümlerin çizelgeleme zamanını belirler. Yeni yaratılan düğümler henüz atanmamış faaliyetlerin olurlu altkümeleridir. Düğüm seçimi hiyerarşik olarak uygulanan altı seçim kuralı ile gerçekleştirilir. Dal-sınır ağacını budamak için iki baskınlık kuralı kullanılmıştır. Demeulemeester ve Herroelen (1992), dikey inişli bir dal-sınır yöntemi uygulaması önermişlerdir. Aday faaliyet kümesinde yer alan faaliyetler, belirli bir çizelgeleme anında öncüllük ilişkilerine göre atanabilir durumda olan faaliyetlerdir. Dallanılan düğümden geçerli çizelgeleme anında aday faaliyet kümesindeki tüm faaliyetler atanır. Kaynak kısıtlarının ihlali durumunda, ertelenmeleri sonucu çizelgeyi kaynak kullanımını açısından olurlu kılacak en küçük faaliyet altkümeleri saptanır. Bu altkümeler, herhangi birisi diğer birisine baskın olmayacak şekilde belirlenir. En küçük ertelenen faaliyetler kümesi olarak nitelendirilen bu altkümelerin her biri dallanılan düğüme bağlı yeni bir düğümü oluşturur. Burada ilginç olan, en küçük ertelenen faaliyetler kümesindeki bazı faaliyetlerin dallanan düğümden çizelgede yer alabiliyor olmasıdır. Buna göre, bazı faaliyetler dal-sınır ağacı üzerinde dikey olarak aşağıya inerken önce kısmi çizelgeye dahil olup sonra çıkabilmektedir. Düğüm budanması için iki baskınlık kuralından yararlanılmaktadır. Bunlardan birincisinde, başka bir düğüm ile sola kaydırma sonucu aynı kısmi çözüme sahip olacak bir düğüm budanır. Dikey iniş için seçilecek düğüm en küçük kritik sıralama bazlı altsınıra sahip düğüm olarak belirlenmektedir. Birden fazla böyle düğüm bulunması durumunda seçim rassal olarak yapılır. Dikey iniş sonucu bir dal sonuna ulaşıldığında dallanılmamış bir düğüme ilk ulaşılanı kadar geri dönüş uygulanır.

3.4.3 Sezgisel yöntemler

Kesin çözüm yöntemleri ile çözüm sağlanabilen problem büyüklüklerinin kısıtlı olması ve ihtiyaç duyulan hesap sürelerinin büyüklüğü araştırmacıları daha büyük proje serimleri için eniyi çözüm olmazsa da “iyi” bir çözüme süratle ulaşan sezgisel yöntemler üzerinde çalışmaya yöneltmiştir. Sezgisel yöntemlerin sonuçları aynı

zamanda kesin çözüm yöntemleri için bir alt veya üst sınır oluşturmaları nedeni ile de önemlidir. Sezgisel yöntemleri dört ayrı başlık halinde incelenir:

- Öncelik kurallarına dayalı tek çözüm türeten sezgisel yöntemler
- Çok çözüm türeten sezgisel yöntemler
- Meta-sezgisel yöntemler
- Diğer Sezgisel Yöntemler

3.4.3.1 Öncelik kurallarına dayalı tek çözüm türeten sezgisel yöntemler

Öncelik kurallarına dayalı sezgisel yöntemlerin iki boyutu vardır: Kullanılan öncelik kuralı ve çizelgenin oluşturulma yöntemi. Bu başlık altında incelenen algoritmalar tek bir çözüm oluşturmaya yöneliktir ve tüm faaliyetler öncelik kuralları kullanılarak atandıktan ve böylece öncüllük ilişkileri ve kaynak kısıtları açısından olurlu bir çizelge oluşturulduktan sonra algoritmanın uygulaması sonlandırılır.

Teknik yazında çok sayıda sezgisel öncelik kuralı önerilmiştir ve bu kuralların performansına ilişkin olarak da çeşitli değerlendirme çalışmaları yapılmıştır. Bu kurallardan nispeten daha çok kullanılanlarını teknik yazındaki kısaltmalarını da kullanarak şöyle sıralayabiliriz (Paksoy, 2007):

1. *En Küçük Bolluk Değeri (Minimum Total Slack - MINSLACK):*

Bu kuralda amaç, en az bolluk süresine (slack time) sahip kritik faaliyeti belirlemektir.

$$P_j = LFT_j - EST_j - d_{jm}$$

Burada;

P_j : j faaliyetinin öncelik değeri

LFT_j : En geç bitirme zamanı

EST_j : En erken başlama zamanı

d_{jm} : j faaliyetinin m modundaki faaliyet süresidir.

2. *En Kısa En Geç Bitirme Zamanı (Minimum LFT):*

LFT kuralının amacı, proje bitirme zamanının artma riskini azaltacak, küçük LFT 'lere sahip faaliyetleri çizelgelemektir.

$$P_j = LFT_j$$

Burada;

P_j : j faaliyetinin öncelik değeri

LFT_j : En geç bitirme zamanıdır.

3. En Kısa İşlem Süresi (SPT):

SPT'nin amacı, en kısa faaliyet süreli faaliyeti göndermektir.

$$P_j = d_{jm}$$

Burada;

P_j : j faaliyetinin öncelik değeri

d_{jm} : j faaliyetinin m modundaki faaliyet süresidir.

4. Rastgele (Random):

Bu kural, seçilebilir koşullarını sağlayan faaliyetler arasından bir faaliyetin rastgele seçilmesini sağlamaktadır.

$$P_j = \text{RANDOM}(\text{ELIGSET})$$

Burada;

P_j : j faaliyetinin öncelik değeri

ELIGSET : Seçilebilir faaliyetler kümesidir

RANDOM : Bu fonksiyon, aday çözümler içeren faaliyetler kümesinden bir faaliyeti rastgele seçmektedir.

5. Ağırlıklandırılmış Kaynak Kullanım Oranı ve Ardılık (Weighted Resource Utilization and Precedence - WRUP):

WRUP kuralını kullanmada amaç, proje ağında yoğun kaynak ihtiyacında olan ve kendisinden sonra gelen faaliyetlerin (successors) fazla olduğu faaliyetleri mümkün olduğunca erken aşamalara çekerek kaynak darboğazını ortadan kaldırmaktır.

$$P_j = W_p * n_{pj} + (1 - W_p) * \sum r_{jmh} / R_h$$

Burada;

P_j : j faaliyetinin öncelik değeri

W_p : öncelik ağırlığıdır ve rastgele seçilir (precedence weight, ve $0 \leq W_p \leq 1$)

n_{pj} : j faaliyetinin ardıl/öncül faaliyetlerinin sayısıdır (ardıl-ileriye doğru çizelgelemede, öncül-geriye doğru çizelgelemede)

r_{jmh} : m modunda, j aktivitesi için gerekli h tipi (renewable) kaynak miktarı

R_h : h tipi kaynağın kaynak sınırlandırmasıdır.

6. En Kısa En Geç Başlama Zamanı (Minimum LST):

Bu kural da amaç, LST'leri erken olan faaliyetleri belirlemek ve böylece kritik faaliyetlerin ertelenmesinden doğacak proje gecikme sürelerini elimine etmektir.

$$P_j = LFT_j - d_{jm}$$

Burada;

P_j : j faaliyetinin öncelik değeri

LST_j : En geç başlama süresidir.

7. En Kısa En Erken Başlama Süresi (Minimum EST):

Bu kural da amaç, EST'leri erken olan faaliyetleri belirlemek ve böylece kritik faaliyetlerin ertelenmesinden doğacak proje gecikme sürelerini elimine etmektir.

$$P_j = EST_j$$

Burada;

P_j : j faaliyetinin öncelik değeri

EST_j : En erken başlama süresi

8. En Kısa En Erken Bitirme Süresi (Minimum EFT):

Bu kural da amaç, EFT'leri erken olan faaliyetleri belirlemektir.

$$P_j = EST_j + d_{jm}$$

Burada;

EST_j : En erken başlama zamanı

d_{jm} : j faaliyetinin m modundaki faaliyet süresidir.

9. En Fazla Faaliyet Sayısı (Most Jobs Possible - MTS):

MTS'nin amacı, j den projenin son faaliyetine kadar olan yolda, en fazla sayıda faaliyete sahip faaliyetleri belirlemektir.

$$P_j = allons_j$$

$allons_j$: j faaliyetinden sonra gelen faaliyetlerin sayısıdır.

Geri çizelgeleme iterasyonunda $allons_j$, j faaliyetinden önce gelen faaliyetlerin sayısını temsil etmektedir. Bunlara ek olarak, literatürde çok kullanılan diğer kurallar ise (Boctor,1996,s.351);

- MAX RWK: Faaliyetin süresi ile kendinden sonra gelen faaliyetlerin sürelerinin toplamı en büyük olan faaliyetlerin seçilmesi kuralıdır.
- MAX NIS: Kendinden önce gelen faaliyet sayıları toplamı en büyük olan faaliyetin seçilmesi kuralıdır.
- MAX PTM: En fazla faaliyet süresine sahip olan faaliyetin seçilmesi kuralıdır.
- MIN PTM: En az faaliyet süresine sahip olan faaliyetin seçilmesi kuralıdır.
- MAX CAN: Faaliyetin seçiminden sonra, seçilebilir faaliyet sayısı en fazla olan faaliyetin seçilmesi kuralıdır.

Öncelik kuralları çizelgeleme aşamalarında değişik değer almaları ya da çizelgeleme boyunca sabit değer almalarına göre farklılık göstermektedirler. Örneğin SPT durağan bir kural iken, WRUP dinamik bir kuraldır (Ulusoy, 2006). Faaliyetlerin önceliklerini belirleme aşamasında kullanılan bu kuralların bazılarını deterministik bazılarını da olasılığa dayalı kurallar olarak da tanımlamak mümkündür.

Öncelik kurallarında kullanılan *EST*, *LST*, *EFT*, *LFT* ve Slack (bolluk) değerleri; kritik yol metodu ile hesaplanan Öncelik kurallarında kullanılan *EST*, *LST*, *EFT*, *LFT* ve Slack (bolluk) değerlerdir (Simpson ve Patterson,1996). Bu nedenle de durağan öncelik kurallarıdır. Durağan öncelik kuralı, her bir KKPÇP'de, bir kereye mahsus olarak hesaplandığından algoritmanın işlem süresini artırmayan kurallardır. Ancak dinamik kurallar, her bir faaliyetin seçim aşamasında yeniden hesaplanarak dikkate alınacağından, algoritmanın işlem süresini etkilemektedirler.

3.4.3.2 Çok çözüm türeten sezgisel yöntemler

Çok çözüm türeten sezgisel yöntemleri üç başlık altında inceleyebiliriz: (i). Örnekleme, (ii) öncelik kuralları (iii) Başlangıç-bitiş çıkışlı iteratif çizelgeleme. Çok çözüm türeten sezgisel yöntemlerde önceden belirlenmiş sayıda çizelgeleme türetilerek

aralarından en iyisi seçilir. Bir çözüm elde etmenin çözüm süresinin azlığı, algoritmayı çok kere uygulamayı hesap süresi bakımından olurlu hale getirmiştir.

3.4.3.3 Meta-Sezgisel Yöntemler

Meta-sezgisel yöntemlerin genelde kombinatoriyal eniyilemede özelde ise proje ve makine çizelgelemede giderek daha yaygın bir kullanımı vardır. Meta-sezgisel yöntemler; (i). Genetik algoritmalar; (ii). Tavlama benzetimi yöntemi (iii). Yasaklı arama yöntemi olarak sınıflandırılabilir. Meta-sezgisel yöntemler arasında sinir ağları (*neural networks*) ve karınca kolonileri de (*ant colonies*) bulunmaktadır.

Meta-sezgisel yöntemlerde, öncelikle, çözümün gösterimi ve bu gösterimi çizelgeye dönüştüren çizelgeleme yöntemleri bulunmaktadır. Bu gösterimlerden bir tanesi, rassal anahtar gösterimidir. Rassal anahtar gösteriminde, gösterimin her bir elemanı bir faaliyete ilişkin bir değere karşı gelir. Faaliyet listesi gösterimi olarak nitelendirilen gösterimde, gösterimin her bir elemanı bir faaliyetin proje serimi üzerinde atanmış gösterim sayısına karşı gelir. Üçüncü gösterimde ise, gösterimin elemanlarını öncelik kuralları oluşturur: Bu gösterime öncelik kuralları gösterimi denir. Bu gösterim daha önce atölye çizelgeleme problemlerinde kullanılmıştır (Dorndorff ve Pesch, 1995 ve Herrmann vd., 1995). Çözüm gösteriminin çizelgeye dönüştürülmesinde seri ve paralel çizelgeleme kullanılır.

Genetik algoritma. Genetik algoritmalar Holland (1975) tarafından biyolojik evrim sürecinden hareketle önerilmiştir. Burada tanıtılan diğer iki meta-sezgisel yöntemden farklı olarak, genetik algoritma tek bir çözüm yerine aynı anda birden fazla sayıda çözümü değerlendirir. Bu özellik, genetik algoritmanın paralel arama özelliği olarak nitelendirilir. Problem çözümü genlerden oluşan bir kromozom ile ifade edilir. Genetik algoritma, bir dizi çözümün oluşturduğu bir kuşak ile başlatılır. Güncel kuşak içindeki kromozomlara kopyalama, çaprazlama ve mutasyon operatörleri uygulanarak yeni bir kuşak oluşturulur. Yeni kuşak oluşturulurken uygulanan operatörlerde “daha iyi” kromozomların bir sonraki kuşağa aktarılması ve yeni kromozomların oluşturulmasındaki etkinliklerinin olasılıkları uygunluk değerlerine doğrudan bağlıdır. Daha iyi kromozomlar, daha yüksek uygunluk değerlerine sahip olan kromozomlardır.

Bu şekilde, biyolojik evrim sürecinin “güçlünün yaşamını sürdürmesi” anlayışı genetik algoritmaya yansıtılmaktadır.

Tavlama benzetimi. İlk defa Kirkpatrick vd. (1983) tarafından sunulan tavlama benzetimi yöntemi, ilk karşılaşılan daha iyi çözümü kabul ederek eniyiyi arama stratejisinin temelde bir değişik uygulama biçimidir. Her adımda daha iyi bir noktaya gitmeyi hedefleyen gradyan ile arama yönteminden farklı olarak tavlama benzetimi yöntemi süreç içinde giderek azalan bir olasılıkla daha kötü bir çözüme gitmeye izin verir. Tavlama benzetimi yöntemi ile bir çözüm önerisi Boctor (1996) tarafından sunulmuştur. Önerilen algorithmada, faaliyet listesi kullanılarak seri çizelgeleme uygulanmıştır. Komşu çözüm kümesinin tanımlanması ve aranmasında, Sampson ve Weiss (1993) tarafından önerilen öteleme vektörü kavramı kullanılmıştır. Sampson ve Weiss, herhangi bir çözümün gösterimi amacı ile $(1 \times J)$ boyutunda ve elemanları tam sayı olan bir öteleme vektörü, V , tanımlamışlardır. Buna göre, j faaliyetinin başlangıç anı bu faaliyetin öncüllerinin en geç biteninin bitiş anı ile V vektörünün j faaliyetine karşı gelen $V(j)$ elemanının değerinin toplamı ile ifade edilir. Her j faaliyeti için $V(j)$ elemanının alabileceği değerler üzerinden bir arama yapılır.

Yasaklı Arama. Glover (1989a, 1989b) tarafından geliştirilen yasaklı arama temelde güncel çözümün çevresinde yerel eniyi çözümü arayan gradyan yönteminin bir uyarlamasıdır. Algoritmanın bir yerel eniyi çözümden ayrıldıktan hemen sonra tekrar aynı yerel en iyi çözüme yönelmemesi için algoritmanın attığı son adımlar yasaklı olarak ilan edilir. Yasak listesi dinamiktir. Her yeni bir eleman yasak listesine girdiğinde, yasak listesinde en çok kalmış olan eleman liste dışına çıkarılır. Bu şekilde algoritmaya bir hafıza kazandırılmış olur.

3.4.3.4 Diğer sezgisel yöntemler

Diğer sezgisel yöntemler aşağıdaki gibidir;

- Yerel kısıt analizi
- Yerel arama
- Budanmış dal-sınır yöntemi

3.5 Çok Modlu Kaynak Kullanımlı Projelerde Proje Süresinin En Küçüklenmesi Problemi

Bu bölümde çok modlu kaynak kullanımlı projelerde proje süresinin en küçüklenmesi probleminin tanımı ve matematiksel gösterimi, kesin çözüm yöntemleri ve ardından metasezgiseller anlatılmıştır.

3.5.1 Problemin tanımı ve matematiksel programlama gösterimi

Problemin tanımı, KKPCP'nden sadece kaynak kullanımında birden fazla modun varlığına izin verilmesi ile farklılaşır. Bu şekilde, problem tanımının daha gerçekçi olması sağlanmıştır. Aşağıdaki gösterimde mod indisi m kullanılarak değişken ve parametrelere mod boyutu eklenmiştir. Yenilenemez kaynakların matematiksel gösterimini örneklemek amacı ile aşağıdaki gösterime yenilenemez kaynaklar da dâhil edilmiştir.

Değişkenler:

$$X_{jmt} \in \{0,1\} \quad j = 1, \dots, J; \quad m = 1, \dots, M_j; \quad t = EFT_j, \dots, LFT_j \quad (3.6)$$

Kısıtlar:

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

$$\sum_{j=1}^J \sum_{m=1}^{M_j} k_{jmr} \sum_{\tau=t}^{t+d_{jm}-1} X_{jm\tau} \leq K_r \quad (3.8)$$

$$\sum_{m=1}^{M_i} \sum_{t=EFT_i}^{LFT_i} t X_{imt} \leq \sum_{m=1}^{M_i} \sum_{t=EFT_j}^{LFT_j} (t - d_{jm}) X_{jmt} \quad j = 2, \dots, J, \quad i \in P_i \quad (3.9)$$

$$\sum_{j=1}^J \sum_{m=1}^{M_j} k_{jmr} = \sum_{t=EFT_j}^{LFT_j} X_{jmt} \leq K_r \quad r \in N \quad (3.10)$$

Amaç Fonksiyonu:

$$\text{enk } Z = \sum_{t=EFT_j}^{LFT_j} tX_{jmt} \quad (3.11)$$

Yukarıda (3.11) probleminde kullanılan semboller dışında kalanlar için tanımlar aşağıda verilmiştir:

$M_j = j$ faaliyetinin mod adedi,

$m = \text{mod indisi } (m=1, \dots, M_j)$,

$N = \text{yenilenemez kaynakların kümesi.}$

(3.11) probleminde, amaç işlevi proje süresini en küçükmektir. (3.7) kısıt kümesi her faaliyetin mutlaka çizelgelenmesini sağlar. Kısıt kümesi (3.8) yenilenebilir kaynakların birim zaman içinde kullanımlarının üst sınırını gösterir. Kısıt kümesi (3.9) ise, j faaliyeti ile bu faaliyetin öncülü olan i faaliyeti arasındaki öncüllük ilişkisinin yerine getirilmesi gereğini ifade eder. (3.10) kısıt kümesi yenilenemez kaynakların birim zaman içinde tüketimlerinin üst sınırını gösterir. (3.6) ise $\{0,1\}$ değişkeni x_{jmt} 'nin tanımıdır. Karar değişkeni x_{jmt} 'nin üç indisli olması değişken adedini tek modlu gösterime göre büyük oranda artırmaktadır. Bu nedenle, $[EFT_j, LFT_j]$ zaman aralıklarının eniyi çözümü dışlamadan en dar şekilde tanımı bu gösterimde daha da önem kazanmaktadır.

3.5.2 Kesin çözüm yöntemleri

Talbot (1982) faaliyet listesi üzerinde tanımladığı bir birerleme yöntemi önermiştir. Bu yöntemde, faaliyet listesinde bir sonraki faaliyet en hızlı modunda atanmakta ve en erken zamanda başlatılmaktadır. Eğer faaliyet zaman aralığı içinde atanamazsa, yöntem bir önceki öncüle geri dönmekte ve o faaliyetin daha geç bir

zamanda veya daha yavaş modunda atanması olanağını araştırmaktadır. Patterson vd. (1989) bu yaklaşımı geliştirmişler ve sayısal olarak da irdelenmişlerdir. Bu ilk çözüm önerileri 15 faaliyete kadarki problemlere çözüm getirebilmiştir.

3.5.3 Meta-sezgisel yöntemler

Genetik algoritma. Mori ve Tseng (1997) sadece yenilenebilir kaynakların bulunduğu KKPCP'ni ele almışlardır. Çalışmada, kromozomun her geninde bir faaliyet numarası, bu faaliyet için atanmış mod, çizelgeleme sırası ve başlama-bitiş zamanları bilgilerini taşıyan bir kodlama uygulanmıştır. Sundukları bu GA yaklaşımını Drexl ve Grünwald'ın (1993) sundukları stokastik bir yaklaşımla karşılaştırarak, GA'nın daha iyi sonuçlar verdiğini belirtmektedirler.

Hartmann (1997), hem yenilenebilir hem de yenilenemez kaynakların yer aldığı bir GA yaklaşımı sunmuştur. Bu çalışmada kromozom, öncüllük ilişkileri bakımından olurlu bir faaliyet sıralaması ve bu faaliyetler için seçilmiş birer modun bulunduğu bir kodlama ile sunulmaktadır. Bu yaklaşım için farklı bir dizi deneysel çalışmalar yapılarak en iyi olanı seçilmiş ve bu GA yaklaşımını literatürde raporlanmış bulunan üç sezgisel yöntemle karşılaştırılmıştır. Sunulan yeni yaklaşımın, eniyi proje süresinden ortalama sapma kriterine göre diğer üç sezgisel yöntemden daha iyi sonuçlar verdiği gösterilmiştir.

3.6 Kaynak Hareketli Çizelgeleme

Kaynak hareketli çizelgeleme yapı yönetiminin en önemli konularından biridir. Süreleri kaynak eklemelerinden ve çıkarmalarından etkilenen görevler kaynak hareketli görevler olarak adlandırılırlar. Gerçek dünya projelerinde, bu hesaplama nadiren kesindir, çünkü insanlar değişik seviyelerdeki yeteneklere sahiptir ve bir işi farklı hızlarda tamamlar. İki insan hiç bir zaman bir görevin zamanını tam anlamıyla yarıya düşüremez (Marmel, 2004). Geleneksel programlama tekniklerinden farklı olarak, kaynak hareketli çizelgeleme direk olarak etkin kaynak kullanımı yanında takım çalışması devamlılığını açıklar (El-Rayes ve Moselhi, 1996). Projenin içinde yer alan

kaynakları modellemede Gordon ve Tulip (1997) tarafından aktarılan beş temel kategori bulunmaktadır:

- a) Kaynak birleştirilmesi
- b) Kaynak biriktirme
- c) Kaynak dağılımı
- d) Kaynak düzleştirme
- e) Kaynak dengeleme

3.6.1 Kaynak birleştirilmesi

Bütün kaynak programlama prosedürlerinin en basiti toplamadır. Projenin hayata geçirilme süreci boyunca birim zaman tabanında gerekli kaynak birimlerinin toplam miktarını belirleme metodudur. En sık kullanılan zaman birimi gündür (Gordon ve Tulip, 1997). Kaynak toplama muhtemel taşıma oranlarıyla belirli bir yerde oluşturulan proje faaliyetleriyle proje süreci boyunca her bir kaynak için zaman dönemlerinde gerekenlerin basit bir özetidir.

3.6.2 Kaynak biriktirme

Bu yaklaşım projenin hayata geçme sürecinde kaynak birikimi sağlar. Bu işlemin girdisi toplama hesaplamalarının çıktısıdır. Biriktirmede toplama sonuçları akar tabanda biriktirilir (Gordon ve Tulip, 1997).

Projenin finansal başarısı, büyük ölçüde proje yöneticisinin kaynakları verimli şekilde istihdam etme yeteneğine dayanır. Eğer kaynaklar verimsiz istihdam edilirse, proje için ekstra maliyetler söz konusu olur. Kaynaklarının kullanımında en verimsiz proje, en çok ekstra maliyet gerektiren projedir. Maliyetleri düşüren kaynakların en yüksek verimlilikle kullanımını garantilemek bir proje yöneticisi için zorlayıcı bir durumdur.

Bu durum;

- Düzgün ve sürekli bir iş akışı,

- Kesintisiz sürekli çalışma (boş kalma maliyeti),
- Yeterli tahsis edilmiş kaynak hacmi,
- Becerikli çalışan kaynaklarının, teknolojik olarak yeterli yer/donanım kaynaklarını ve yüksek kalitede malzeme kaynaklarının uygun bir şekilde istihdamını,
- Doğru çalışan ve yer/donanım karışımının istihdamını, gerektirir.

3.6.3 Kaynak dağılımı

Kaynak yükleme diye de bilinen kaynak dağılımı, faaliyetleri işi uygulamak için ihtiyaç duyulan çalışan, ekipman ve malzemeyi tür ve kalite olarak ayıran işlemdir. Sonuç olarak, beklenen kaynak ihtiyacını ve doğru planlamayı belirleyebilir (Jenkins ve Orth, 2002). Programdaki her bir görev için çalışan, ekipman ve malzemelerin dağılımını yaparak program faaliyete geçmeden önce biri her iş günü için beklenen kaynak ihtiyaçları ön planını yapabilir. Daha sonra bu toplam kaynak gereksinimi halihazırdaki miktarla karşılaştırılır. Mevcut kaynak gerekeninin üzerindeyse, program planlandığı gibi çalışılabilir. Mevcut olandan daha fazla kaynağa ihtiyaç tespit edilirse, ihtiyaç duyulan zamanda elde bulunacak yeterli kaynak miktarını sağlamak için adımların atılması gerekir (Jenkins ve Orth, 2002). Kaynak dağılımı, yönetim kısıtları içinde projenin tamamlanması için uygulanabilir program sağlamada kullanılır. Kaynak dağılımı, kaynak erişiminde kesit kısıtlar olduğunda kullanılır. İşlem, belirli bir dönemde kısıtlı kaynakların nasıl dağıtılacağıyla ilgili bir kriter listesine dayanır. Kaynak erişiminin belirli bir faaliyetin ihtiyaç duyduğu kaynakla karşılaştırılması gerekir. Yetersiz kaynak erişimi varsa, faaliyet gerekli olmayan kaynaklara göre yeniden programlanmalıdır. İki faaliyetin aynı anda aynı kaynağa gereksinim duyduğu durumlarda kaynak faaliyetlere planlandığı gibi aynı anda başlamaya yetersizse, daha önemli olan faaliyet nadir bulunan kaynağı kullanmada öncelikli olur (Khatab ve Soyland, 1998).

3.6.4 Kaynak düzleştirme

Kaynak düzleştirme tek biçim kaynak gereği profili ya da mümkün olduğu kadar planlanmış kaynak kullanımıyla zaman kısıtı içinde uygulanabilir bir program oluşturma tekniğidir (Lockyer ve Gordon, 1997). Dağılım rutinlerinde zaman kısıtlı program oluşturmak oldukça zordur. Düzleştirme prosedürü kullanıcı tarafından oluşturulmuş zaman dilimleri içinde çalışır ve bütün projeye bakar. Bütün faaliyetler programlanana kadar tekrar edilen dört adım vardır:

- Herhangi bir kritik faaliyeti programlama
- Programlanacak en önemli faaliyeti bulma
- Bu faaliyeti programlamak için en uygun yeri bulma
- Yeni programlanmış faaliyeti hesaba katmak için programlanmamış faaliyetlerin erken ve geç zamanlarını düzenlemek (Gordon ve Tulip, 1997).

Düzleştirme oluşması için herhangi bir önceki kaynak programına gerek yoktur. Başlangıç tarihi, belirlenecek proje süreci ve göz önünde bulundurulması için her bir kaynağın önceden belirlenmiş olmasını gerektirir. Prosedür, daha sonra proje zaman dilimiyle mümkün olan kaynak kullanımını tek biçim profili vermesi için faaliyetleri programlar (Lockyer ve Gordon, 1997)

3.7 Çoklu Proje Yönetimi

Çoklu proje yönetimi, tek bir projenin yönetiminden oldukça farklıdır. Tek bir proje yöneticisi sadece bir projenin entegrasyon, maliyet, zaman, kalite, insan kaynakları, iletişim, risk ve tedarik işlerinden sorumludur. Çoklu proje yöneticisi tüm bunlarla aynı anda birden çok proje için savaşmaktadır. Birçok örnekte projeler birçok amaçta, farklı sorumlularla, çeşitli safhalardadır. Üst yönetim tahsis edilen zaman, maliyet ve bütçe içinde her bir projeden eşit başarı bekler. Yönetim;

- Proje uygulama süresince görünürlük,
- Proje süresini azaltmada karşıt organizasyonlar arasında koordinasyon,
- Kısıtlı kaynakla üretilen iş miktarında artış olduğunu görmek ister.

Geleneksel proje yönetimi bu meydan okumaya rağmen eksiktir ve çoklu proje yaklaşımı organizasyonel stratejiyle çakışmaları çözmek, ihtiyaçları dengelemek, kaynakları düzenlemek amacıyla geliştirilmiştir.

Proje Yönetimi disiplinin altındaki metodolojik yapılanmayı (tanımla - planla - uygula -kontrol et) tek bir projeye uygulamakla, birden fazla projeye uygulamak açısından teoride hiçbir fark yoktur. Yeni geliştirilen Proje Yönetimi yazılımları (MS Project 2000/Central -MS Project 2002, Project Gateway gibi) kuruluşta tek bir elden projelerin yönetilebileceği varsayımı üzerine hazırlanmıştır.

Tüm insan, makine, malzeme ve para kaynağını değişen önceliklere göre yönlendirebilen yazılımlar, çoklu proje uygulayan firmalar için zorunlu hale gelmektedir (Ünle, 2007).

3.7.1 Kaynak kısıtlı birden çok projenin çizelgelenmesi

Kaynak kısıtlı birden çok projenin çizelgelenmesi uygulamadaki gerçek durumu yansıtması bakımından kaynak kısıtlı proje çizelgelemede önemli bir araştırma alanıdır. Örneğin, değişik şantiyelerde iş yapan bir inşaat şirketi bu şantiyelerde ortak ve kısıtlı kaynaklarını kullandığından tipik olarak bu proje çizelgeleme modeline uygun bir görünüm sergiler. Aynı şekilde, bir otomotiv firmasının aynı anda birden fazla ürün geliştirme projesinde ortak ve kısıtlı kaynaklarının kullanıldığı ortamın planlanması da kaynak kısıtlı birden çok projenin çizelgelenmesini gerektirir. Kaynak kısıtlı birden çok projenin çizelgelenmesi için iki temel yaklaşım vardır:

- i. Her projenin kendi başına ele alınması (diğer bir deyişle, her projenin kendi başlangıç ve bitiş düğümünün olması);
- ii. Tüm projelerin bir proje halinde bütünleştirilmesi (diğer bir deyişle, tüm projeler için bir başlangıç ve bir bitiş düğümü olması).

Çizelgeleme için kullanılan sezgisel kurallar, genelde, iki ayrı modelleme için farklı değerler verecektir. Genelde, zaman bazlı amaçlar üzerinde çalışılmıştır. Çizelgeleme için amaç işlevi iki tür modelleme için de temelde aynıdır: kaynakların

kısıtlı olması sonucu oluşan gecikmenin, yani kaynak kısıtlı çizelgeleme ile elde edilen proje süresi ile kaynak kısıtsız (CPM) proje süresi arasındaki farkın bir işlevidir.

Her projenin kendi başına ele alınması durumunda, gecikme her bir proje için ayrı ayrı hesaplanır ve proje sayısına bölünerek ortalaması alınır (MPD- *mean project delay*). Tüm projelerin tek bir proje halinde birleştirildiği durumda ise, bu tek proje boyutunda gecikme hesaplanır ve tek proje boyutundaki CPM proje süresine bölünerek oransal bir değer olarak kullanılır (MDI- *multiproject duration increase*).

Tüm projelerin tek bir proje halinde bütünleştirilmesi durumu için tek proje çizelgelemede sürenin en küçüklenmesinde başarılı olmuş LFT, LST, MSLK gibi sezgisel kurallar kullanılmıştır.

Kaynak kısıtlı birden çok projenin çizelgelenmesi için önerilmiş sezgisel kuralların bazılarını şöyle listeleyebiliriz:

- i. En büyük toplam iş kapsamı (MAXTWC- maximum total work content).
- ii. En kısa süreli projenin en kısa süreli faaliyeti (SASP- shortest activity from shortest project).
- iii. En kısa boşluk süresi (MSLK-MP- minimum total slack).
- iv. En geç bitirme zamanı (MINLFT– MP- minimum latest finish time).

Her projenin kendi başına ele alındığı modelleme durumu için, tek proje için geliştirilmiş sezgisel kuralların bu duruma uygulamasına göre daha iyi sonuçlar vermiştir (Kurtulus ve Davis, 1982; Kurtulus ve Narula, 1985). Yine aynı araştırmacılar, Tsubakitani ve Deckro (1990) ev inşaatlarında birden çok projenin çizelgelenmesi ve kontrolü için dinamik bir algoritma önermişlerdir. SASP sezgisel kuralını kullanarak her biri 100'den fazla faaliyet içeren 50'den fazla projeyi bir arada çizelgelemişlerdir. Bir UPDATE modülü ile proje yöneticisine proje planını güncelleme olanağı sağlanmıştır. Faaliyet bölme ve uygulanan bir faaliyeti durdurarak uygulamasını erteleme, öncüllük ve kaynak kısıtı gibi hususlarda güncel durumu yansıtılabilmek için bunları göz ardı edebilme olanakları proje yöneticisine sağlanmıştır.

Projeler için termin belirlenmesi pratik açıdan önemli bir problemdir. Dumond ve Mabert (1988) projelerin değişik zamanlarda sisteme girebildiği dinamik bir ortamı incelemişlerdir. Ortalama sapma, ortalama bitiş zamanı ve toplam gecikme amaç işlevlerini irdelenmişlerdir. Dumond ve Mabert, MSLK, LFT, SASP gibi sezgisel çizelgeleme kurallarını dinamik ortama uyarlayarak kullanmışlar ve değişik termin belirleme kurallarını sınımışlardır. Terminlerin proje yüküne göre belirlendiği durumlarda FIFO sezgisel kuralının diğerlerine göre daha iyi sonuçlar verdiğini rapor etmişlerdir. Bock ve Patterson (1990), Dumont ve Mabert'in elde ettiği sonuçları daha değişik proje ortamlarında sınımışlar ve benzer sonuçlara ulaşmışlardır. Faaliyetlerin yarıda kesilip ertelendiği durumlarda uygulanacak politikaların termine uyabilme konusunda kritik rol oynadığını göstermişlerdir.

Lawrence ve Morton (1993) her projenin gecikmesinin farklı bir maliyeti olması durumunu irdelenmişlerdir. Bu amaçla, ağırlıklı gecikme maliyetinin en küçüklenmesi için kaynak kullanımının yoğunluğuna bağlı kaynak fiyatlandırma politikaları kullanmışlardır. Lova ve Tormos (2000), her projenin kendi başına ele alındığı modelleme durumu için iki aşamalı bir sezgisel algoritma önermişlerdir. Atanacak faaliyetin saptanması için ilk aşamada toplam iş kapsamı (TWK) sezgisel kuralı kullanılarak proje seçilmiş, ikinci aşamada ise, GRD, en erken başlama zamanı (EST), ilk gelen ilk önce (FCFS) ve LST sezgisel karar kurallarında birisi kullanılarak atanacak faaliyet belirlenmiştir. Yapılan sayısal karşılaştırma çalışmalarında iki aşamalı algoritma MAXTWK sezgisel kuralı kullanan tek aşamalı sezgisel algoritma ile karşılaştırılmış ve iki aşamalı algoritmanın genelde daha iyi sonuçlar verdiği görülmüştür. Yukarıda sayılan sezgisel kurallar arasında ikinci aşamada LST'nin en iyi sonucu verdiği belirlenmiştir. Sezgisel kurallardan hiçbirisinin diğerlerinden çok öne çıkması sonucu, Lova ve Tormos (2000), TWK-LST, TWK-FCFS ve TWK-EST kombinasyonlarının uygulanarak elde edilen en iyi çözümün nihai çözüm olarak önerildiği bir uygulama önermişlerdir. Tek proje çizelgeleme ortamındaki birden çok geçiş uygulaması ile paralel bir uygulama olan bu uygulamanın daha çok hesaplama süresi kullanarak daha iyi sonuçlar verdiğini göstermişlerdir.

Bouleimen ve Lecocq (1999) birden çok proje çizelgeleme ortamı için birçok amaçlı sezgisel yöntem önermişlerdir. İlginç bir yaklaşımla, projeler amaç olarak ifade

edilmişlerdir. Bu şekilde, karar vericinin projeler arasında bir öncelik sıralaması yapabilmesine olanak sağlanmıştır. Kaynak kullanımı çok modludur. Gösterimde yenilenemez kaynaklar da içirilmektedir. Dinamik bir ortam söz konusudur. Diğer bir deyişle, projelerin hepsinin sıfır anında hazır olması gerekmekte; projeler değişik zamanlarda sisteme girebilmektedir.

3.7.2 Kaynak kısıtlı çoklu proje çizelgeleme için kurallar

Kaynak kısıtlı proje çizelgeleme probleminin amacı, proje tamamlanma zamanının en küçüklenmesi amacıyla projenin her bir faaliyeti için uygun başlama zamanını belirlemektir. Tanımlanması çok kolay fakat çözümü çok zor olan bu problem $m,1/cpm/C_{max}$ şekli ile gösterilebilir (Herroelen vd., 1998). Çoklu proje modelinin matematiksel gösterimi Cristofides et al.(1987) tarafından şöyle tarif edilmiştir.

Değişkenler:

$$S_{ijt} \in (0,1), \quad (3.12)$$

Kısıtlar:

$$\sum_t S_{ijt} = 1, \quad i = 1, \dots, M \quad J = 1, \dots, i_i + 1, \quad (3.13)$$

$$\sum_t t(S_{imt} - S_{ijt}) \geq d_{ij}, \quad (j,m) \in H_i \quad i = 1, \dots, M, \quad (3.14)$$

$$\sum_{i=1}^M \sum_{j=1}^{J_i} \sum_{q=t-d_{ij}+1}^t r_{ijk} S_{ijq} \leq R_k, \quad k = 1, \dots, K, \quad t = 1, \dots, T, \quad (3.15)$$

Amaç Fonksiyonu:

$$Enküçük \quad f(süre) \quad (3.16)$$

Burada S_{ijt} , proje i 'de faaliyet j 'nin t zamanında başladığı durumlarda 1, diğer durumlarda 0 değerini alır. M , çoklu projede, proje numarasını; j_i proje i 'de faaliyet numarasını ve K 'da yenilebilir kaynak numarasını temsil eder. Amaç fonksiyonu (3.16), zaman ölçütünü minimum eden performans ölçümüdür. Kısıt (3.13) her bir faaliyetin bir kere başlamasını işaret eder. Kısıt (3.14) öncelik kısıtlarını temsil eder, burada H_i proje i 'de öncelik kısıtlı bir dizi faaliyettir. Faaliyet J_{i+1} , proje i 'de tamamlanma süresi olanaklı kukla faaliyete uyar. Kısıt (3.15) her bir k çeşit kaynak ve t anı için sınırdır. Faaliyetlerin kaynak talebi halen kullanıldığı için R_k sınırını aşmaz. r_{ijk} proje i 'de faaliyet j 'nin k kaynak gereksinimidir. Son olarak kısıt (3.16) ikili karar değişkenlerini tanımlar.

Öncelik kuralları üzerine sezgisel tabanlı yaklaşım bu problemin çözümü için en önemli çözüm tekniklerine sahiptir. Çünkü bu metotları bir proje yönetim yazılımı içerisine bağlamak çok kolaydır. Bu metotlar ile geniş ölçekli projeler için bile sıklıkla kabul edilebilir sonuçlar elde edilebilir (Kolisch, 1996). Bu teknikler ticari proje yönetimi yazılımının içerisinde bir sıra ile birleştirilmiştir, fakat her zaman aynı performans sağlanamamaktadır.

Öncelik kuralı tabanlı metotların her zaman en iyi sonucu vermemesine rağmen büyük problemlerin kısa sürede çözümü için vazgeçilmezdirler. Öncelik tabanlı metotların iyi bir uygulamasını elde etmek için örneğin tabu araştırması, genetik algoritmalar gibi yöntemlerin başlangıç çözümlerini belirtmek gerekmektedir. Her ne kadar çoklu proje tamamlanma zamanı veya eş zamanlı analizlerin projelere etkisi üzerine çalışmalar olmasa da aşağıdaki örnekler verilebilir.

- Öncelik kuralları üzerine kurulmuş sezgisel taban
 - Çizelgeleme planları üretmek
 - Öncelik kuralı
- Yaklaşımlar: çoklu ve tekli projeler

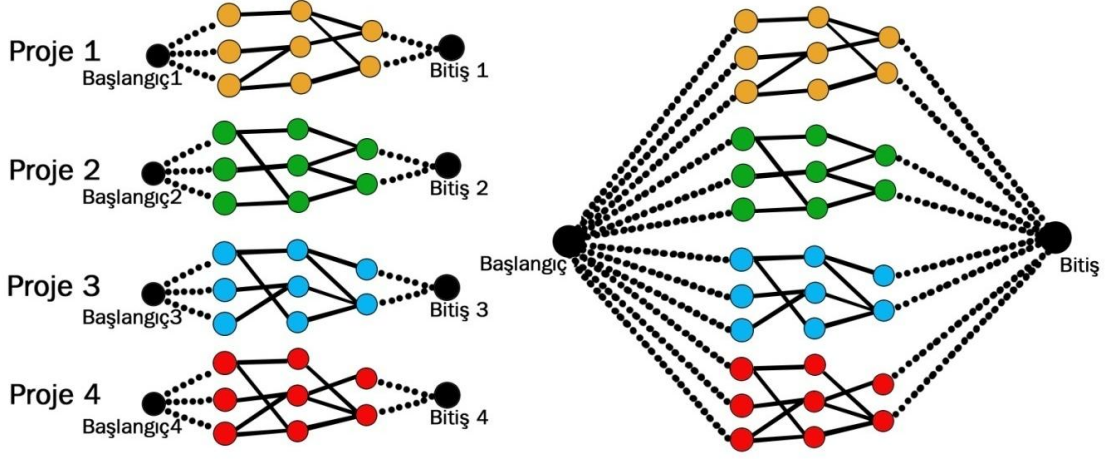
3.7.3 Çoklu proje çizelgeleme

Yaygın ve yenilenebilir kaynaklar ile çoklu projelerin sezgisel yollar ile çizelgelenmesini içeren birkaç çalışma vardır. Bu çalışmalar içerisinde problem çoğunlukla öncelik kuralları üzerine kurulmuş sezgisel tabanlar kullanılarak çözülmüştür (Bock ve Patterson,1990; Dumond ve Mabert, 1988; Fendley, 1968; Kurtulus ve Davis,1982). Bu çalışmalarda birçoklu çizelgeleme problemi özel bir yaklaşım kullanılarak çözülmüş ve öncelik kuralları üzerine kurulmuş çeşitli sezgisel tabanlı yöntemler değerlendirilmiştir.

Dantzig-Wolfe yöntemi uygun kaynak profillerini ve aralık çözümlerini sağlamaktadır. Karar verici kendi deneyimi ve risk toleranslarına bağlı olarak en iyi çözümün yanı sıra çözümlerden herhangi birini seçebilir. Yine de sezgisel alanda önceki çalışmalarda, yaklaşımın etkisi, çizelgeleme planı üretmek ve öncelik kurallarının performans ölçülerindeki etkilerine bağlı olarak test edilmiş, fakat eş zamanlı olarak hiç yapılmamıştır.

3.7.3.1 Yaklaşımlar ve çizelgeleme amaçları

Kaynaklarını ortak bir havuzda paylaşan çoklu projeleri çizelgelemek, iki yaklaşım ile gerçekleştirilmiş olabilir: *çoklu projeler* ve *tekli projeler*. Temsilci bir (activity on node-AON) serim ağı kullanılarak önceki çalışmalarda her projenin kukla faaliyetlerinin başlangıç ve bitiş zamanları uyumlu olarak ele alınmıştır. Sonraki çalışmalarda tekli bir projenin başlangıç ve bitiş olarak tanımlanan iki kukla faaliyet eklenmesi ile projeler tek bir proje içerisinde yapay olarak birleştirilmiştir. Her iki yaklaşımda aynı sezgisel tabanlı öncelik kuralını kullanarak farklı sonuçlar üretebilir. Aslında eğer uygulanan öncelik kuralı kritik yol uzunluğuna bağlıysa tekli proje yaklaşımı uygulandığı zaman faaliyetlerin parametreleri tekli projenin süresinden elde edilmektedir. Ancak çoklu proje yaklaşımında bu parametreler her bir bireysel projenin süresinden elde edilmektedir. Uygun bir çoklu proje çizelgelemesi elde etmek için bu iki yaklaşımdan birini kullanabilir (Kurtuluş ve Narula, 1985) Şekil 3.1'de bu yaklaşımlar gösterilmiştir.



(a) Çoklu proje yaklaşımı ve (b) Tekli proje yaklaşımı

Şekil 3.1 Çoklu Proje Çizelgeleme Yaklaşımı

Başlangıç ve bitiş kukla faaliyetleri çizelgeleme probleminin çözümü için gerekli olmasa da, serim problem yapısını anlatmak için eklenmişlerdir.

Çoklu Proje yaklaşımı kullanıldığında, zaman, amaç fonksiyonunda ortalama proje gecikmesiyle (*mean project delay-MPD*) en küçüklenir. $f(\text{proje gecikmesi})$ aşağıdaki gibi hesaplanır.

$$\text{Ortalama Proje Gecikmesi} = \sum_{i=1}^M \sum_t \frac{tS_{i,j_i+1,t} - CP_i}{M} \quad (3.17)$$

Zaman t anında proje i 'de kaynak kısıtlı tamamlanma zamanını verir, $S_{i,j_i+1,t}$ 'de 1'e eşittir. CP_i , proje i 'de kaynak kısıtsız kritik yol uzunluğudur. Açıkça görülmektedir ki; bu kriteri en küçüklemek kaynak kısıtlı projelerde tamamlanma zamanını en küçüklemeye eşdeğerdir.

Tekli proje yaklaşımı kullanıldığında, zaman, amaç fonksiyonunda çoklu proje süre artışını (*multi-project duration increase-Multi-Dur-Inc*) en küçükler. $f(\text{çoklu proje gecikmesi})$ aşağıdaki gibi hesaplanır.

$$\text{Çoklu Süre Artışı(\%)} = \sum_{i=1}^M \sum_t \frac{tS_{i,J_i+1,t} - CP}{CP} * 100 \quad (3.18)$$

Burada M, 1'e eşittir çünkü projeler tekli proje gibi sınırlıdır. J_i çoklu projede her bir faaliyetin numarasıdır. Zaman t anında tekli projede kaynak kısıtlı tamamlanma zamanını verir, S_{i,J_i+1,t} 'de 1'e eşittir. CP, tekli projede kritik yol uzunluğudur. Bu kriteri en küçükleme tamamlanma zamanını en küçüklemeğe eşdeğerdir.

3.7.3.2 Öncelik kuralı üzerine sezgisel taban

Bu yaklaşım çizelgeleme planı üretmek ve öncelik kuralları olarak tanımlanan iki bileşenden oluşturulmuştur.

İki farklı plan genellikle seri plan ve paralel plan olarak ayırt edilmektedir. Kısmi bir plan çizelgelenmiş olan faaliyetlerin sadece bir alt kümesi olarak çizelgelenmesidir. Faaliyetler, her adımda belirli bir öncelik kuralına göre sıralanmış kaynakları kullanmak için yarışır.

Çizelgeleme Planı Üretmek: Seri çizelgeleme planları üretmek Kelley (1963) tarafından önerilmiştir. Bir aktivite her aşamada uygun tamamlanma zamanı ve en erken öncelik kuralına göre seçilmiş ve çizelgelenmiştir. Her bir aşamada birleşmiş iki ayrık faaliyet kümesi vardır. Faaliyetlerin kümesi zaten çizelgelenmiş ve planlar için uygun durumdadır. Çizelgelenmemiş faaliyetlerin hepsi önceliklerine göre çizelgelenmek için hazır durumdadırlar.

Her aşamadaki bir faaliyet, çizelgelenmesi mümkün olur olmaz bir öncelik kuralına göre uygun faaliyetlerin kümesinden seçilmiştir. Seçilen faaliyetler planlanan kümeye eklenmiş ve uygun kümeden kaldırılmıştır. Ek olarak, uygun küme faaliyetleri güncellenmiştir. Son olarak seri çizelgeleme planları bütün faaliyetlerin planlanma kümesine girmesi ile bitmektedir.

Çizelgeleme planlarına bağlı olan öncelik kuralları üzerine sezgisel tabanlı yaklaşımların performans analizinde kullanılan birkaç çalışma vardır. Bu çalışmalar tekli proje çizelgeleme probleminde kaynak kısıtlarını analiz ederler. Sayısal olarak

1000 ve 1500 faaliyetten oluşan projeler göstermiştir ki, paralel planlar ile daha iyi sonuçlar elde edilmektedir. Yine de kesin karakteristik özellikler ile seri planlar, proje içerisinde paralel planları saf dışı bırakabilirler. Aynı zamanda, Kolish ve Hartmann (1998) seri planların daha geniş ölçekli projeler içerisinde daha üstün olduklarına dikkat çekmektedirler. Son zamanlarda Schirmer'da (2000) kaynak kısıtlı tekli proje çizelgeleme problemini çözmek için sezgisel bir yöntem geliştirilmiştir. Öncelik kuralına bağlı olan sezgisel örneklerin performansının analizinden sonra, çizelgeleme planı üretmek, göz önünde bulundurulmuş sezgisel yöntemin parametreleri ve proje karakteristikleri ile yazar uygun bir sezgisel araştırma geliştirmiştir.

Projelerin faaliyetleri çizelgeleme süresinde faaliyet kümeleri içerisinde birleştirilmişlerdir. Bu açıdan, çizelgeleme planı üretmenin ve öncelik kurallarının birleşimi, kaynak kısıtlı çoklu proje çizelgeleme için öncelik kuralı üzerine kurulmuş sezgisel bir yapı inşa eder.

Öncelik Kuralları: Öncelik kuralı çizelgeleme süresince uygun faaliyetlerin kümesini sıralar. Farklı çizelgeler elde etmede kullanılan yaklaşımlara bağlı olan bazı öncelik kuralları ile sıra elde edilir. Anlatılmak istenen şudur ki, eğer öncelik kuralı toplam boşluğu (*minimum total slack-MINSLK*) en küçükleme ise, tekli proje yaklaşımının kullanımı ile bir kritik yol olacaktır. Yine de, eğer çoklu proje yaklaşımı kullanılır ise projeler için en azından birkaç kritik yol olacaktır. Ortalama proje gecikmesini en küçüklemeye çoklu proje yaklaşımı üzerine kurulmuş olan öncelik kuralları ile daha iyi performans sonuçları elde edilir (Kurtuluş and Davis, 1982). Tekli proje yaklaşımı üzerine kurulmuş öncelik kuralları ile tekli projelerin bitiş zamanını en küçüklemeye daha etkili sonuçlar elde edilir (Cooper,1976; Davis, 1975).

Öncelik kuralları 3 kategoride sınıflandırılmıştır. Her bir öncelik kuralında; öncelikle kuralı g'nin her bir faaliyet için nasıl hesaplanacağı belirtilmiştir:

(1) Çoklu Proje Yaklaşımı İle:

En büyük Toplam Çalışma İçeriği (*MAXTWK-Maximum Total Work Content*) E_{nb} g_{ij} ile tanımlanır. g_{ij} aşağıda verilen (3.19) ile hesaplanır.

$$g_{ij} = \sum_{k=1}^K \sum_{ij \in AS_i} d_{ij} r_{ijk} + d_{ij} \sum_{k=1}^K r_{ijk} \quad (3.19)$$

İlk ifade; proje i 'de çoktan çizelgelenen (AS_i) faaliyetler için toplam çalışma programıdır, ikinci ifade proje i 'de faaliyet j ile tükenen toplam çalışmayı ifade eder (Kurtuluş and Narula, 1985).

En Kısa Faaliyetten En Kısa Projeye (SASP-*Shortest Activity From Shortest Project*) Enk g_{ij} ile tanımlanır. g_{ij} , proje i 'nin başlangıç süresine eşittir, ayrıca proje i 'de faaliyet j 'nin süresidir.

En Küçük Toplam Boşluk (MINSLK-MP-*Minimum Total Slack*) Enk g_{ij} ile tanımlanır ve $g_{ij} = LST_{ij} - Max(EST_{ij}, Zaman)$.

En Küçük En Geç Bitirme Zamanı (MINLFT-MP- *Minimum Latest Finish Time*) Enk g_{ij} ile tanımlanır. g_{ij} , proje i 'de faaliyet j 'nin en geç bitme zamanıdır, proje grafiğinde kritik yola göre elde edilir.

Öncelik kurallarından MAXTWK ve SASP önceki çalışmalarda ortalama proje geçikmesini minimize ederken en iyi kurallardı (Kurtuluş and Davis, 1982)

(2) Tekli Proje Yaklaşımı:

En Küçük En Geç Bitirme Zamanı (MINLFT-SP - *Minimum Latest Finish Time*) Enk g_j ile tanımlanır. g_j , tekli projede faaliyet j 'nin en geç bitme zamanıdır, proje grafiğinde kritik yola göre elde edilir.

En Küçül Toplam Boşluk (MINSLK-MP-*Minimum Total Slack*) Enk g_j ile tanımlanır ve $g_j = LST_j - Enb(EST_j, Zaman)$.

Bu öncelik kuralları tekli projede tamamlanma zamanını en küçükleme için düşünülmüştür (Kolisch ve Hartmann, 1998)

(3) Bağımsız Yaklaşım Kullanımı:

İlk Gelen İlk Çıkar (FCFS-*First Come First Served*) öncelik kuralı, uzun süredir kaynak bekleyen faaliyetlerdir. Eğer olaylar bağılı ise faaliyetler sözlük sırasına göre seçilir.

Öncelik kurallarını umursamadan, proje tamamlanma zamanı, projedeki tüm faaliyetlerden en geç proje çizelgelenmiş bitiş zamanından hesaplanır. Çoklu proje tamamlanma zamanı, çoklu projedeki tüm faaliyetlerden en geç çizelgelenmiş bitme zamanından değerlendirilir.

4. KISITLI KAYNAK VE DEVRESEL DEĞİŞKENLİ PROJE ÇİZELGELEME İÇİN BİR ALGORİTMA ÖNERİSİ

Proje çizelgeleme problemlerinde genellikle proje tamamlanma zamanının en erken olması amaçlanmaktadır. Kısıtlı kaynakların olması durumunda ise kısıtlı kaynağın proje süresince kısıtı aşmayacak şekilde düzgün kullanımı amaçlanır. Kaynak maliyetini dikkate alan yöntemler, kaynakların tüm proje süresi boyunca eşit maliyetli olduğunu ve kaynak kapasitesinin sabit olduğunu varsaymaktadır.

Gerçek hayat problemlerinde ise kaynak maliyetleri devresel olarak değişiklik gösterebilir. Proje yöneticileri kaynak maliyetlerinin devresel olarak değiştiği durumlarda maliyet artışlarından kaçınmak isteyeceklerdir. Proje yöneticilerinin devresel kaynak maliyetlerini dikkate alarak kaynak kapasitelerini değiştirmeleri ve/veya faaliyetlerin başlangıç zamanlarını değiştirmeleri devresel kaynak maliyetini dikkate almadan hazırlanan proje çizelgesini geçersiz kılacaktır. Kaynak maliyetlerinin devresel olarak değiştiği ortamlarda bu maliyet değişikliklerini göz önünde bulundurmamak daha gerçekçi ve uygun bir çizelge elde edilmesini sağlayacaktır.

Bu çalışma kapsamında devresel kaynak maliyetlerini ve kısıtlarını dikkate alarak çizelgeleyen bir algoritma geliştirilmiştir. Geliştirilen algoritmanın kullanımı için MS Windows tabanlı bir yazılım C# programlama dili ile VS.NET ortamında hazırlanmıştır.

Algoritmanın geçerliliğinin ve tutarlılığının test edilmesi için test verilerini çözmek üzere bir başka yazılım yine C# programlama dili ile VS.NET ortamında hazırlanmış ve test problemleri bu yazılım üzerinde denenmiştir.

4.1 Devresel Değişiklikler Altında Proje Çizelgeleme

Kaynak kısıtlı proje çizelgeleme yapılırken, kaynak maliyeti ve kapasitelerinin proje süresi boyunca sabit olduğu varsayılarak çizelgeleme yapılır. Gerçek hayatta projeler gerçekleşirken, kaynak kapasite ve maliyetlerinin değişken olduğu gözlemlenebilir. Kaynak maliyetleri piyasa koşullarına bağlı olarak tahmin edilebilir. Özellikle hammadde ve malzeme kaynaklarının fiyatları devresel olarak değişiklikler gösterebilir.

Mevsimsel etkiler sebebiyle, hava şartlarından etkilenen faaliyetlerin gerçekleştirilmesinde ek kaynaklara gereksinim duyulabilir. Verimlilik düşebilir, aydınlatma, ısıtma gibi dolaylı maliyetler ile maliyet artışları gözlenebilir. Örneğin yapı projelerinde beton üretme işleminde kış aylarında malzeme ve işçilik maliyetleri artmakta, buna karşın ekipman kiralama maliyetleri de düşebilmektedir.

Yerel yönetim, yasalar ve yerel koşullar, projenin ve faaliyetin fiziksel koşulları sebebiyle kaynak kullanım kapasiteleri devresel olarak farklı kısıtlara sahip olabilirler.

Birden çok projenin yönetildiği durumlarda ise proje için ayrılan kaynak kapasitesi devresel olarak değiştirilebilir.

Devresel maliyet değişikliklerinin olduğu durumlarda maliyetin düşük olduğu devrede faaliyeti gerçekleştirmek maliyeti düşürebilir. Dolayısıyla maliyetin düşürülmesi istenen bir durum olduğundan proje yönetimi tarafından faaliyetler maliyetin düşük olduğu devrelerde yapılmaya çalışılacaktır. Kaynak kapasitelerinin devresel olarak değişmesi ve bu kapasitelere uyulması durumunda iş programı güncellenmek zorunda kalacaktır. Bu sebeplerden dolayı devresel değişikliklerin dikkate alınmadan yapılan iş programlarında gecikmeler ve tutarsızlıklar olacaktır.

Devresel değişikliklerin dikkate alınarak kaynak kısıtlı proje çizelgelemesi ile elde edilecek bir iş programı hem mevcut durumu daha iyi yansıtan geçerli bir iş programı olacak hem de maliyetin yüksek olduğu durumlardan kaçınarak maliyeti düşürecektir.

Devresel maliyetler dikkate alınarak faaliyetlerin çizelgelenmesi, istenilen proje tamamlanma süresi içerisinde kalması sağlanabilir, ya da proje tamamlanma süresinin uzamasına izin verilebilir. Proje süresinin uzaması durumunda gecikmeden dolayı oluşacak maliyet dikkate alınarak çizelgeleme yapılmalıdır. Gecikmeden dolayı oluşacak maliyet, projenin sabit maliyetleri ve gecikmeden dolayı oluşacak ceza maliyetidir.

4.2 Önerilen Algoritmanın Tanıtımı

Önerilen algoritma iki aşamadan oluşmaktadır. Birinci aşamada kaynak kapasitelerine uygun bir çizelge elde edilir. İkinci aşamada ise elde edilen çizelge kaynak maliyetlerine göre tekrar çizelgelenir.

Veri girişi olarak faaliyet bilgileri ve öncelik ilişkilerinin yanı sıra devreler ve devrelere ait kaynak maliyet ve kapasite verileri de girilir.

Birinci aşama başlangıç çizelge oluşturmak için kullanılacaktır. Hızlı sonuç veren bir yöntem oluşturulmak istenmiştir ve öncelik kuralına dayalı tek çözüm üreten bir yöntem geliştirilmiştir.

Birinci aşamada tüm faaliyetler atanmamış faaliyetler kümesine atanır. Atanmamış faaliyetlerin hepsi için atanabilir olup olmadığı kontrol edilir. Öncül faaliyetlerin tümü atanmış ve tamamlanmış olan ve atanmak istenen zamanda faaliyet kaynak gereksiniminin kaynak kapasitesini aşmıyor olması faaliyetin atanabilir olduğunu gösterir. Atanabilir faaliyetler kümesinden öncelik kuralına uygun olan faaliyet atanarak atanmış faaliyetler kümesine dahil edilir. Eğer atanabilir bir faaliyet yoksa bir sonraki zaman dilimine geçilir ve atanabilir faaliyet kümesi güncellenir. Atanmamış faaliyet kümesi boşalincaya kadar bu işlemlere devam edilir.

Birinci aşamada yer alan atanabilir faaliyetler arasından faaliyet seçilmesi önemli bir noktadır. Atanacak faaliyetin hangi öncelik kuralına göre atanacağı belirlenmelidir. Burada birçok öncelik kuralı tanımlanabilir. Algoritmada öncelik kuralının kullanıcı tarafından seçileceği düşünülmüştür. Sunulan öncelik kuralları şunlardır;

- En Kısa Bolluk Süresi
- En Yüksek Toplam Kaynak Gereksinimi
- En Kısa Faaliyet Süresi
- Ardılların Toplam Bolluğu En Kısa
- En Fazla Toplam Ardıl Sayısı
- Birincil Ardılların Toplam Bolluğu En Kısa
- En Düşük Toplam Kaynak Gereksinimi

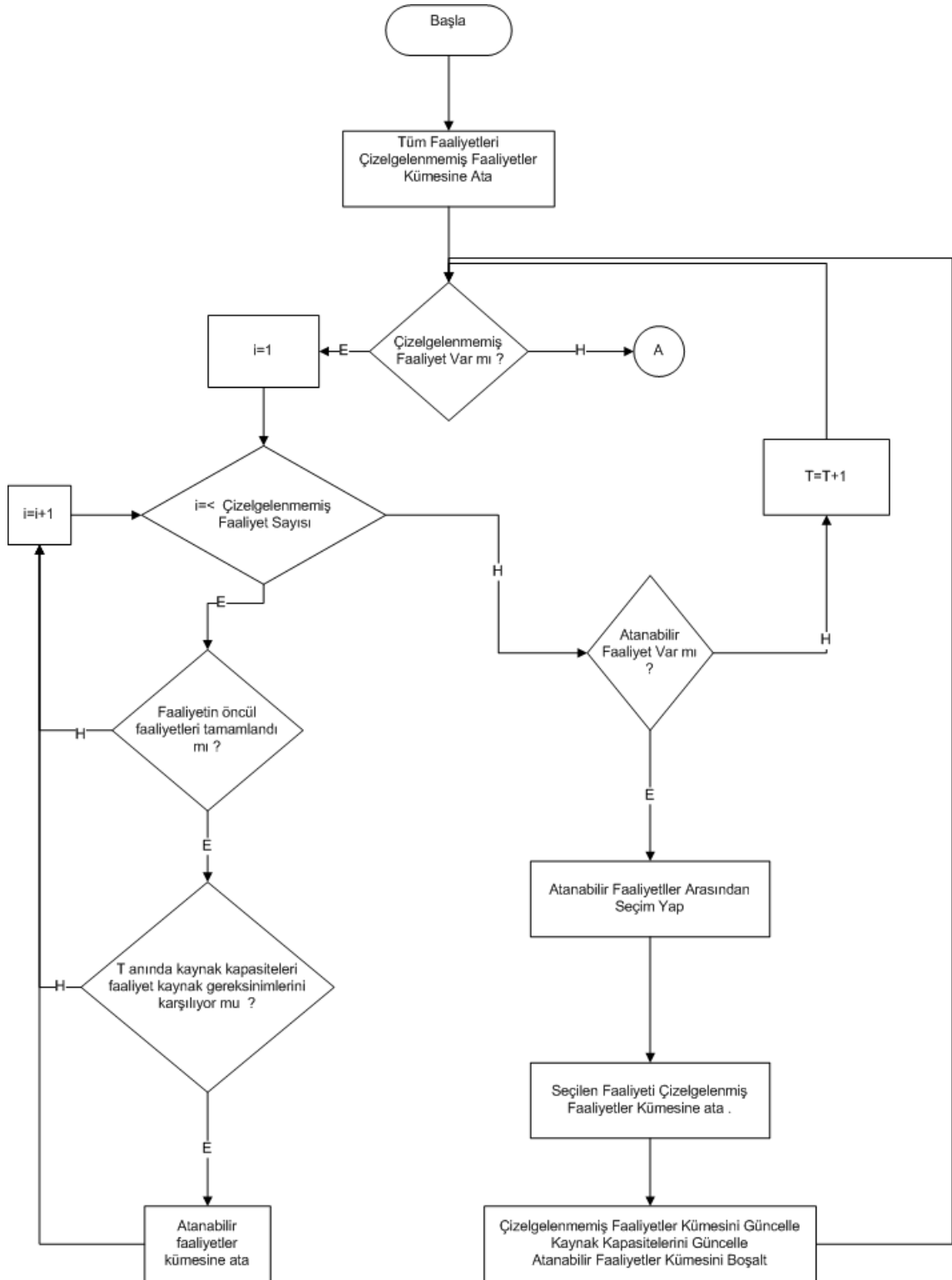
Yukarıda sıralanan ölçütlerden biri seçilip kullanılabilmesi gibi birden fazlası da ağırlıklandırılarak kullanılabilir, birden fazla öncelik kuralı ile çözümler üretilebilir ve sonuçlar arasından en iyi olanı seçilebilir.

İkinci aşamada tüm faaliyetler yeniden çizelgeleme için kontrol edilir. Eğer faaliyetin atandığı devreden bir önceki devrenin maliyeti daha düşük ise, faaliyet başlangıç zamanı ve kaynak kapasiteleri uygun ise faaliyet başlangıç zamanı öne çekilir. Atandığı devreden bir sonraki devrenin maliyeti daha düşük ise ve faaliyetin bolluğu varsa bir sonraki devreye doğru kaydırılır. Ayrıca bir sonraki devrenin maliyeti daha düşük ve kaynak kapasiteleri de uygun ancak faaliyetin bolluğu yok ise faaliyetin gecikmesinden etkilenen diğer faaliyetler de ötelenerek oluşacak gecikme durumunda gecikme maliyetine bakılır. Eğer toplam maliyet azalıyor ise faaliyet bir devre sonrasına ötelenir. Tekrar çizelgeleme işlemi yapılan faaliyet kalmayıncaya kadar algoritma çalışır.

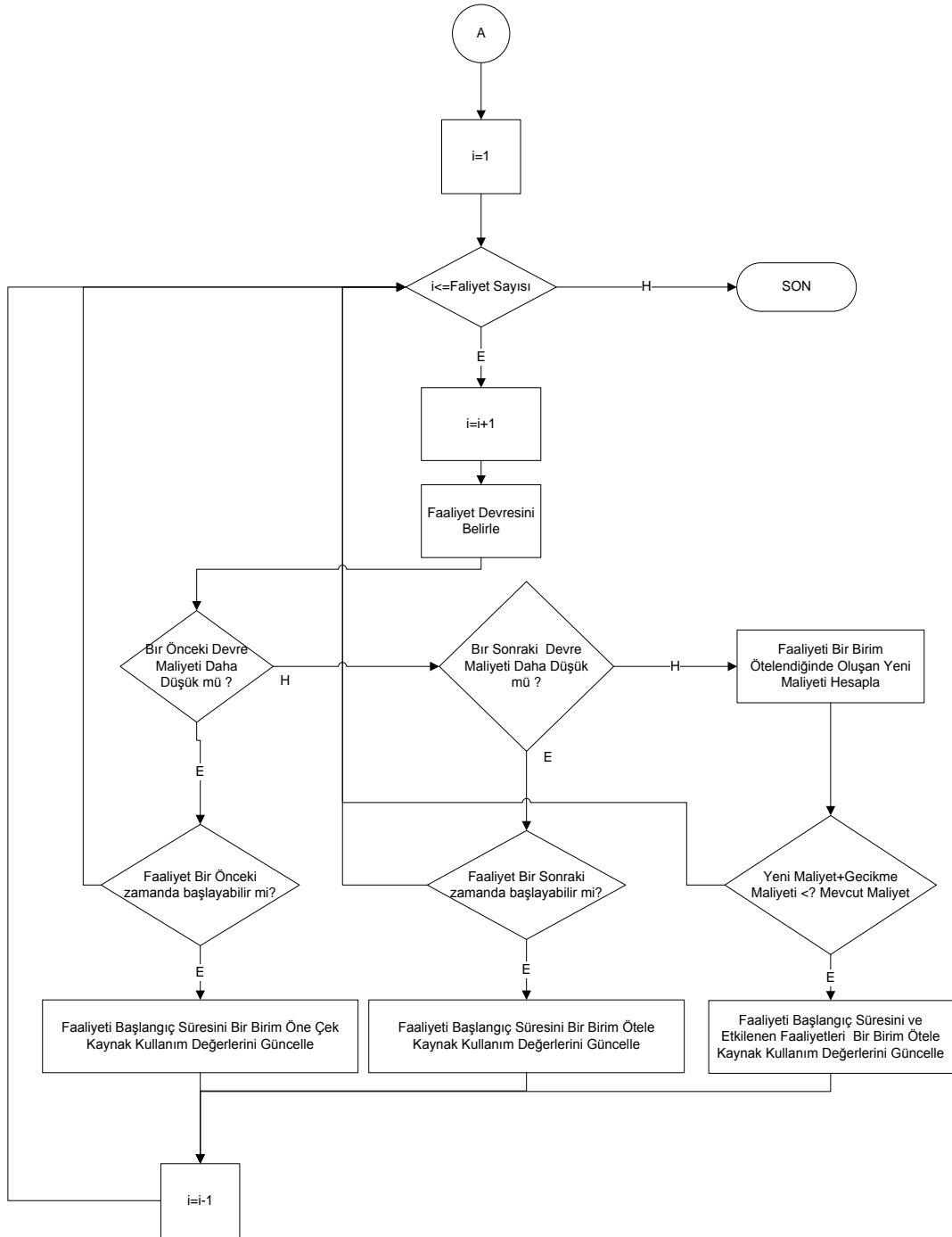
Önerilen algoritmanın akış diyagramı Şekil 4.1’de verilmiştir. Tek kaynak kısıtını dikkate alarak çözüm üretebileceği gibi çok kaynak kısıtı da dikkate alınarak çözüm üretilebilir. Çok kaynak kısıtı kullanılması durumunda, kaynak kapasite ve kullanım uygunluğunun bütün kaynaklar için dikkate alınması gerekecektir.

Çok projeli çizelgeler elde edilmesi için faaliyet süresi ve kaynak maliyetleri sıfır olan yapay başlangıç ve bitiş faaliyetleri oluşturulur. Tüm projeler bu başlangıç ve bitiş faaliyetleri arasına eklenerek projelerin birleştirilmesi ile tek bir proje elde edilir ve bu projenin çizelgelenmesi ile çoklu projeler de çizelgelenmiş olur.

Algoritmanın C# dili ile hazırlanmış kaynak kodu EK 1’ de verilmiştir.



Şekil 4.1 Algoritma Akış Diyagramı



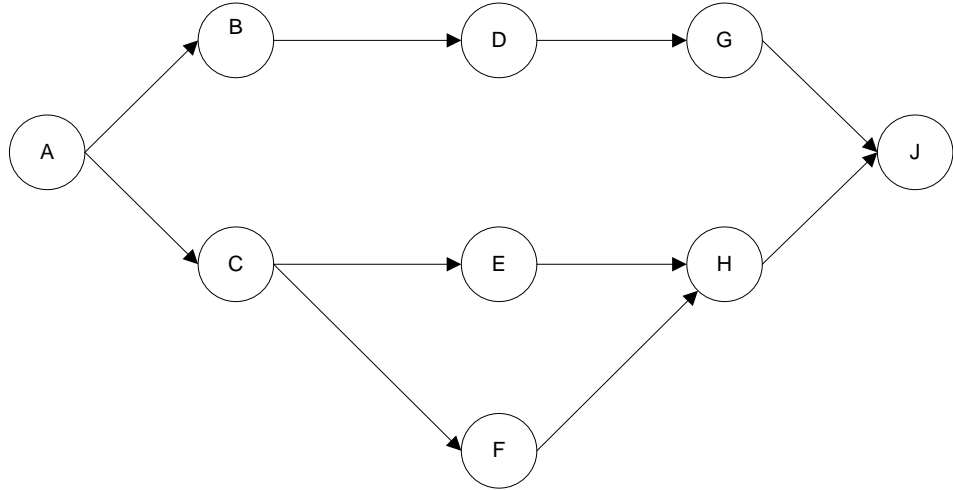
Şekil 4.1 Algoritma Akış Diyagramı (Devam)

4.3 Algoritmanın Örnek Problemler Üzerinde Çalıştırılması

Geliştirilen algoritmanın anlatılması ve çözümlerin değerlendirilmesi amacı ile örnek problem oluşturularak çözümleri gösterilmiştir. Öncelikle bir örnek proje ele alınarak tek kaynak kısıtı ile kaynak maliyetinin sabit olduğu ve kaynak maliyetinin devresel değişkenlik gösterdiği durumlarda çözümler üretilmiştir. Aynı proje çok kaynak kısıtı eklenerek sabit maliyet ve devresel maliyetler dikkate alınarak çizelgelenmiştir. Başka bir proje daha eklenerek çoklu proje elde edilmiş ve elde edilen çözüm gösterilmiştir.

4.3.1 Tek kaynak kısıtının olması durumunda çözüm

Çözümü araştırılan örnek proje dokuz faaliyetten oluşmaktadır ve faaliyetler arasındaki ilişkilerin hepsinin Bitiş-Başlangıç (FS) olduğu varsayılmıştır. Probleme ait serim diyagramı Şekil 4.2’de görülmektedir.



Şekil 4.2 Örnek Problem Serim Diyagramı

Faaliyetlerin süreleri ve kaynak gereksinimleri Çizelge 4.1’de verilmiştir. Kaynak kısıtı 11 adet olarak alınmıştır. CPM ile elde edilen çözüm Çizelge 4.2’de verilmiştir. Tek kaynak ve sabit kaynak maliyeti ile yapılan çözüm Çizelge 4.3’te verilmiştir.

Çizelge 4.1 Örnek Proje Faaliyet Süreleri ve Kaynak Gereksinimleri

Faaliyet	Süre (gün)	Kaynak Gereksinimi (adet)
A	3	8
B	5	6
C	4	7
D	5	7
E	3	4
F	2	1
G	5	7
H	3	2
J	1	6

Çizelge 4.2 Örnek Proje İçin CPM ile Elde Edilen Çözüm

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
A	A	A																	
			B	B	B	B	B												
			C	C	C	C													
								D	D	D	D	D							
							E	E	E										
							F	F											
													G	G	G	G	G		
										H	H	H							
																		J	
KK	8	8	8	13	13	13	13	11	12	11	9	9	9	7	7	7	7	7	6

Çizelge 4.3 Örnek Proje Tek Kaynak ve Sabit Maliyet ile Çözüm

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	A	A	A																				
								B	B	B	B	B											
				C	C	C	C																
													D	D	D	D	D						
								E	E	E													
								F	F														
																			G	G	G	G	G
											H	H	H										
																							J
KK	8	8	8	7	7	7	7	11	11	10	8	8	9	7	7	7	7	7	7	7	7	7	6

CPM çözümünde proje süresi 19 gün olarak hesaplanmış ancak kaynak kısıtı dikkate alınmadığından geçerli bir proje çizelgesi elde edilememiştir. Tek kaynak kısıtı ve sabit maliyet ile elde edilen çözümde proje süresi 23 gün olmuş ve kaynak kısıtı sağlanmıştır. Birim kaynak maliyeti 10 TL olarak alındığında her iki çözüm içinde toplam kaynak kullanım maliyeti 1780 TL olarak hesaplanır.

Devresel kaynak maliyet ve kapasiteleri Çizelge 4.4'te verilmiştir. Devreler birbirini takip eder şekilde algoritmaya aktarılır. Örneğin 4. devreden sonra tekrar 1. devre gelir ve 16-20 günleri arası 1. devrenin maliyet ve kapasite değerlerini alır.

Çizelge 4.4 Devresel Kaynak Maliyet ve Kapasiteleri

Devre	Zamanı Aralığı	Maliyet(TL)	Kapasite (adet)
1	0-4	20	10
2	4-8	14	14
3	8-12	25	8
4	12-16	16	12

Verilen devreler ve gecikme maliyeti 2 TL alınarak Çizelge 4.5'te verilen, gecikme maliyeti 5 TL olarak alınmış ve Çizelge 4.6'da verilen çözümler elde edilmiştir. Kaynak kullanımı (KK), birim kaynak maliyeti (BKM), kaynak kullanım maliyeti (KKM) bilgileri çizelgelerde verilmiştir. Gecikme maliyetinin 2 TL olduğu çözümde J faaliyetine dikkat edilecek olursa bir sonraki ve maliyeti daha düşük olan devreye kaydırıldığı görülür. Gecikme maliyeti 50 TL olan çözümde ise böyle bir ötelemeden doğacak maliyet düşüşü gecikme maliyetinin altında olacağından böyle bir öteleme yapılmadığı görülür.

Çizelge 4.5 Örnek Proje Tek Kaynak ve Devresel Maliyet ile Çözüm

(Gecikme Maliyeti = 2 TL)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
	A	A	A																		
				B	B	B	B	B													
					C	C	C	C													
									D	D	D	D	D								
													E	E	E						
													F	F							
														G	G	G	G	G			
																H	H	H			
																					J
KK	8	8	8	6	13	13	13	13	7	7	7	7	12	12	11	9	9	9	0	0	6
BKM	20	20	20	20	14	14	14	14	25	25	25	25	16	16	16	16	20	20	20	20	14
KM	160	160	160	120	182	182	182	182	175	175	175	175	192	192	176	144	180	180	0	0	84

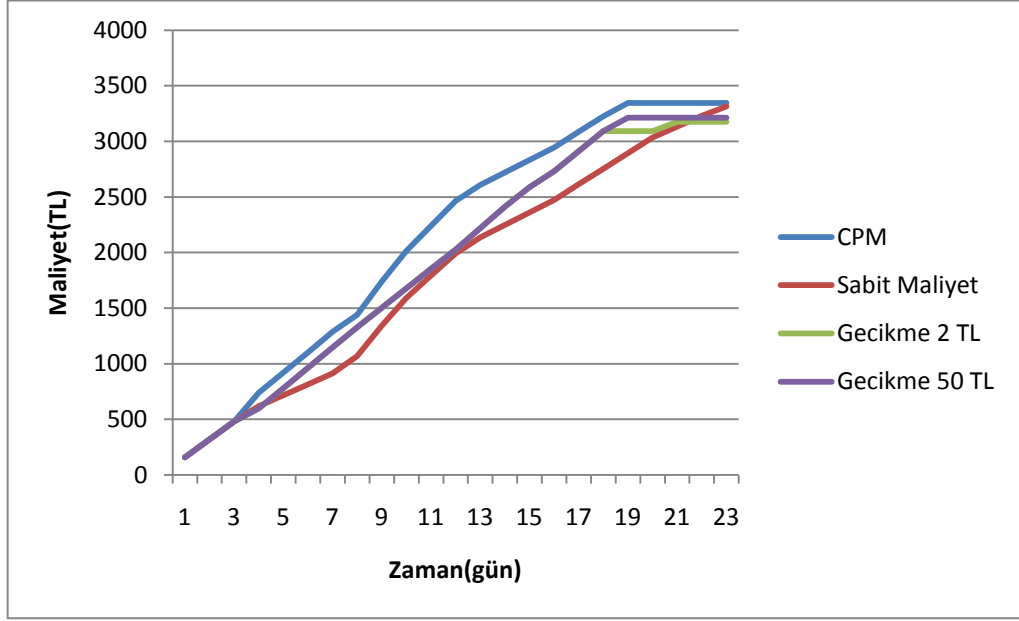
Çizelge 4.6 Örnek Proje Tek Kaynak ve Devresel Maliyet ile Çözüm
(Gecikme Maliyeti 50 TL)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
	A	A	A																
				B	B	B	B	B											
					C	C	C	C											
									D	D	D	D	D						
													E	E	E				
													F	F					
														G	G	G	G	G	
																H	H	H	
																			J
KK	8	8	8	6	13	13	13	13	7	7	7	7	12	12	11	9	9	9	6
BKM	20	20	20	20	14	14	14	14	25	25	25	25	16	16	16	16	20	20	20
KKM	160	160	160	120	182	182	182	182	175	175	175	175	192	192	176	144	180	180	120

Gecikme maliyetinin 2 TL olduğu durumda toplam kaynak kullanım maliyeti 3176 TL ve gecikme maliyeti 4 TL olacaktır. Gecikme maliyetinin 50 TL olduğu durumda ise toplam kaynak kullanım maliyeti 3212 TL ve gecikme maliyeti 0 TL olacaktır. CPM ile elde edilen çözümün verilen devresel maliyetler dikkate alındığında toplam kaynak kullanım maliyeti 3345 TL'dir. Sabit maliyet ile yapılan çözümün maliyeti devresel maliyetler ile hesaplandığında 3313 TL olarak bulunur. Bu çözümlerin birikimli toplam kaynak maliyet dağılımları Şekil 4.3'te verilmiştir.

4.3.2 Kısıtlı çok kaynağın olması durumunda çözüm

Projeyi çok kaynak kısıtlı hale getirmek için 3 kaynak daha tanımlanmıştır. Faaliyetlerin kaynak gereksinimleri Çizelge 4.7'de verilmiştir. Kaynak kapasite ve maliyetleri her devre için tanımlanmış ve Çizelge 4.8'de verilmiştir.



Şekil 4.3 Birikimli Toplam Maliyetler

Çizelge 4.7 Örnek Proje Kaynak Gereksinimleri

Faaliyet	Kaynak 1 (adet)	Kaynak 2 (adet)	Kaynak 3 (adet)	Kaynak 4 (adet)
A	8	5	7	2
B	6	0	9	0
C	7	4	5	4
D	7	7	6	3
E	4	6	4	1
F	1	2	5	5
G	7	6	2	1
H	2	3	4	0
J	6	2	7	2

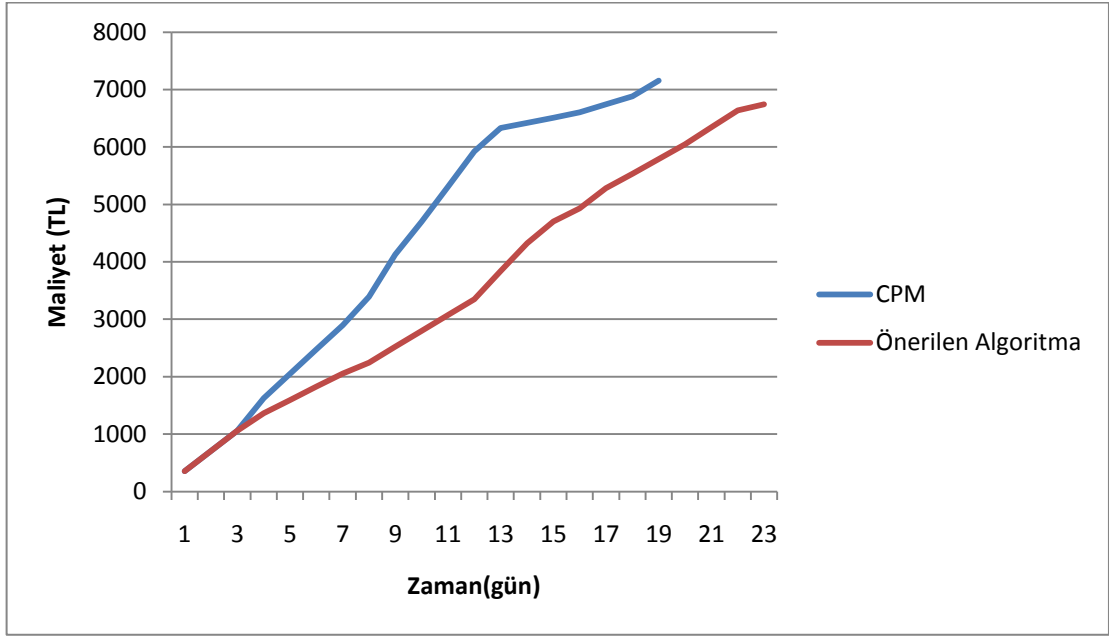
Çizelge 4.8 Örnek Proje Devresel Kaynak Maliyet ve Kapasiteleri

Devre	Zaman Aralığı	Kaynak 1		Kaynak 2		Kaynak 3		Kaynak 4	
		Maliyet (TL)	Kapasite (adet)	Maliyet (TL)	Kapasite (adet)	Maliyet (TL)	Kapasite (adet)	Maliyet (TL)	Kapasite (adet)
1	0-4	20	10	12	8	16	10	10	6
2	4-8	14	14	10	9	12	14	8	8
3	8-12	25	8	8	12	14	12	12	5
4	12-16	16	12	6	16	10	16	6	10

Örnek proje gecikme maliyeti 50 TL alınarak çok kaynak kısıtı altında çözülmüştür ve Çizelge 4.9’da yer alan çizelge elde edilmiştir. Toplam kaynak maliyeti 6742 TL olmuştur. Aynı kaynak gereksinimi ve maliyetleri ile CPM ile elde edilen toplam maliyet 6935 TL olup birikimli toplam kaynak maliyeti grafiği Şekil 4.4’te verilmiştir.

Çizelge 4.9 Örnek Projenin Çok Kaynak Kısıtı Altında Çözümü

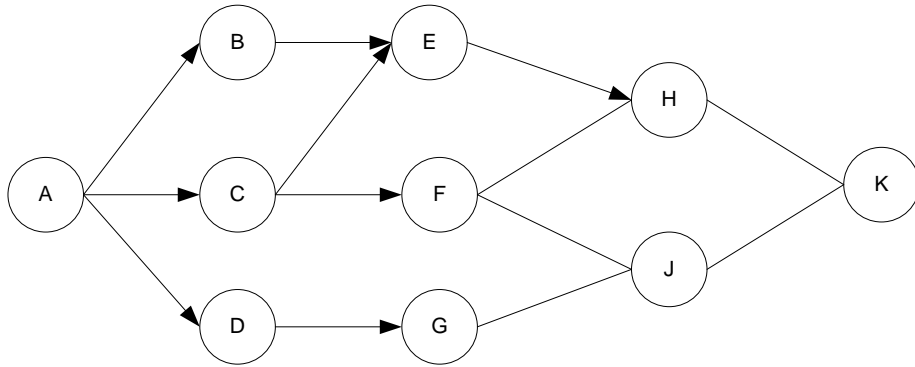
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	A	A	A																				
					B	B	B	B	B														
				C	C	C	C																
										D	D	D	D	D									
													E	E	E								
													F	F									
															G	G	G	G	G				
																				H	H	H	
																							J
KK1	8	8	8	7	13	13	13	6	6	7	7	7	12	12	11	7	7	7	7	2	2	2	6
KK2	5	5	5	4	4	4	4	0	0	7	7	7	15	15	12	6	6	6	6	3	3	3	2
KK3	7	7	7	5	14	14	14	9	9	6	6	6	15	15	6	2	2	2	2	4	4	4	7
KK4	2	2	2	4	4	4	4	0	0	3	3	3	9	9	2	1	1	1	1	0	0	0	2



Şekil 4.4 Toplam Birikimli Kaynak Maliyetleri

4.3.3 Çoklu projelerin çözümü

Çoklu proje üzerinde algoritmanın çalıştırılması için bir proje daha tanımlanmıştır. İkinci projenin serim diyagramı Şekil 4.5’de verilmiştir. Faaliyetlerin süreleri ve kaynak gereksinimleri Çizelge 4.10’da verilmiştir.



Şekil 4.5 Proje 2 Serim Diyagramı

Çizelge 4.10 Proje 2 Faaliyet Süreleri ve Kaynak Gereksinimleri

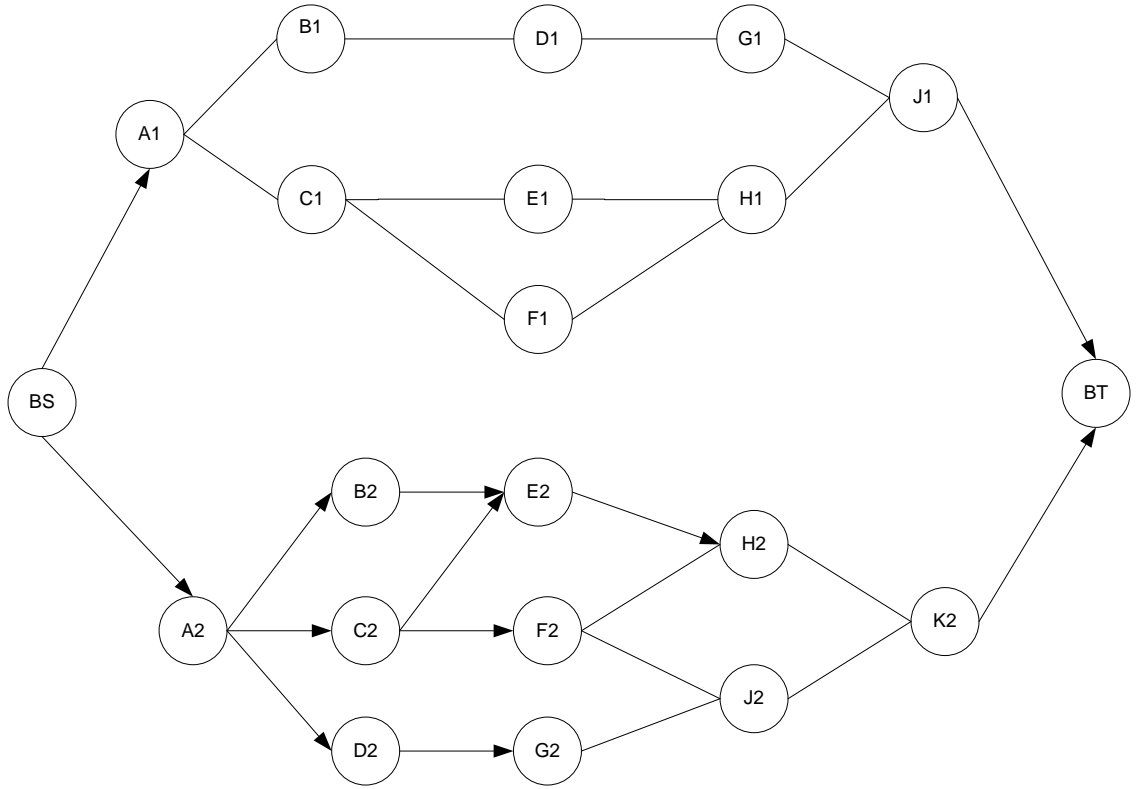
Faaliyet	Süre	Kaynak 1	Kaynak 2	Kaynak 3	Kaynak 4
A	2	5	2	4	2
B	5	4	6	4	4
C	4	2	4	1	1
D	3	4	5	3	3
E	3	3	3	0	3
F	2	5	4	3	4
G	2	7	3	4	2
H	4	6	5	3	2
J	3	3	2	2	1
K	1	4	1	0	3

Projenin gecikme maliyeti 50 TL ve verilen devresel maliyetler alınarak yapılan çözümü Çizelge 4.11’de verilmiştir. Toplam kaynak kullanım maliyeti 4,801 TL olmaktadır.

Çizelge 4.11 Örnek Proje 2 Çözümü

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
A	A																					
										B	B	B	B	B								
		C	C	C	C																	
				D	D	D																
															E	E	E					
							F	F														
								G	G													
																		H	H	H	H	
													J	J	J							
																						K
KK1	5	5	2	2	6	6	9	12	7	4	4	4	4	7	6	6	3	6	6	6	6	4
KK2	2	2	4	4	9	9	9	7	3	6	6	6	6	8	5	5	3	5	5	5	5	1
KK3	4	4	1	1	4	4	6	7	4	4	4	4	4	6	2	2	0	3	3	3	3	0
KK4	2	2	1	1	4	4	7	6	2	4	4	4	4	5	4	4	3	2	2	2	2	3

Çoklu projelerin çizelgelenmesinde, faaliyet süresi ve kaynak gereksinimleri sıfır olan bir başlangıç (BS) ve bir bitiş (BT) faaliyeti eklenerek projeler bu başlangıç ve bitiş faaliyetlerine bağlanarak tek bir proje haline getirilmektedir. Şekil 4.6'da birleştirme sonucunda elde edilen serim diyagramı verilmiştir.



Şekil 4.6 Çoklu Projelerin Birleştirilmesi

Çoklu projelerin CPM ile elde edilen çözümü Çizelge 4.12 de verilmiştir. CPM ile elde edilen çözümün projelerin tek tek çözülerek elde edilen CPM çözümü ile aynı olduğu görülmüştür. Birinci projenin tamamlanma zamanı J1 faaliyetinin tamamlanma zamanıdır ve 19. gün, ikinci projenin tamamlanma zamanı K2 faaliyetinin tamamlanma zamanıdır ve 15. gündür. Projelerin toplam kaynak kullanım maliyeti 12,244 TL olarak hesaplanmıştır.

Çizelge 4.12 Çoklu Projelerin CPM ile Çizelgelenmesi

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	
A1	A1	A1																	
			B1	B1	B1	B1	B1												
			C1	C1	C1	C1													
								D1	D1	D1	D1	D1							
							E1	E1	E1										
							F1	F1											
													G1	G1	G1	G1	G1		
										H1	H1	H1							
																		J1	
A2	A2																		
		B2	B2	B2	B2	B2													
		C2	C2	C2	C2														
		D2	D2	D2															
							E2	E2	E2										
						F2	F2												
					G2	G2													
										H2	H2	H2	H2						
									J2	J2	J2								
															K2				
KK1	13	13	18	23	23	26	29	19	18	17	18	15	15	13	11	7	7	7	6
KK2	7	7	20	19	19	17	17	15	20	18	17	15	15	11	7	6	6	6	2
KK3	11	11	15	22	22	23	25	21	17	12	15	13	13	5	2	2	2	2	7
KK4	4	4	10	12	12	11	14	13	13	8	6	5	5	3	4	1	1	1	2

Projeler birleştirilirken devresel kaynak maliyetleri aynı olarak alınmış, devresel kaynak kapasiteleri proje kaynak kapasiteleri toplanarak kullanılmıştır. Gecikme maliyetinin 100 TL alınarak yapılan çözüm Çizelge 4.13'te verilmiştir. Elde edilen çizelgede birinci proje J1 faaliyetinin tamamlanmasıyla 20. günde, ikinci proje K2 işleminin tamamlanmasıyla 21. günde tamamlanmaktadır. Toplam kaynak kullanım maliyeti 11,898 TL olarak hesaplanmıştır. CPM ile yapılan çözüme göre toplam maliyet 346 TL azalmıştır.

Çizelge 4.13 Çoklu Projelerin Çok Kaynak Kısıtı ile Çözülmesi

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	
A1	A1	A1																			
			B1	B1	B1	B1	B1														
			C1	C1	C1	C1															
									D1	D1	D1	D1	D1								
							E1	E1	E1												
							F1	F1													
														G1	G1	G1	G1	G1			
												H1	H1	H1							
																			J1		
A2	A2																				
		B2	B2	B2	B2	B2															
		C2	C2	C2	C2																
												D2	D2	D2							
							E2	E2	E2												
						F2	F2														
															G2	G2					
												H2	H2	H2	H2						
																	J2	J2	J2		
																				K2	
KK1	13	13	14	19	19	19	22	19	8	14	7	7	19	17	13	22	20	10	10	9	4
KK2	7	7	15	14	14	14	14	15	11	16	7	7	20	17	14	17	14	8	8	4	1
KK3	11	11	12	19	19	19	21	21	9	10	6	6	16	12	9	13	9	4	4	9	0
KK4	4	4	7	9	9	9	12	13	9	7	3	3	8	8	4	5	5	2	2	3	3

4.4 Önerilen Algoritmanın Literatürde Yer Alan Test Problemleri İle Denenmesi

Geliştirilen algoritma, çeşitli problemler üzerinde çalışmasını gözlemek, tutarlılığını ve ürettiği sonuçları incelemek amacıyla literatürde yer alan test problemleri üzerinde denenmiştir.

Kaynak kısıtlı proje çizelgeleme çalışmalarında Kolisch ve Sprecher (1997) tarafından geliştirilmiş “Project Scheduling Library” (PSPLIB) test problemleri kullanılmıştır.

PSPLIB içerisinde tek modlu ve çok modlu veri setleri yer almaktadır. Geliştirilen algoritma yapısı tek modlu olduğundan tek modlu veri setleri kullanılmıştır. Kullanılan veri setlerinin özellikleri Çizelge 4.14’de verilmiştir. Tüm setlerde kaynak sayısı 4 ve yenilenebilir olarak ele alınmıştır. J30 setinin çözümleri olarak en iyi sonuçlar verilmekte, diğer setler için ise geliştirilen yöntemler tarafından bugüne kadar bulunan en iyi çözümler verilmektedir.

Çizelge 4.14 PSPLIB Veri Setleri

VERİ SETLERİ			
AD	FAALİYET SAYISI	PROBLEM SAYISI	KAYNAK SAYISI
J30	30	480	4
J60	60	480	4
J90	90	480	4
J120	120	600	4

Test problemlerinin toplu olarak çözülmesi için ayrı bir yazılım C# dili ile geliştirilmiştir. Bu yazılım ile seçilen veri setindeki her problem dosyası tüm öncelik kurallarında çözdürülerek sonuçlar kaydedilmektedir. Öncelik kuralları Çizelge 4.15’deki gibi adlandırılmış ve ilgili diğer çizelgelerde de bu şekilde kullanılmıştır.

Çizelge 4.15 Öncelik Kuralları

A	En Kısa Bolluk Süresi
B	En Yüksek Toplam Kaynak Gereksinimi
C	En Kısa Faaliyet Süresi
D	Ardılların Toplam Bolluğu En Kısa
E	En Fazla Toplam Ardıl Sayısı
F	Birincil Ardılların Toplam Bolluğu En Kısa
G	En Düşük Toplam Kaynak Gereksinimi

Veri setleri içerisinde tek kaynaklı projelere yer verilmemiştir. Algoritma ile yapılan çok kaynaklı çözümlerin tek kaynak altında da sonuç vereceği düşüncesi ile tek kaynaklı problem ve çözüm üretmeye gerek görülmemiştir.

PSPLIB veri setleri içerisinde çoklu projeler için veri setleri yer almamaktadır, ancak veri setleri içerisinde seçilen projeler ile çoklu projeler türetilerek test verileri elde edilmiştir. Veri setlerinde yer alan projeler, istenilen sayıdaki kadar proje birleştirilerek çoklu projeler elde edilmiştir. Birleştirme işlemi esnasında, faaliyet süresi ve kaynak gereksinimleri sıfır olan bir başlangıç ve bir bitiş faaliyeti tanımlanarak tüm projeler bu başlangıç ve bitiş faaliyetlerine bağlanarak tek bir proje haline getirilmektedir. Kaynak kapasiteleri eklenen her projenin kaynak kapasiteleri toplanarak belirlenmektedir.

Veri setlerinde kaynak maliyetleri belirtilmemiştir. Devresel olarak kaynak maliyetlerinin değiştiği herhangi bir problem setine de rastlanmamıştır. Devresel kaynak maliyetlerinin dikkate alındığı durumlarda algoritmanın etkinliğini gözlemlemek için devresel maliyetler tanımlanmış ve benzer şekilde PSPLIB veri setleri üzerinde denenmiştir.

Tüm çözümler Intel Core 2 Duo E4700 2.6 Ghz işlemci ve 2 GB hafızaya ve Windows XP işletim sistemi kullanılan bir PC üzerinde denenerek elde edilmiştir. Program çalışma süreleri milisaniye olarak kaydedilmiştir.

4.4.1 Projelerin kaynak maliyetlerinin dikkate alınmadan çizelgelenmesi

Veri setlerinde yer alan projeler kaynak maliyetleri dikkate alınmadan çözülmüştür. Çözümler sonucunda önerilen yazılımın literatürdeki en iyi sonuçlara ulaşma oranları Çizelge 4.16'da verilmiştir. Çizelgede algoritmanın öncelik kuralında kaç defa en iyi sonuca ulaştığı sayı olarak ve yüzde olarak verilmiştir. ENK satırında öncelik kurallarının hepsinin sırayla çalıştırıldığı ve en iyi sonucun dikkate alındığı durum verilmiştir. Bu şekilde toplam çalışma süresi tüm öncelik kurallarının toplamı kadar olacak ancak en iyi sonuca ulaşma sıklığı artacaktır. En iyi sonuca ulaşma oranı j30 problemlerinde %47 olarak hesaplanmıştır. A öncelik kuralı tüm veri setlerinde en fazla en iyi sonuca ulaşan yöntem olmuştur. Faaliyet sayısı arttıkça en iyi sonuca ulaşma oranının düştüğü görülmektedir.

Çizelge 4.16 En iyi Sonuca Ulaşma Oranları

	J30		J60		J90		J120	
	SAYI	YÜZDE	SAYI	YÜZDE	SAYI	YÜZDE	SAYI	YÜZDE
A	204	43%	187	39%	201	42%	18	3%
B	148	31%	141	29%	129	27%	3	1%
C	152	32%	141	29%	138	29%	3	1%
D	195	41%	184	38%	184	38%	9	2%
E	192	40%	179	37%	173	36%	14	2%
F	185	39%	167	35%	178	37%	5	1%
G	166	35%	156	33%	154	32%	3	1%
ENK	227	47%	219	46%	216	45%	29	5%

Çizelge 4.17’de önerilen algoritma ile elde edilen sonuçlarla bilinen en iyi sonuçlar arasındaki proje tamamlanma süresi farkının ortalaması, standart sapması ve yüzdesi verilmiştir.

Çizelge 4.17 En İyi Sonuca Yakınlık

	J30			J60			J90			J120		
	ORT.	SS	%	ORT.	SS	%	ORT.	SS	%	ORT.	SS	%
A	4.47	6.50	93	6.96	9.84	92	7.87	11.54	93	22.72	17.59	83
B	7.70	8.72	88	12.21	13.51	86	15.30	16.58	85	37.70	22.66	71
C	6.19	6.78	90	9.59	10.48	89	12.21	13.43	88	29.49	17.65	77
D	5.38	7.55	92	8.47	11.48	91	10.13	14.11	91	29.95	21.40	77
E	3.27	4.07	95	5.35	6.58	94	6.20	8.09	94	15.92	11.03	88
F	5.45	7.26	92	8.80	11.47	90	10.30	14.04	91	27.81	19.08	79
G	4.43	5.11	93	7.14	8.12	92	8.58	10.10	92	21.15	12.46	83
ENK	2.30	3.19	96	4.26	5.83	95	5.14	7.55	95	14.69	10.99	89

Çizelge 4.18’de öncelik kurallarının kendi aralarında kıyaslanması ve her öncelik kuralına ait çalışma süreleri gözlenmektedir. Kıyaslama yapılırken öncelik kuralları ile bulunan en iyi sonuca ulaşma oranı belirlenmiştir. A kuralı ve E kuralının diğer kurallara göre baskın olduğu görülmektedir.

Çizelge 4.18 Öncelik Kurallarının Karşılaştırılması

	J30			J60			J90			J120		
	EN İYİ SONUC	%	CALISMA SÜRESİ (ms)	EN İYİ SONUC	%	CALISMA SÜRESİ (ms)	EN İYİ SONUC	%	CALISMA SÜRESİ (ms)	EN İYİ SONUC	%	CALISMA SÜRESİ (ms)
A	305	64	3.12	284	59	10.50	201	42	20.84	137	23	39.17
B	180	38	3.19	157	33	10.78	129	27	21.55	6	1	41.55
C	202	42	3.15	164	34	10.71	138	29	21.34	21	4	40.63
D	276	58	3.38	249	52	11.55	184	38	23.45	40	7	47.13
E	332	69	3.30	327	68	11.19	173	36	22.64	397	66	43.91
F	252	53	3.12	221	46	10.57	178	37	21.07	41	7	40.01
G	249	52	3.12	216	45	10.53	154	32	20.94	81	14	39.29
ENK	480	100	22.37	480	100	75.82	480	100	151.83	600	100	291.68

Çizelge 4.19’da J30 problemleri için diğer algoritmalar tarafından elde edilen ortalama sapma, çalışma süresi ve kullanılan işlemciler verilmiştir. Önerilen algoritma ile j30 problemlerinde öncelik koşullarının hepsinin çalıştırılması ve en iyi sonucun alınmasıyla ortalama sapma %4 ve çalışma süresi ortalama 22 ms, herhangi bir öncelik koşulunun tek başına alınarak yapıldığı çözümde ise sapmanın ortalama %8, çalışma süresi ortalama 3 ms olmaktadır. Diğer algoritmalarla kıyaslandığında ortalama sapmanın daha düşük olduğu söylenemese de çalışma süresi açısından daha hızlı olduğu görülmektedir. Bu verilere dayanarak geliştirilen algoritmanın çalışma süresi incelendiğinde oldukça hızlı çözüme ulaştığı söylenebilir.

Çizelge 4.19 J30 Problemleri için Ortalama Sapma ve Çalışma Süresi Değerleri

Yazar	Yöntem	Ort. Sapma	Çalışma Süresi (ms)	İşlemci
Fleszar, Hindi (2004)	VNS-Activity List	0.01	640	1.0 GHz
Mendes, Goncalves (2005)	Random Key Based GA	0.01	5200	1.33 Ghz
Hindi et al. (2002)	Activity List GA	0.37	170	400 MHz
Brucker (1997)	Tabu Search	0.4	15200	166 MHz
Önerilen Algoritma	Kural tabanlı	0.4	22	2.6 Ghz

4.4.2 Çoklu projelerin kaynak maliyetlerinin dikkate alınmadan çizelgelenmesi

Veri setlerinde yer alan problemler 2,5 ve 10'arlı gruplar halinde birleştirilerek çok projeli problemler elde edilmiş ve yazılım bu tarz problemlere uyarlanarak çözümler elde edilmiştir.

Çoklu proje çözümlerinin değerlendirilmesi için 2 performans ölçütü tanımlanmıştır:

(1) PG: Elde edilen çözümde proje tamamlanma zamanının kaynak kısıtlarını dikkate almadan CPM ile yapılan proje tamamlanma süresine yakınlığı,

(2) CPG: Her bir projenin tek başına çözüm yapıldığında elde edilen tamamlanma süresi ile çoklu projedeki tamamlanma süresi arasındaki farkın çoklu proje tamamlanma süresine bölümü.

CPG değerinin negatif olması çoklu çözümde projenin tekli çözüme göre daha erken tamamlandığını ifade etmektedir. Çizelge 4.20'de bütün ortalama CPG değerlerinin negatif olduğu görülmektedir. Çalışma sürelerine bakıldığında 30 aktiviteli j30 problemlerinin çözümünün ortalama 0,08 saniye 120 faaliyete sahip j120 problemlerinin 2,08 saniye olduğu görülmektedir.

4.4.3 Projelerin devresel kaynak maliyetlerinin dikkate alınarak çizelgelenmesi

Kaynak maliyetlerinin devresel olarak değiştiği veri setleri elde etmek için D1, D2, D3 adında üç farklı devresel veri seti tanımlanmıştır. PSPLIB problemlerinde kaynak maliyeti olarak bu devresel veri setlerinde yer alan maliyetler kullanılmıştır. D1 ve D2 setleri 4'er devre olarak tanımlanmış ve her kaynak için her devreye ait farklı kaynak maliyeti atanmıştır. D1 veri setinde tüm kaynakların maliyetleri devre boyunca eşit verilmiş, D2 setinde ise farklı verilmiştir. D3 setinde ise 6 devre için farklı maliyetler atanmıştır. Türetilen devresel veri setleri Çizelge 4.21'de verilmiştir.

Çizelge 4.20 Çoklu Projelerin Kaynak Maliyetsiz Çizelgenmesi

PROJE SAYISI		2			5			10		
VERİ SETİ	ÖNCELİK KURALI	PG	CPG	ÇALIŞMA SÜRESİ (ms)	PG	CPG	ÇALIŞMA SÜRESİ (ms)	PG	CPG	ÇALIŞMA SÜRESİ (ms)
J30	A	93	-0.22	8.4	97	-0.72	39.8	96	-1.20	141.0
	B	89	-0.26	8.5	93	-0.86	40.0	94	-1.57	145.0
	C	90	-0.23	8.5	94	-0.76	40.5	93	-1.37	149.0
	D	92	0.04	8.5	96	-0.73	40.5	94	-1.10	151.0
	E	94	-0.26	8.5	98	-0.65	40.5	97	-1.25	148.0
	F	91	-0.16	8.5	96	-0.74	39.8	96	-1.19	146.0
	G	90	-0.21	8.4	94	-0.67	38.8	93	-1.17	142.0
J60	A	91	-0.15	29.4	96	-0.45	147.0	97	-1.12	540.0
	B	98	0.66	30.0	95	-1.20	151.0	92	-1.46	544.0
	C	88	-0.51	30.2	94	-0.90	152.0	92	-1.15	549.0
	D	86	-0.21	30.3	97	-0.85	149.0	95	-1.14	586.0
	E	90	0.74	30.9	98	-0.55	150.0	98	-0.98	571.0
	F	81	-0.50	30.2	97	-0.42	148.0	96	-1.20	543.0
	G	87	-0.16	29.3	93	-0.58	149.0	93	-1.02	539.0
J90	A	90	-0.12	61.4	95	-0.65	310.0	96	-1.32	1179.0
	B	84	-0.19	62.5	93	-0.72	317.0	92	-1.25	1188.0
	C	86	-0.14	62.7	94	-0.69	312.0	92	-1.12	1225.0
	D	88	-0.12	69.8	92	-0.65	315.0	95	-1.19	1236.0
	E	92	-0.11	67.3	94	-0.82	308.0	98	-1.09	1195.0
	F	89	-0.13	619.0	93	-0.92	306.0	95	-1.33	1167.0
	G	87	-0.13	61.9	92	-0.59	302.0	92	-1.29	1159.0
J120	A	69	-0.25	110.0	77	-0.98	555.0	87	-1.10	2063.0
	B	54	-0.36	113.0	71	-1.12	567.0	82	-1.22	2098.0
	C	60	-0.24	113.0	76	-0.85	572.0	84	-1.36	2087.0
	D	63	-0.29	114.0	73	-0.78	563.0	86	-1.27	2089.0
	E	74	-0.22	113.0	82	-0.90	572.0	91	-1.19	2081.0
	F	64	-0.26	113.0	74	-0.86	558.0	84	-1.42	2075.0
	G	62	-0.27	111.0	72	-0.89	552.0	85	-1.11	2071.0

Türetilen devresel maliyet veri setleri ile PSPLIB veri setleri birlikte değerlendirilerek oluşturulan test problemleri maliyetler gözetilerek çözülmüştür. Elde edilen çözümlerin kaynak kısıtlarını dikkate almayan CPM çözümleri ile kıyaslanması yerine, yine aynı algoritmanın devresel kaynak maliyetlerini dikkate almadan elde edilen çözümle kıyaslanması daha uygun olacağı düşünülmüştür.

Çizelge 4.21 Devresel Maliyetler İçin Veri Setleri

KAYNAK	D1				D2			
	DEVRESEL MALİYETLER (TL)				DEVRESEL MALİYETLER (TL)			
	1	2	3	4	1	2	3	4
R1	10	8	12	14	15	20	10	5
R2	10	8	12	14	10	12	15	10
R3	10	8	12	14	8	14	10	15
R4	10	8	12	14	12	10	16	14
Zaman Aralığı	0-10	10-20	20-30	30-40	0-10	10-20	20-30	30-40

D3						
KAYNAK	1	2	3	4	5	6
R1	8	10	16	20	14	12
R2	6	10	14	18	16	14
R3	4	8	18	20	14	10
R4	6	6	12	14	16	18
Zaman Aralığı	0-10	10-20	20-30	30-40	40-50	50-60

Çizelge 4.22’de devresel maliyetlerin dikkate alınmadan erişilen çözüm ile gecikme maliyeti 25 TL ve 250 TL alınarak elde edilen çözümler kıyaslanmıştır. Karşılaştırma tablosunda kullanılan kısaltmaların açıklaması aşağıda verilmiştir:

DME : Devresel maliyetlerin eşit olduğu durum

PTZ: Proje tamamlanma zamanı. Projenin devresel maliyetleri dikkate almadan elde edilen çözümüne göre tamamlanma zamanı.

KKM: Kaynak kullanım maliyeti. Projenin önemsiz maliyetleri dikkate almadan çözülmesi ile oluşan kaynak kullanım maliyeti.

GS: Gecikme süresi. Projenin devresel maliyetler dikkate alınarak çözüldüğündeki proje tamamlanma zamanı ile devresel maliyetler dikkate alınmadan yapılan çözümün proje tamamlanma zamanı arasındaki fark.

GSY: Gecikme süresinin, proje maliyetler dikkate alınmadan yapılan çözümün proje tamamlanma zamanına bölümü.

KKMF: Devresel maliyetlerin dikkate alındığı ve alınmadığı çözümler arasındaki kaynak kullanım maliyetleri farkı.

KKMF: Kaynak kullanım maliyet farkının, devresel maliyetlerin dikkate alınmadığı çözümün kaynak kullanım maliyetine bölünmesi.

CS (ms): Çalışma süresi (milisaniye cinsinden)

Verilen 3 devre setinde de devresel maliyetlerin dikkate alınması ile yapılan çözümlerde toplam maliyetin azaldığı görülmüştür. Toplam maliyet; toplam kaynak kullanım maliyeti ve gecikme maliyetinin toplanması ile hesaplanmaktadır. Gecikme maliyetinin yüksek olduğu çözümde gecikme süresinin, gecikme süresinin düşük olduğu devreye göre daha düşük olduğu görülmektedir. Çalışma süreleri açısından bakıldığında 30 faaliyeti olan j30 problemleri için ortalama 0,05 saniye, 120 faaliyeti olan j120 problemleri için ortalama 0,48 saniye olduğu görülmektedir.

Çizelge 4.22 Devresel Maliyetler ile Yapılan Çözüm

DEVRE SETİ	VERİ SETİ	DME		Gecikme Maliyeti = 25 TL					Gecikme Maliyeti = 250 TL				
		PTZ	KKM(TL)	GS	GSY	KKMF(TL)	KKMFY	CS(ms)	GS	GSY	KKMF(TL)	KKMFY	CS(ms)
D1	J30	61.3	24,258	3.10	5%	-1,165	-4.8%	17	0.02	0.0%	-1,081	-4%	48
	J60	84.1	48,558	2.56	3%	-2,266	-4.7%	49	0.06	0.1%	-2,174	-4%	136
	J90	100.1	72,986	3.81	4%	-3,417	-4.7%	100	0.11	0.1%	-3,311	-4%	276
	J120	136.9	97,727	3.12	2%	-3,952	-4.0%	173	0.01	0.0%	-3,855	-4%	476
D2	J30	61.3	28,218	2.85	5%	-1,312	-4.6%	17	0.03	0.0%	-1,209	-4%	48
	J60	84.1	55,909	3.70	4%	-2,659	-4.8%	49	0.06	0.1%	-2,531	-5%	135
	J90	100.1	83,469	3.26	3%	-3,949	-4.7%	94	0.11	0.1%	-3,847	-4%	258
	J120	136.9	106,147	3.41	2%	-4,371	-4.1%	174	0.01	0.0%	-4,266	-4%	477
D3	J30	61.3	27,156	4.80	8%	-1,638	-6.0%	17	0.14	0.2%	-1,235	-4%	47
	J60	84.1	54,998	2.77	3%	-3,087	-5.6%	48	0.41	0.5%	-2,947	-5%	133
	J90	100.1	83,241	2.71	3%	-4,580	-5.5%	94	0.26	0.3%	-4,408	-5%	258
	J120	136.9	103,437	4.24	3%	-4,673	-4.5%	177	0.07	0.1%	-4,470	-4%	488

4.4.4 Çoklu projelerin devresel kaynak maliyetlerinin dikkate alınarak çizelgelenmesi

Devresel maliyet değişikliklerinin çoklu projeler ile denenmesi amacı ile PSPLIB j120 veri seti projeleri 10'arlı birleştirilerek çoklu projeler elde edilmiştir. Elde edilen bu projeler D1, D2, D3 devre setleri ile çözülmüştür. Çizelge 4.23'te elde edilen çözümlerin kaynak maliyeti dikkate alınmadan elde edilen çözüm ile kıyaslanması görülmektedir. 120 faaliyetten oluşan j120 problemlerinin 10'arlı birleştirilmesi ile elde edilen projeler 1200 faaliyetten oluşmaktadır. Çalışma süresi öncelik kuralları ile elde edilen çözümlerin ortalama çalışma süresi olarak hesaplanmıştır. Gecikme maliyeti 25 TL olduğunda ortalama 9,1 saniye gecikme maliyeti 250 TL olduğunda ise 25,2 saniye olarak ölçülmüştür.

Çizelge 4.23 Devresel Maliyeler ile Yapılan Çözüm –Çoklu Proje

	DME				Gecikme Maliyeti = 25					Gecikme Maliyeti = 250		
	PTZ	KKM(TL)	GS	GSY	KKMF(TL)	KKMFY	CS(ms)	GS	GSY	KKMF(TL)	KKMFY	CS(ms)
D1	127.1	\$562,033	4.28	3.4%	-\$14,070	-3%	9151	0.10	0.1%	-\$13,859	-2%	25245
D2	127.1	\$640,236	7.05	5.5%	-\$15,788	-2%	9112	0.00	0.0%	-\$16,512	-3%	25135
D3	127.1	\$378,860	8.20	6.5%	-\$11,524	-3%	9143	0.08	0.1%	-\$8,678	-2%	25222

4.4.5 Test sonuçlarının değerlendirilmesi

Önerilen algoritma ile PSPLIB veri setleri öncelikle verildiği şekilde çözülmüş ve sonuçlar en iyi sonuçlarla kıyaslanmış ve en iyi çözüme yakınsayan çözümler ürettiği gözlenmiştir. Farklı veri setleri üzerinde elde edilen sonuçların tutarlı olduğu gösterilmiştir. Faaliyet sayısı arttıkça çalışma süresinin ve en iyi çözümden sapmanın arttığı gözlemlenmiştir. Çalışma süresi açısından diğer algoritmalarla kıyaslandığında, algoritmanın milisaniyeler mertebesinde çalışan ve hızlı sonuç veren bir algoritma olduğu söylenebilir. Algoritmanın kısa sürede sonuç vermesi, başlangıç çözüm üretme ve karar desteğinde kullanılması açısından olumludur.

Algoritmanın çoklu projeler üzerinde çalıştığının gösterilmesi için kullanılan veri setlerindeki projeler birleştirilerek çoklu projeler türetilmiş ve çözümler elde edilmiştir. Elde edilen çözümlerin tutarlılığı ve etkinliği gösterilmiştir.

Kaynak maliyetlerinin devresel olarak deęiřtięi durumlarda algoritmanın etkinlięini gözlemlenmesi için devre setleri belirlenmiř ve bu devrelerdeki kaynak maliyetleri dikkate alınarak çözüme ulařılmıřtır. Elde edilen çözümlerde toplam maliyetlerin azaldıęı gözlemlenmiřtir. Gecikme maliyetinin azalmasının çözümden gecikmenin artmasını ve toplam maliyetin düşmesini sağladığı görülmüřtür. Çalışma süresi açısından bakıldığında etkili sonuçlar elde edilmiřtir.

Devresel maliyet deęişikliklerinin çoklu projeler üzerinde denenmesi için veri setlerinden yararlanılarak çoklu projeler türetilmiř ve devre setleri ile çözümler üretilmiřtir. Üretilen çözümlerde toplam maliyetin azaldığı gösterilmiřtir. Çalışma süresinin maliyetlerin dikkate alınmadığı çözüme göre arttığı gözlemlenmiřtir.

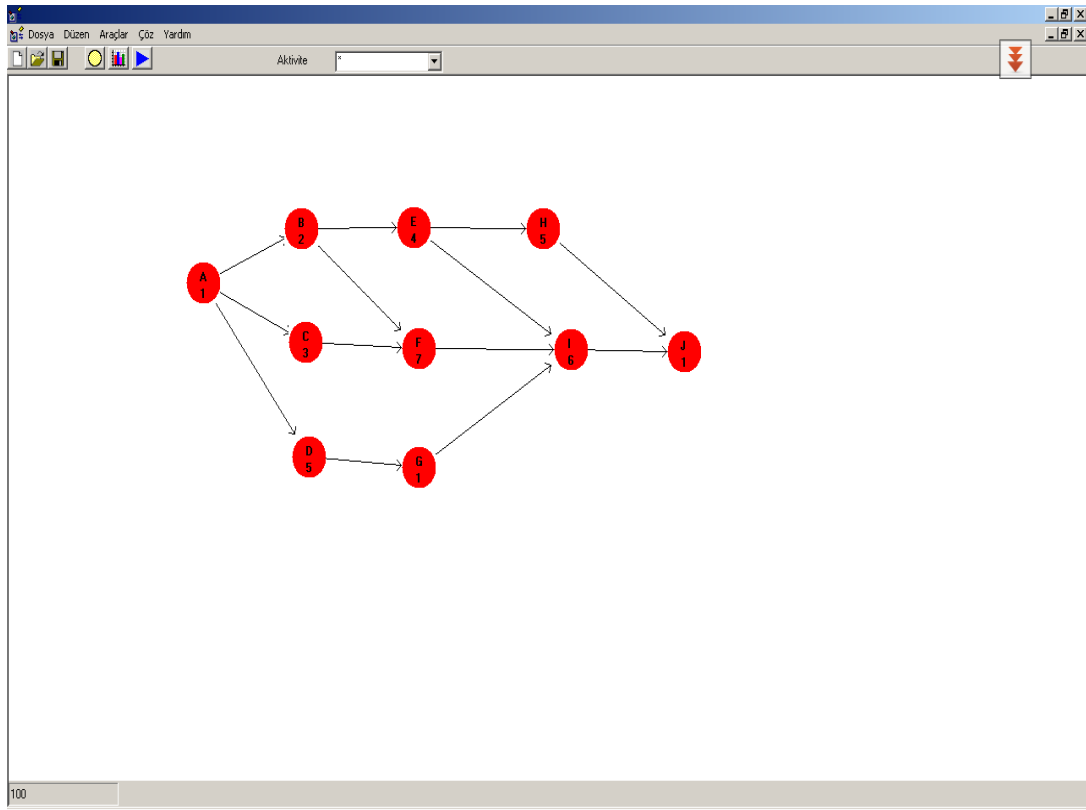
Bu bilgilere dayanılarak algoritmanın kaynak maliyetlerini dikkate alarak ve almayarak, tekli ve çoklu projeler için tutarlı çözümler üreteceęi söylenebilir. Devresel maliyetlerin dikkate alınmasıyla toplam maliyetin tekli ve çoklu projelerde azaltılabileceęi gösterilmiřtir.

4.5 Geliřtirilen Karar Destek Sistemi

Proje kaynaklarının dengelenmesinde kullanıcıya çözüm önerisi sunabilen ve kendi geliřtireceęi çözümleri deęerlendirme imkânı sunabilen bir yazılım C# dili ile VS. NET ortamında geliřtirilmiřtir. Hazırlanan yazılım Windows tabanlıdır ve tek başına çalışan(standalone) ve tek kullanıcıya bir sistemdir. Sistemde kullanıcının verileri kolayca girebilmesi, çözüm oluşturabilmesi ve oluşturulan çözümleri deęerlendirebilmesi hedeflenmiřtir. Hazırlanan sistem ekran görüntüleri ile anlatılmıřtır.

Kullanıcı isterse tüm proje faaliyetleri, öncüllük ilişkilerini ve kaynak atamalarını bu yazılım üzerinde yapabilir veya bilinen proje yönetimi yazılımlarında hazırladığı projeleri aktarabilir.

Yazılımın kullanımında kullanıcı ya başka bir projeden parametreleri aktarır ya da yazılımı kullanarak yeni bir proje oluşturur. Yeni bir proje oluşturulduğunda öncelikle faaliyetler ve parametreleri (faaliyet süresi, faaliyet adı v.s) girilerek daha sonra serim diyagramı oluşturulur.



Şekil 4.7 Serim Diyagramı Oluşturulması Ekran Görünümü

Faaliyetler eklendikten sonra faaliyetler arasındaki ilişkiler faaliyet ilişkileri formu aracılığı ile oluşturulur. Eklenen her ilişki FS (finish to start) olarak eklenir daha sonra ilişki tipleri değiştirilmek istenirse faaliyet ilişkileri formu aracılığı ile düzenlenir. Bu form aracılığı ile her işlem ve buna bağlı işlemlerin ilişki türleri belirlenir.

Faaliyet	İlişki	Tip	Süre
A	<input checked="" type="checkbox"/> B	FS	0
	<input checked="" type="checkbox"/> C	FS	1
	<input checked="" type="checkbox"/> D	SS	3
	<input type="checkbox"/> E		
	<input type="checkbox"/> F		

Şekil 4.8 Faaliyet İlişkileri Formu

Kaynakların tanımlanması sırasında projede yer alan tüm kaynaklar tanımlanabileceği gibi sadece dengelenmesi istenen kaynaklar da tanımlanabilir. Genellikle projelerde proje boyunca yer alacak personel veya ekipmanın dengelenmesi istenmeyeceğinden bu tip kaynakların göz önüne alınması bir fayda sağlamayacaktır. Kaynaklar tanımlanırken kaynak miktarı elde bulunan ya da bulunması planlanan miktar olarak girilir.

KAYNAK ADI	MIKTAR	BİRİM
Formen	2	Adet
İşçi	11	Adet
Mobil Vinç	2	Adet
Kamyon	2	Adet

EKLE/SİL

Şekil 4.9 Kaynak Bilgileri Formu

Sistemde her gün, bir devre olarak tanımlıdır. Yeni bir devre tanımlanırken başlangıç ve bitiş süresi girilir ve aradaki günler o devreye atanır.

Devre Kodu	Başlangıç	Bitiş
1	1	90
2	91	180
3	181	270
4	271	360

Şekil 4.10 Devre Bilgileri Formu

Kaynak maliyeti her devre için farklı girilebileceği gibi bazı devreler veya bütün devreler için aynı değer olarak da girilebilir. Burada girilen maliyetler ile ters orantılı olarak devrelik kaynak kapasiteleri sistem tarafından hesaplanır. Kullanıcı tarafından istenirse devresel kaynak kapasiteleri değiştirilebilir.

Devre	Maliyet
1	0,5
2	0,5
3	0,6
4	0,5

Şekil 4.11 Devresel Kaynak Maliyetleri Formu

Faaliyetler; faaliyet kodu ve faaliyet süresi olarak serim diyagramı oluşturulurken tanımlanır. Faaliyet tanımı ve faaliyete kaynakların atanması faaliyet bilgileri formu aracılığı ile yapılır. Faaliyetlere kaynaklar atanırken faaliyetin ihtiyacı olan toplam kaynak yerine faaliyetin birim sürede kullandığı kaynak girilir.

Faaliyet Bilgileri	
Faaliyet Kodu	A
Faaliyet Tanımı	Kazı Yapılması
Faaliyet Süresi	23

Kaynaklar	
Formen	3
İşçi	20
Ekskavator	2

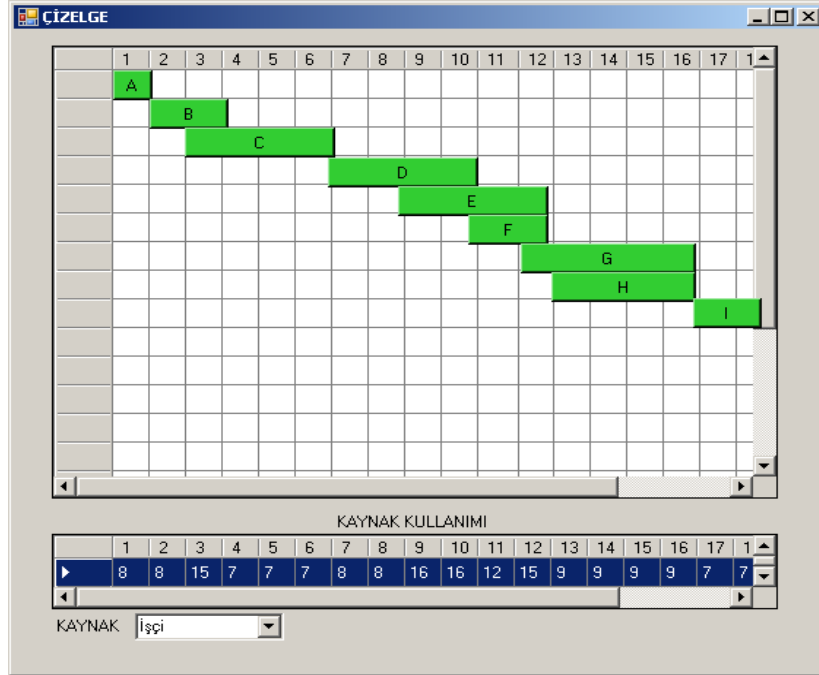
Kaynak Ekle/Sil

Şekil 4.12 Faaliyet Bilgileri Formu

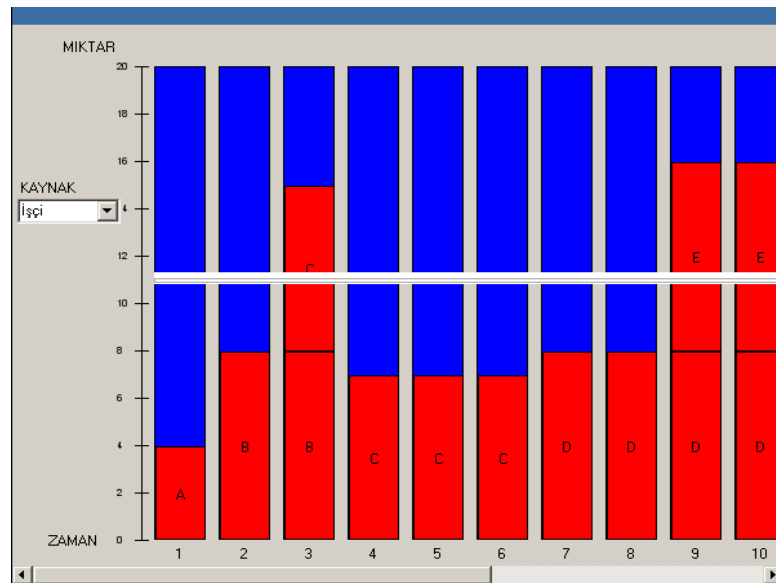
Kullanıcı tüm bilgileri girdikten sonra çözüm aşamasına geçer. Çözüm için kullanılacak algoritma daha önce açıklanan algoritmadır.

Kullanıcı oluşan çözümün sonuçlarını çizelge üzerinde inceleyebileceği gibi çizelgede elle değişiklikler yaparak yaptığı değişikliklerin etkisini gözlemleyebilir ve kendi çözümünü üretebilir.

Kaynak kullanımlarını kaynak kullanım grafiği ile de görmek mümkündür. Miktar/Zaman grafiğinde birim zamanda kullanılan kaynak miktarı ve hangi faaliyetler tarafından ne kadar kullanıldığı görülebilir.



Şekil 4.13 Faaliyet Çizelge Formu



Şekil 4.14 Kaynak Kullanım Grafiği

4.6 Geliştirilen Algoritmanın Yapı Sektörüne Uygulanması

Yapı sektöründe proje yönetimi; projelerin istenilen performans ve teknolojik düzeyde, zamanında bütçe sınırları içinde, başarılı tamamlanabilmesi için kaynakların etkili olarak kullanılmasını sağlamayı amaçlamaktadır. Projenin kapsamı ve karmaşıklığı, zamanlama, coğrafik konum, sözleşme şartları, finansal düzenlemeler proje yönetimi organizasyonunu etkileyen önemli hususlardır.

Yapı sektörü projeleri mevsimsel ve iklimsel şartlardan etkilenen faaliyetler barındırır. Mevsim ve iklim değişikliklerine göre faaliyetlerin yapılış biçimi ve kaynak gereksinimleri değişiklik gösterebilir.

Yapı sektöründe kullanılan hammadde fiyatları (demir, çelik, çimento gibi) devresel olarak değişiklik göstermektedir.

Kaynak kısıtlarının dikkate alınmadan hazırlanan bir iş programı oluşturulması, iş programının sahada uygulanamayacağı anlamına gelir. İş programının gerçeği yansıtması için kaynak kapasitelerine uygun bir çizelge yapılmalıdır.

Proje yöneticileri, maliyet değişikliklerinden kaçınmak için, fiziksel veya yapısal koşullar gereğiyle kaynak kapasitelerini değiştirmek isteyeceklerdir. Kaynak kapasiteleri devresel olarak belirlenmekte ve devre boyunca sabit tutulmaya çalışılmaktadır. Ancak bu değişiklikler dikkate alınmadan yapılan çizelgelerden elde edilen iş programları uygulanabilir olmayacaktır.

Uygulama kapsamında yapı sektöründe faaliyet gösteren bir firmanın, yüklenicisi olduğu bir ofis binası projesi ele alınmıştır. Proje faaliyetleri tanımları ve süreleri Ek 2’de verilmiştir.

Kaynak kısıtları olarak işçilik, hazır beton, hareketli platform ve çimento kaynakları seçilmiştir. Faaliyetlerin kaynak gereksinimleri ve faaliyetler arası ilişkiler Ek 3’de verilmiştir.

Proje devreleri mevsimlere göre belirlenmiş ve Çizelge 4.24 de verilmiştir. Devresel Kaynak kapasiteleri ve maliyetleri Çizelge 4.25 de verilmiştir.

Çizelge 4.24 Uygulama Projesi Devreleri

Devre	Zaman Aralığı
1	Haziran - Ağustos
2	Eylül - Kasım
3	Aralık - Şubat
4	Mart-Mayıs

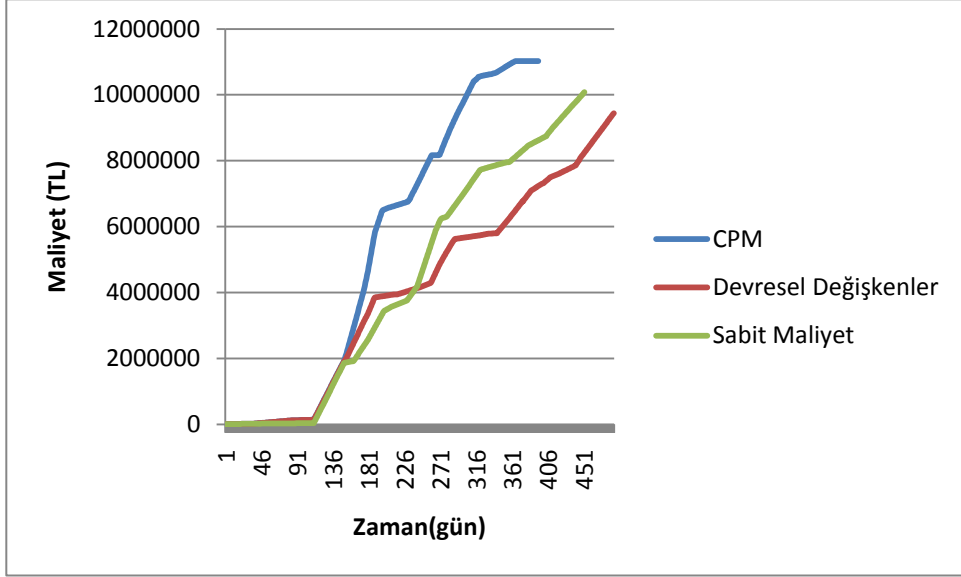
Çizelge 4.25 Uygulama Projesi Devresel Kaynak Maliyet ve Kapasiteleri

Devre	Hareketli Platform		Hazır Beton		Çimento		İşçilik	
	Kapasite (Makina saat)	Birim Maliyet (TL)	Kapasite (m ³)	Birim Maliyet (TL)	Kapasite (kg.)	Birim Maliyet (TL)	Kapasite (Adam saat)	Birim Maliyet (TL)
1	12	10	40	200	250	0,5	500	8
2	24	10	30	250	250	0,5	400	10
3	36	10	25	320	200	0,6	200	20
4	24	10	35	225	250	0,5	300	14

CPM ile elde edilen çizelgeye göre proje tamamlanma süresi 392 gün toplam kaynak maliyeti 11 022 983 TL olarak bulunmuştur. CPM ile elde edilen iş programı kaynak kapasitelerini sağlamadığından bu iş programının kullanılması mümkün olmayacaktır.

Devresel değişiklikler dikkate alınmadan, hareketli platform 24 saat, hazır beton 33 m³/gün, çimento 250 kg., işçilik 380 adam saat alınarak sabit kaynak kapasiteleri ile çizelgeleme yapılmıştır. Çizelgeleme sonucunda proje süresi 452 gün toplam kaynak kullanım maliyeti 10 081 282 TL olmaktadır.

Verilen devresel kaynak kapasite ve maliyetleri ile gecikme maliyeti 10 000 TL/gün alınarak elde edilen çözümde, proje tamamlanma süresinin 489 gün ve proje kaynak kullanım maliyetinin 9 443 118 TL olduğu görülmektedir. Elde edilen iş programı Ek 4' te verilmiştir. Sabit maliyetler kullanılarak yapılan çözüm ile kıyaslandığında ek gecikme maliyeti $(489-452)*10\ 000\ TL = 370\ 000\ TL$ olacaktır. Devresel değişkenlerin dikkate alınmasıyla sabit kaynak kapasitelerini dikkate alan çözüm arasında toplam maliyet farkı $(10\ 081\ 282 - 9\ 443\ 118 - 370\ 000 = 268\ 164\ TL)$ olarak hesaplanmaktadır.



Şekil 4.15 Uygulama Projesi Birikimli Toplam Maliyetler

5. SONUÇ VE ÖNERİLER

Rekabetin giderek artması, teknolojik gelişmeler ve özellikle bilgi teknolojileri uygulama ve kullanım alanlarının hızla artması firmaların proje yönetim tekniklerine ilgisini giderek arttırmaktadır.

Bu kapsamda proje yönetimi; gereksinim ve beklentilerin karşılanması amacıyla bilgi, yetenek, araç ve tekniklerin verimli kullanılması yanında bütün sürecin işleyişini kolaylaştıran bir yaklaşım sergilemektedir. Proje yönetimi, belli bir hedefi gerçekleştirmek için insanların, kaynakların ve zamanın birbiriyle uyumlu ve verimli kullanılmasını sağlamaktadır.

Projenin kaynaklar göz önüne alınmadan çizelgelenmesi, oluşacak çizelgelerin geçersiz ve kullanışsız olacağı anlamına gelir. Kaynak kısıtlarının dikkate alınması iş programının gerçeğe daha uygun olmasını sağlayacaktır. Kaynak kısıtlarını sağlayan ve devresel maliyet ve kapasite değişikliklerini dikkate alarak hazırlanmış iş programları daha düşük maliyetli olabileceği gibi proje yöneticisi için daha gerçekçi bir iş programı olacaktır.

Bu çalışmada kaynak maliyetlerinin ve kapasitelerinin devresel değişkenlik gösterdiği durumlarda kullanılmak üzere sezgisel bir algoritma önerilmiş ve bir karar destek sistemi geliştirilmiştir.

Önerilen algoritmanın literatürde yer alan test problemleri üzerinde denenmesi için bir yazılım geliştirilmiş ve elde edilen sonuçlar değerlendirilmiştir. Test problemlerinin denenmesi ile elde edilen sonuçlara dayanarak, algoritmanın tutarlı ve geçerli sonuçlar ürettiği, hızlı sonuç veren bir algoritma olduğu söylenebilir. Algoritmanın hızlı çalışıyor olması, karar destek sistemlerinde, başlangıç çözüm oluşturmakta ve diğer proje yönetim yazılımların içerisinde kullanılmasını da sağlamaktadır.

Yapılan çalışmada, devresel kaynak kapasite değişiklikleri dikkate alınarak kaynak kısıtlarını sağlayan iş programları elde edilmiştir. Değişken kaynak maliyeti

olduđu durumlarda önerilen yöntemle elde edilen sonuçların sabit maliyet ve kapasite kısıtı ile elde edilen sonuçlardan daha düşük maliyetli olabileceđi gösterilmiştir.

Geliştirilen karar destek sistemi hem algoritmanın kullanılması hem de kullanım kolaylığı açısından proje yöneticileri için karar desteđi sağlması amacı ile hazırlanmıştır. Yazılım C# dili kodlanmış ve nesne tabanlı bir yazılım olması nedeniyle yeni özelliklerin ve eklentilerin yapılabileceđi bir yazılımdır. Yazılımda yer alan bileşenler proje yönetim yazılımları tarafından kullanılabilir.

Proje yönetimi ve kaynak çizelgeleme tekniklerinin kullanımı yapı sektöründe geniş bir kullanım alanına sahiptir. Yapı sektöründe kaynak çizelgeleme teknikleri kullanımı yurtiçi firmaları için %32 yurtdışı firmaları için ise %44 olarak ölçülmüştür ve bu oranın giderek artacağı beklenmektedir (Uğur, 2007). Bu da yapı sektöründe proje yönetiminde kaynak çizelgelemenin önemimin artacağını göstermektedir.

Önerilen yöntem ile yapı sektöründe yer alan bir firmanın verileri ile hazırlanan bir problem çözülmüş ve olumlu sonuçlar elde edilmiştir. Elde edilen iş programının uygulanması ile hem daha gerçekçi bir proje çizelgesi elde edilmiş olmakta hem de daha düşük bir maliyet ile proje tamamlanmaktadır.

Geliştirilen algoritma tek modlu projeler için tasarlanmıştır. Çok modlu projeler için de benzer bir algoritma tasarlanabilir.

KAYNAKLAR DİZİNİ

- Aral, N., 1979, Yapı Üretiminde Proje Yönetimi İçin Üretkenlik Kavramına Dayalı Bir Değerlendirme Modeli, İTÜ, İstanbul,230.
- Barkley, B.T., Saylor, J.H., 1994, Costumer driven project management, New York, Mc Graw Hill, 34.
- Blazewicz, J., Lenstra, J.K., Rinnooy Kan, A.H.G., 1983, Scheduling subject to resource constraints: Classification and complexity, Discrete Applied Mathematics, 5, 11-22.
- Barutçugil, S., 1983, Proje yönetiminde örgütlenme sorunu ve proje örgütü, Uludağ Üniversitesi İktisat ve İdari Bilimler Dergisi, C:IVS:2, 157.
- Bayar, D., 1990, Yatırım ve Proje Değerlemesi, Anadolu Üniversitesi Açıköğretim Fakültesi Yayınları, No 44.
- Biroğul, S., 'Genetik Algoritma Yaklaşımıyla Atölye Çizelgeleme', Yüksek lisans tezi, Gazi Üniversitesi Fen Bilimleri Enstitüsü, Ankara, Ocak, 2005.
- Bock, D.B., Patterson, J.H., 1990, A Comparison of Due Date Setting, Resource Assignment and Job Preemption Heuristics for the Multiproject Scheduling Problem, Decision Sciences, 21, ss. 387– 402.
- Boctor, F.F., 1990, Some Efficient Multi-Heuristic Procedures for Resource Constrained Project Scheduling, European Journal of Operational Research, 49, ss. 3–13.
- Boctor, F.F., 1996a, An adaptation of the simulated annealing algorithm for solving resource constrained project scheduling problems, International Journal of Production Research, 34, 2335-2351.
- Bouleimen, K., Lecocq, H., 2003, A New Efficient Simulated Annealing Algorithm for the Resource Constrained Project Scheduling Problem and Its Multiple Mode Version, European Journal of Operational Research, 149, ss. 268–281.
- Brucker, P., Drexl, A., Mohring, R., Neumann, K., Pesch, E., 1999., Resource-constrained project scheduling: Notation, classification, models, and methods. European Journal of Operational Research 112, 3–41.

KAYNAKLAR DİZİNİ (devam)

- Burke, R., 1999, Project management 3rd ed., 1999, John Wiley, 767 p.
- Callahan, T.M., Quackenbush, G.D. and Rowings, E.J., 1992, Construction Project Scheduling., New York: McGraw Hill, Inc.
- Christofides, N., Alvarez-Valdés, R. and Tamarit, J.M., 1987, Project scheduling with resource constraints: A branch and bound approach, *European Journal of Operational Research* 29, 262–273.
- Cleland, I.D., Ireland, R.L., 2002, Project Management: Strategic Design and Implementation, New York: McGraw Hill Companies, Inc.
- Cooper, D.F., 1976, Heuristics for scheduling resource constrained scheduling projects: an experimental investigation, *Management Science*, 22, 1186–1194.
- Cottrell, D., 1999, Simplified program evaluation and review technique, *Journal of Construction Engineering and Management*, Jan/Feb, 125/1:17
- Davis, E.W., Patterson, J.H., 1975, A comparison of heuristics and optimum solutions in resource constrained project scheduling, *Management Science* 21, 944–955.
- Davis, E.W., Kurtuluş, 1982, I.S., Multi-project scheduling: Categorization of heuristic rules performance, *Management Science*, 28(2) 161–172.
- Drexl, A., Grünewald, J., 1993, Nonpreemptive multi-mode resource-constrained project scheduling, *IIE Transactions*, 25, 5, 74-81.
- Drexl, A., Juretzka, J., Salewski, F., Schirmer, A. 1999, New modelling concepts and their impact on resource-constrained project scheduling, *J. Weglarz içinde*, 413-432.
- Dorndorff, U., Pesch, E., 1995, Evolution based learning in a job shop scheduling environment, *Computers and Operations Research*, 2, 25-40.
- Dumond, J., Mabert, V.A., 1988, Evaluating project scheduling and due date assignment procedures: An experimental analysis, *Man. Science* 34 101–118.

KAYNAKLAR DİZİNİ (devam)

- Elmaghraby, S.E., 1995, Activity Nets: A Guided Tour Through Some Recent Developments, *European Journal of Operational Research* 82, 383-408.
- El-Rayes, K., Moselhi, O., 1998., “Resource-Driven Scheduling of Repetitive Activities on Construction Projects.” *J. Constr. Engrg. and Mgmt.*, ASCE, 107(2), 18-27.
- Fendley, L.G., 1968, Towards the Development of A Complete Multiproject Scheduling System”, *Journal of Industrial Engineering*, October, pp 505–515.
- Fleming, W. Q., Koppelman, J. M., July 1999, Earned value project management an introduction, Primavera Systems Inc.
- Fleming, W. Q., Koppelman, J.M., July 1998, Earned value project management a powerful tool for software projects, Primavera Systems Inc.
- Gitmez, Y., 1998, Proje Yönetiminde Yönetim Fonksiyonları ve Bir Uygulama, Anadolu Üniversitesi, Eskişehir, 88.
- Glover, F., 1989, Tabu search – Part I, *ORSA Journal on Computing*, 1, 190-206.
- Glover, F., 1989, Tabu search – Part II, *ORSA Journal on Computing*, 1, 190-206.
- Gonçalves, J.F., Mendes, J.J.M., Resende, M.G.C. (2004) “A Genetic Algorithm for the Resource Constrained Multi-Project Scheduling Problem”, AT&T Labs Technical Report, pp.119.
- Gordon, J., Tulip, A., 1997, Resource scheduling. *International Journal of Project Management* 15 (1997), pp. 359–370.
- Hartmann, S., 1997, Project scheduling with multiple modes: A genetic algorithm, Çalışma Raporu 435, Manuskripte aus den Institut für Betriebswirtschaftslehre der Universitaet Kiel, Almanya.
- Herrmann, J.W., Lee, C.-Y., Hinchman, J., 1995, Global job shop scheduling with a genetic algorithm, *Production and Operations Management*, 4, 30-45.

KAYNAKLAR DİZİNİ (devam)

- Herroelen, W.S., P. Van Dommelen and E.L. Demeulemeester, 1997, Project network models with discounted cash flows: A guided tour through recent developments, *European Journal of Operational Research*, 100, 97-121.
- Herroelen, W.S., Demeulemeester, E. and Reyck, B.D., 1998, A classification scheme for project scheduling, in: *project scheduling: recent models, algorithms and applications*, ed. J. Weglarz, KluwerAcademic, Amsterdam, pp. 1–26.
- Hindi, K.S., Yang, H., Fleszar, K., 2002, An evolutionary algorithm for resource-constrained project scheduling. *IEEE Transactions on Evolutionary Computation* 6, 512–518.
- Holland, J.H., 1975, *Adaptation in natural and artificial systems*, The University of Michigan Press, Ann Arbor, Michigan.
- Ives, M., Mart 2005, *Identifying the Contextual Elements of Project Management Within Organizations and Their Impact on Project Success*, *Project Management Journal*.
- İpeköz, B., 1994, *PERT analizi, Yüksek Lisans Tezi, İstanbul Üniversitesi, Sosyal Bilimler Enstitüsü, İstanbul, 44, (yayınlanmamış)*.
- Kamburowsky J., Michael, D., and Stallmann, F.M., 2000, *Minimizing the Complexity of an Activity Network*," *Networks*, Vol. 36(1) 47-52, John Wiley & Sons, Inc.
- Kirkpatrick, C.A., Levin, R. I., 1973, *PERT ve CPM İle Planlama Ve Denetim*, ODTÜ, İdari Bilimler Fakültesi, 2.Baskı, Yayın:12.
- Kirkpatrick, S., Gelatt, Jr., C.D., Vechhi, M.P., 1983, *Optimization by simulated annealing*, *Science*, 220, 671-680.
- Koçel, T., 1993, *Proje Yönetimi ve Matriks Organizasyon*, İstanbul Üniversitesi İşletme Fakültesi Dergisi, 22 (1), 66.
- Kelley, J.E., 1963, *The critial-path method: Resources planning and scheduling*, in: *industrial scheduling*, Prentice-Hall, New Jersey, 347–365.

KAYNAKLAR DİZİNİ (devam)

- Khattab, M., Soyland, K., 1998, Limited-Resource Allocation in Construction Projects, Proceeding of the Associated Schools of Construction, Annual Conference, Portland, Maine, April 15-18, 1998.
- Kolisch, R., 1996, Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation, *European Journal of Operational Research* 90(2), 320–333.
- Kolisch, R., Hartmann, S., 1998, Heuristic algorithms for the resource-constrained project scheduling problem: classification and computational analysis, in: *Project Scheduling: Recent Models, Algorithms and Applications*, ed. J.W. eglarz, Kluwer Academic, Amsterdam, 147–178.
- Krajewski, L.J., Thompson, H.E., 1981, *Management Science: quantitative methods in context*, John Wiley & Sons, Inc., 274.
- Kurtuluş, I.S., Davis, E.W. (1982) “Multi-project Scheduling: Categorization of Heuristic Rules Performances”, *Management Science*, 28 (2), ss. 161–172.
- Kurtuluş, I.S., Narula, S.C. (1985) “Multi-project Scheduling: Analysis of Project Performance”, *IIE Transactions*, 17 (1), ss 58–66.
- Lawrence, S.R., Morton, T.E., 1993, Resource-constrained multi-project scheduling with tardy costs: comparing myopic, bottleneck and resource pricing heuristics, *European Journal Of Operational Research*, 64, 168-187.
- Lockyer, K.G., Gordon, J.H., 1996, *Project management and project network techniques*, 6th edition., Pitman Publishing, London.
- Lu, M., Li, H., 2003. “Resource-activity critical-path method for construction planning”, *J. Constr. Eng. Manage.* 129, pp. 412–420.
- Malcolm, D.G., Roseboom, J.H., Clark, C.E., Fazar, W., 1959, Applications of a technique for research and development program evaluation, *Operations Research*, 7, 646-669.
- Marmel, E., 2004, *Microsoft Office Project, 2003*, Wiley Publishing Inc.
- Mori, M., Tseng, C.C., 1997, A genetic algorithm for multi-mode resource constrained project scheduling problem, *European Journal of Operational Research*, 100, 134-141.

KAYNAKLAR DİZİNİ (devam)

- Moder,J., Phillips,R., 1970,Project Management With CPM and PERT, New York, Van Nostrand Reinhold Co.,360.
- Paksoy, S., 2007, Genetik Algoritma İle Proje Çizelgeleme, Çukurova Üniversitesi,Sosyal Bilimler Enstitüsü, İşletme A.B.D., Doktora Tezi (Yayınlanmamış).
- Patterson, J.H., Huber, W.D., 1974, A horizon varying, zero-one approach to project scheduling, *Management Science*, 2, 990-998.
- Patterson, J.H., Roth, G.W., 1976, Scheduling a project under multiple resource constraints: a zeroone programming approach, *AIIE Transactions*, 8, 449-455.
- Patterson, J.H., Slowinski, R., Talbot, F.B., Weglarz, J., 1989, An algorithm for a general class of precedence and resource constrained scheduling problem.
- Peşkircioğlu, N., 1989, Proje yönetimi ve gelişmekte olan ülkelerdeki uygulama özellikleri, Ankara, MPM Yayınları, 4.
- Pritsker, B., Watters, L.,Wolfe, P.M., 1969, Multiproject scheduling with limited resources: A zero–one programming approach. *Management Science* 16, pp. 93–107.
- Ray, M., Rinzler, A., 1993, The new paradigm in business, Tarcher/Perigee Books, New York.
- Rowe, K., 1975, Management techniques for civil engineering construction, Applied Science Publishers Ltd., 129.
- Sampson S.E., Weiss, E.N., 1993, Local search techniques for the generalized resource-constrained project scheduling problem. *Naval Research Logistics* 40 (1993), pp. 665–675.
- Schirmer, A., 2000, Project Scheduling with Scarce Resources, Verlag Dr. Kovacs, Hamburg.
- Sht, A., Bard, J.F., Globerson, S., 1994, Project management, Engineering, Technology and Implementation, Prentice-Hall, New Jersey, USA.

KAYNAKLAR DİZİNİ (devam)

- Simpson, W.P., Patterson P.H., 1996, A multiple-tree search procedure for the resource-constrained project scheduling problem. *European Journal of Operational Research* 89 (1996), pp. 525–542.
- Sprecher, A., 1994, Resource-constrained project scheduling: exact methods for the multi-mode case, No. 409 in *Lecture Notes in Economics and Mathematical Systems*, Springer Verlag, Berlin.
- Stevenson, W. J., 1996, *Production/operations management*, IRWIN, Inc., 784.
- Talbot, F.B., Patterson, J.H., 1978, An efficient integer programming algorithm with network cuts for solving resource-constrained scheduling problems, *Management Science*, 24, 1163-1174.
- Talbot, F.B., 1982, Resource-constrained project scheduling with time-resource tradeoffs. The nonpreemptive case, *Management Science*, 28, 1197-1210.
- Thierauf, R. J., 1978, *An Introduction to Operations Research*, John Wiley & Sons, Inc., A Wiley / Hamilton Publication, 173.
- Turan, A., 1995, *Proje planlama ve kullanılan teknikler*, Ankara, MPM Yayınları.
- Tütek, H.H., Gümüšoğlu, 2000, Ş., *Sayısal Yöntemler: Yönetmel Yaklaşım*, Yenilenmiş 3. Bası, Beta Basım Yayım Dağıtım A.Ş., İstanbul.
- Trevor, Y., 1998, *Daha İyi Nasıl– Proje Yönetimi*, Timaş Yayınları, İstanbul, 168.
- Uğur, L.O., 2007, *TMB Üyesi İnşaat Firmalarının Planlama, Yapı Maliyeti Hesaplama ve Risk Yönetimi Yaklaşımı*, TMB 2007, Ankara. 25.
- Ulusoy, G., 1996, *Proje Planlama Yönelmleri*, Tübitak Yayınları, Ankara.
- Ulusoy, G., Özdamar, L., 1996, A framework for an interactive project scheduling system under limited resources, *European Journal of Operational Research*, 90, 362-375.
- Ünle, M., 2007, *Çoklu Proje Yönetimi ve Yerel Yönetimde Bir Uygulama*, Eskişehir Osmangazi Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek Lisans Tezi, Eskişehir.
- Vanhoucke, M., Demeulemeester, E., Herroelen, W., 2001, An exact procedure for the resource constrained weighted earliness-tardiness project scheduling problem, *Annals of Operations Research*, 102, 179-196.

Watters, L. J., 1967, Reduction of integer polynomial programming problems to zero-one linear programming problems, Operations Research, Vol. 15, No. 6, 1171-1174.

Yamak, O., 1994, *Üretim Yönetimi*, 1.Baskı, Alfa Basım Yayım Dağıtım, İstanbul, 266.

Wermter, M., 1996, *Stratejik Proje Yönetimi*, Evrim Yayınevi, İstanbul, 478.

EKLER

Ek 1 – Algoritma C# Kaynak Kodu

Ek 2 – Uygulama Projesi Faaliyet Tanımları

Ek 3 – Uygulama Projesi Faaliyet Öncüllük İlişkileri, Süre ve Kaynak Gereksinimleri

Ek 4 – Uygulama Projesi CPM ve Önerilen Algoritma İş Programı

Ek 1 – Algoritma C# Kaynak Kodu

```

public float[] solve_single_cost(int option, float costofdelay )
{
for (int j = 0; j < f.RL.Length; j++)
{
for (int i = 0; i < 1050; i++)
    { f.RL[j].usage[i] = 0;          }
}
for (int i = 0; i < f.Activities.Count; i++)
{ Activity ar = (Activity)f.Activities[i];
  ar.Starttime = 0;
  ar.scheduled = false;
  ar.Let = 0;
  ar.Lst = 0;
  ar.Eet = 0;
  ar.Lst = 0; }
int projectduration = 0;
int periodnum = f.Periodsext.Count;
periodex pm = (periodex)f.Periodsext[f.Periodsext.Count - 1];
int kx = pm.end;
while (kx < 1000)
{
for (int ptx = 0; ptx < periodnum; ptx++)
{
periodex pcur = (periodex)f.Periodsext[ptx];
periodex pnew = new periodex();
pnew.start = kx;
pnew.end = pcur.end - pcur.start + kx;
pnew.resourcecost = new float[f.resourcecount];
pnew.resourcecapacity = new float[f.resourcecount];
for (int r = 0; r < pcur.resourcecapacity.Length; r++)
{
pnew.resourcecapacity[r] = pcur.resourcecapacity[r];
pnew.resourcecost[r] = pcur.resourcecost[r];
}
for (int l = pnew.start; l < pnew.end; l++)
{
for (int h = 0; h < f.resourcecount; h++)
{
f.RL[h].cost[l] = pnew.resourcecost[h];
}
}
kx = pnew.end;
f.Periodsext.Add(pnew);
}
}
CPM cpm = new CPM();
ArrayList getx = new ArrayList();
getx = cpm.Forward(f.Activities);
getx = cpm.Backward(getx);
ArrayList unscheduled = new ArrayList();
ArrayList scheduled = new ArrayList();
ArrayList S = new ArrayList();

for (int i = 0; i < getx.Count; i++)
{
unscheduled.Add(getx[i]);
} int T = 0;
while (unscheduled.Count > 0)
{
S.Clear();
for (int j = 0; j < unscheduled.Count; j++)

```



```

        {
Activity ap = (Activity)unscheduled[j];
Boolean state = false;
Boolean capacitylimit = true;
for (int cc = 0; cc < ap.Duration; cc++)
{
    for (int tb = 0; tb < f.resourcecount; tb++)
    {
if (f.RL[tb].capacity[T + cc] < ap.ResourceRequirementex[tb] +
f.RL[tb].usage[T + cc])
{    capacitylimit = false;}
    }
}
if (capacitylimit == true)
{
    state = true;
    for (int k = 0; k < ap.Predecessors.Count; k++)
    {
Activity p = (Activity)ap.Predecessors[k];
if (p.scheduled == true && (p.Starttime + p.Duration <= T))
{    state = true;
}
else
{    state = false;
break;
}
}
if (state == true)
{
S.Add(ap);
} }
Selection sl = new Selection();
int index = sl.Select(S, option);
if (index > -1)
{
Activity selected = (Activity)S[index];
selected.Starttime = T;
selected.scheduled = true;
for (int h = 0; h < unscheduled.Count; h++)
{
    Activity ar = (Activity)unscheduled[h];
    if (String.Equals(ar.id, selected.id)
    {
unscheduled.RemoveAt(h);
break;    }
}
scheduled.Add(selected);
for (int tt = T; tt < selected.Duration + T; tt++)
{
    for (int tb = 0; tb < f.resourcecount; tb++)
    {f.RL[tb].usage[tt] = f.RL[tb].usage[tt] +
selected.ResourceRequirementex[tb];
}
}
}
if (S.Count < 1) T = T + 1;
}
for (int yy = 0; yy < scheduled.Count; yy++)
{
    Activity an = (Activity)scheduled[yy];
    if (an.Starttime + an.Duration > projectduration)

```

```

    {
projectduration = an.Starttime + an.Duration;
    } }

    ArrayList rescheduled = new ArrayList();
    for (int i = 0; i < getx.Count; i++)
    {
        rescheduled.Add(scheduled[i]);
    }
    for (int ii = 0; ii < rescheduled.Count; ii++)
    {
        Boolean isrescheduled = false;
        for (int yy = 0; yy < rescheduled.Count; yy++)
        {
Activity an = (Activity)rescheduled[yy];
if (an.Starttime + an.Duration > projectduration)
{
    projectduration = an.Starttime + an.Duration;
}
        Activity ax = (Activity)rescheduled[ii];
        int startperiod = -1;
        int endperiod = -1;
        for (int pp = 0; pp < f.Periodsext.Count; pp++)
        {
periodex pr = (periodex)f.Periodsext[pp];
if ((ax.Starttime >= pr.start) && (ax.Starttime < pr.end))
{
    startperiod = pp;

}
if ((ax.Starttime + ax.Duration >= pr.start) && (ax.Starttime + ax.Duration
<= pr.end))
{
    endperiod = pp;
}
if (startperiod > -1 & endperiod > -1)
{
    break;
} } if (startperiod > 0)
{
periodex pf = (periodex)f.Periodsext[startperiod];
periodex pff = (periodex)f.Periodsext[startperiod - 1];
float tot_pf = 0, tot_pff = 0;
for (int tl = 0; tl < f.resourcecount; tl++)
{
    tot_pf = tot_pf + ax.ResourceRequirementex[tl] * pf.resourcecost[tl];
    tot_pff = tot_pff + ax.ResourceRequirementex[tl] * pff.resourcecost[tl];
}
if (tot_pf > tot_pff)
{
    Boolean state = true;
    while (state == true)
    {for (int gr = 0; gr < f.resourcecount; gr++)
{
        if (f.RL[gr].capacity[ax.Starttime - 1] < ax.ResourceRequirementex[gr] +
f.RL[gr].usage[ax.Starttime - 1])
        {
            state = false;
            break;
        }
}
}
if (state == true)
{
    for (int k = 0; k < ax.Predecessors.Count; k++)

```

```

    {
        Activity p = (Activity)ax.Predecessors[k];
        if (p.scheduled == true && (p.Starttime + p.Duration < ax.Starttime - 1))
        {
            state = true;
        }
        else {
            state = false;
            break;
        }
    }
}
if (state == true)
{
    float oldcost = 0;
    float newcost = 0;
    for (int kk = ax.Starttime; kk < ax.Starttime + ax.Duration; kk++)
    {
        for (int gr = 0; gr < f.resourcecount; gr++)
        {
            oldcost = oldcost + ax.ResourceRequirementex[gr] * f.RL[gr].cost[kk];
            newcost = newcost + ax.ResourceRequirementex[gr] * f.RL[gr].cost[kk - 1];
        }
        if (oldcost > newcost)
        {
            for (int gr = 0; gr < f.resourcecount; gr++)
            {
                f.RL[gr].usage[ax.Starttime - 1] = f.RL[gr].usage[ax.Starttime - 1] +
                ax.ResourceRequirementex[gr];
                f.RL[gr].usage[ax.Starttime + ax.Duration-1] = f.RL[gr].usage[ax.Starttime
                + ax.Duration-1] - ax.ResourceRequirementex[gr];
            }
            ax.Starttime = ax.Starttime - 1;
            isrescheduled = true;
        }
        else { state = false; }
    }
}
}
    if (isrescheduled)
    {
        for (int pp = 0; pp < f.Periodsext.Count; pp++)
        {
            periodex pg = (periodex)f.Periodsext[pp];
            if ((ax.Starttime > pg.start) && (ax.Starttime < pg.end))
            {
                startperiod = pp;
            }
            if ((ax.Starttime + ax.Duration > pg.start) && (ax.Starttime +
            ax.Duration < pg.end))
            {
                endperiod = pp;
            }
            if (startperiod > 0 & endperiod > 0)
            {
                break;
            }
        }
    }
    periodex px = (periodex)f.Periodsext[startperiod];
    periodex pxx = (periodex)f.Periodsext[startperiod + 1];
    float tot_px = 0, tot_pxx = 0;
    for (int tl = 0; tl < f.resourcecount; tl++)

```

```

        {
            tot_px = tot_px + ax.ResourceRequirementex[t1] * px.resourcecost[t1];
            tot_pxx = tot_pxx + ax.ResourceRequirementex[t1] * pxx.resourcecost[t1];
        }
        if (tot_px > tot_pxx)
        {
            Boolean available = true;
            for (int jj = 0; jj < ax.Successors.Count; jj++)
            {
                Activity suc;
                suc = (Activity)ax.Successors[jj];
                if (suc.Starttime <= ax.Starttime + ax.Duration + 1)
                {
                    available = false;
                    break;
                }
            }
            Boolean resok = true;
            if ((available == true))
            {
                for (int gr = 0; gr < f.resourcecount; gr++)
                {
                    if ((ax.Duration + ax.Starttime + 1 <= projectduration) &&
                        (f.RL[gr].capacity[ax.Starttime + ax.Duration + 1] >=
                        f.RL[gr].usage[ax.Starttime + ax.Duration + 1] +
                        ax.ResourceRequirementex[gr]))
                    {
                        resok = true;
                    }
                    else
                    {
                        resok = false;
                        break;
                    }
                }
                if (resok == true)
                {
                    for (int gr = 0; gr < f.resourcecount; gr++)
                    {
                        f.RL[gr].usage[ax.Starttime] = f.RL[gr].usage[ax.Starttime] -
                        ax.ResourceRequirementex[gr];
                        f.RL[gr].usage[ax.Starttime + ax.Duration+1] = f.RL[gr].usage[ax.Starttime
                        + ax.Duration+1] + ax.ResourceRequirementex[gr];
                    }
                    ax.Starttime = ax.Starttime + 1;
                    isrescheduled = true;
                }
            }
        }
        tot_px = 0; tot_pxx = 0;
        for (int t1 = 0; t1 < f.resourcecount; t1++)
        {
            tot_px = tot_px + ax.ResourceRequirementex[t1] * px.resourcecost[t1];
            tot_pxx = tot_pxx + ax.ResourceRequirementex[t1] * pxx.resourcecost[t1];
        }
        if (tot_px > tot_pxx)
        {
            ArrayList aa = new ArrayList();

            aa.Add(ax);
            int rr = 0;
            while (aa.Count > rr)
            {

```

```

        Activity sucy;
        sucy = (Activity)aa[rr];
        for (int ip = 0; ip < sucy.Successors.Count; ip++)
        {
Activity sucrc;
sucrc = (Activity)sucy.Successors[ip];
if (sucrc.Starttime <= sucy.Starttime + sucy.Duration + 1)
{
    aa.Add(sucrc);
}
        }
        rr++;
    }
    float oldcost = 0;
    float newcost = 0;
    for (int tt = 0; tt < aa.Count; tt++)
    {
        Activity sucy;
        sucy = (Activity)aa[tt];
        for (int kk = sucy.Starttime; kk < sucy.Starttime + sucy.Duration; kk++)
        {
for (int gr = 0; gr < f.resourcecount; gr++)
{
    oldcost = oldcost + sucy.ResourceRequirementex[gr] * f.RL[gr].cost[kk];
    newcost = newcost + sucy.ResourceRequirementex[gr] * f.RL[gr].cost[kk +
1];
}
        }
        newcost = newcost + costofdelay;
        if (newcost < oldcost)
        {

            for (int h = 0; h < aa.Count; h++)
            {
                Boolean resok = true;
Activity ak = (Activity)aa[h];
for (int gr = 0; gr < f.resourcecount; gr++)
{
                if (f.R.capacity[ak.Starttime + ak.Duration + 1] >= f.R.usage[ak.Starttime
+ ak.Duration + 1] + ak.ResourceRequirementex[gr])
                {
                    resok = true;
                }
                else
                {
                    resok = false;
                    break;
                }
            }
            if (resok == true)
            {
                for (int gr = 0; gr < f.resourcecount; gr++)
                {
f.RL[gr].usage[ak.Starttime] = f.RL[gr].usage[ak.Starttime] -
ak.ResourceRequirementex[gr];
f.RL[gr].usage[ak.Starttime + ak.Duration+1] = f.RL[gr].usage[ak.Starttime
+ ak.Duration+1] + ak.ResourceRequirementex[gr];
                }
                ak.Starttime = ak.Starttime + 1;
                isrescheduled = true;
            }
        }
    }
}
        }
        if (isrescheduled == true)
        {
ii--;
        }
    }
}

```

```
for (int jx = f.Periodsext.Count - 1; jx >= periodnum; jx--)
{
    f.Periodsext.RemoveAt(jx);
}
float costy = 0;
float[] costx=new float[f.resourcecount] ;
for (int rs = 0; rs < projectduration; rs++)
{
    for (int cc = 0; cc < f.resourcecount; cc++)
    {
        costy = costy + f.RL[cc].usage[rs] * f.RL[cc].cost[rs];
    }
}
float[] ret = new float[2];
ret[0] = projectduration;
ret[1] = costy;
return ret;
}
```

Ek 2 – Uygulama Projesi Faaliyet Tanımları

Faaliyet Kodu	Faaliyet Tanımı	Faaliyet Süresi (gün)
A1000	Hafriyat İşleri	36
A1010	Kazık İşleri	50
A1020	Temel Beton	34
A1030	Kolon Beton	37
A1040	Perde Beton	37
A1050	Döşeme Beton	42
A1060	Merdiven Beton	30
A1120	Şap İşleri	36
A1130	Seramik Kaplama İşleri	48
A1140	Granit Kaplama İşleri	42
A1070	Tuğla Duvar İşleri	38
A1090	Kara Sıva İşleri	38
A1100	Alçı Sıva İşleri	42
A1110	Boya İşleri	48
A1080	Gazblok Duvar İşleri	48
A1150	Sıva İşleri	30
A1160	Boya İşleri	30
A1170	Armstrong Asma Tavan	26
A1190	Kablo Tavalalarının Çekilmesi	25
A1200	Panoların Yerleştirilmesi	28
A1210	Switch, Soket ve	22
A1220	CCTV	24
A1230	Güvenlik Sistemleri	14
A1240	Devreye Alma İşleri -Mekanik	20
A1180	Aydınlatma İşleri	7
A1250	Isıtma ve ITP	19
A1260	Soğutma	22
A1270	Havalandırma	19
A1280	Temiz Su Temini	25
A1290	Kanalizasyon ve Drenaj	24
A1310	Devreye Alma ve Testler	25
A1300	Yagmurlama Sistemleri	12

Ek 3 – Uygulama Projesi Faaliyet Öncüllük İlişkileri Süre ve Kaynak Gereksinimleri

Faaliyet Kodu	Ardıl Faaliyetler	Hareketli Platform (Makina Saat)	Hazır Beton (m3)	Çimento (kg.)	İşçilik (Adam Saat)
A1000	A1010 A1020 A1030 A1050	2	0	0	200
A1010	A1040	4	6	0	220
A1020	A1030 A1040 A1050	0	12	0	200
A1030	A1130	0	12	0	180
A1040	A1060 A1080	0	12	0	180
A1050	A1120 A1070	0	12	0	220
A1060	A1120	0	12	0	180
A1120	A1130 A1140	2	12	60	100
A1130		2	0	40	140
A1140		3	0	120	120
A1070	A1090 A1080 A1190	5	0	180	100
A1090	A1100 A1170	5	0	120	100
A1100	A1110 A1150	10	0	20	120
A1110		8	0	0	140
A1080		8	0	125	140
A1150	A1160	8	0	150	100
A1160	A1170	6	0	0	100
A1170	A1180 A1250 A1280	10	0	0	100
A1190	A1200 A1210 A1220 A1230	10	0	0	100
A1200	A1240	10	0	0	100
A1210	A1240	10	0	0	100
A1220	A1240 A1300	8	0	0	100
A1230	A1240	8	0	0	100
A1240		8	0	0	100
A1180	A1310	8	0	0	80
A1250	A1260 A1270	6	0	0	120
A1260	A1310	6	0	0	100
A1270	A1310	5	0	0	100
A1280	A1290	2	0	12	100
A1290	A1310	0	0	50	120
A1310		0	0	0	140
A1300	A1310	8	0	0	70

Ek 4 – Uygulama Projesi CPM ve Önerilen Algoritma İş Programı

CPM

Faaliyet	Başlangıç	Bitiş
A1000	0	36
A1010	36	86
A1020	36	70
A1030	70	107
A1040	86	123
A1050	70	112
A1060	123	153
A1120	153	189
A1130	189	237
A1140	189	231
A1070	112	150
A1090	150	188
A1100	188	230
A1110	230	278
A1080	150	198
A1150	230	260
A1160	260	290
A1170	290	316
A1190	150	175
A1200	175	203
A1210	175	197
A1220	175	199
A1230	175	189
A1240	203	223
A1180	316	323
A1250	316	335
A1260	335	364
A1270	335	367
A1280	316	341
A1290	341	365
A1310	367	392
A1300	199	211

Önerilen Algoritma

Faaliyet	Başlangıç	Bitiş
A1000	0	36
A1010	36	86
A1020	36	70
A1030	364	401
A1040	86	123
A1050	70	112
A1060	123	153
A1120	153	189
A1130	401	449
A1140	342	384
A1070	112	150
A1090	150	188
A1100	217	259
A1110	284	332
A1080	437	485
A1150	259	289
A1160	289	319
A1170	319	345
A1190	150	175
A1200	189	217
A1210	261	283
A1220	188	212
A1230	270	284
A1240	469	489
A1180	414	421
A1250	345	364
A1260	385	414
A1270	364	396
A1280	360	385
A1290	385	409
A1310	421	446
A1300	246	258