



**TÜBİTAK  
ARAŞTIRMA PROJESİ  
SONUÇ RAPORU**

**İnşaat Projelerinde Kaynak Dengeleme Problemi için  
Yüksek Performanslı Kesin ve Üst-Sezgisel Yöntemler  
Geliştirilmesi**

**Program Kodu: 1001**

**Proje No: 111M140**

Proje Yürütücüsü:  
**Doç. Dr. Rifat SÖNMEZ**

Araştırmacılar

Y. Doç. Dr. Tankut ATAN  
Alphan NURTUĞ

Bursiyerler

Mahdi Abbasi IRANAGH  
Emad REZVANKHAH  
Hüseyin YENİOCAK



## ÖNSÖZ

111M140 Nolu bu proje TÜBİTAK tarafından desteklenmiştir.



## ÖZET

Bu projede kaynak dengeleme probleminin (KDP) çözümünde ilerlenme sağlanabilmesi için yüksek performanslı kesin ve üst-sezgisel yöntemler geliştirilmiştir. Yapılan kapsamlı testler proje kapsamında geliştirilen dal ve sınır algoritmasının literatürdeki mevcut dal ve sınır algoritmalarının performansının önemli ölçüde önüne geçtiğini teyit etmektedir. Dal ve sınır algoritması geliştirilirken KDP için yeni bir alt sınır hesap yöntemi bulunmuş, bu sayede KDP'nin dal ve sınır algoritması ile çözümünde önemli bir gelişme kaydedilmiştir. Geliştirilen dal ve sınır algoritması ile inşaat projeleri için pratik açıdan önemli olan azami kaynak kullanımı ve atıl kaynak günü ölçütü için 30 faaliyet ve dört kaynağa kadar olan problemler literatürde ilk kez optimal olarak çözülmüştür.

Proje kapsamında KDP'nin çözümünde güncel üst-sezgisel yöntemlerin iyileştirilmesi için tavlama benzetimli bir memetik algoritma (TBMA) geliştirilmiştir. TBMA kaynak kullanımının verimsiz olduğu kısımların azaltılması için memetik algoritmaların yerel taramalar için bireysel öğrenme özellikleri kullanmakta, çözüm uzayı tarama işlemini kontrol edilmesi ve özellikle tarama işleminin sonlarına doğru iyi çözümlerin muhafaza edilip, çözümlerin optimale yakınsaması amacıyla tavlama benzetimi yöntemini kullanmaktadır. TBMA'nın performansı mevcut 12 farklı sezgisel ve üst-sezgisel metodun performansı ile 300 faaliyet ve dört kaynağa kadar olan 1920 problem kullanılarak üç farklı ölçüt için kıyaslanmıştır. Performans test sonuçları TBMA'nın KDP'nin çözümünde performansının mevcut sezgisel ve üst-sezgisel yöntemlere çok daha iyi olduğu teyit etmektedir. Proje kapsamında geliştirilen tavlama benzetimli memetik üst-sezgisel algoritmanın yaygın etkisinin artırılması için Microsoft Project 2013 yazılımına bir eklenti olarak entegre edilmiş ve geliştirilen eklenti 522 yapım faaliyeti ve 18 işgücü kaynağını içeren gerçek bir endüstriyel tesis inşaat projesi ile test edilmiştir. Proje kapsamında yapılan çalışmaların sonuçları bildiriler, makaleler ve proje için oluşturulan bir web sitesi yoluyla akademik dünyaya sunulmuştur.



## ABSTRACT

In this project, high performance exact and meta-heuristic algorithms are developed for the resource leveling problem (RLP). The results of extensive tests indicate that the branch and bound algorithm developed in this project outperforms existing branch and bound algorithms significantly for the resource leveling problem. A new lower bound calculation method was presented for the RLP which presents an advancement for improving the existing branch and bound methods for the RLP. Problem instances including up to 30 activities and four resources have been solved to optimality for the first time in the literature for the maximum resource demand and resource idle day metric, which is very important in practice for the construction projects.

A memetic algorithm with simulated annealing (MASA) has been developed in the project to improve existing heuristic and meta-heuristic methods for the RLP. MASA uses the self local learning capabilities of memetic algorithms to improve the inefficient parts of the resource profiles, and uses simulated annealing to control the search process and to improve convergence while keeping the good solutions especially during the later search stages. The performance of MASA is compared with performance of 12 existing heuristic and meta-heuristic methods using 1920 test instances including up to 300 activities and 4 resources. Test results reveal that MASA outperforms existing methods significantly. MASA was developed as a Microsoft Project 2013 add-in to enhance practical effectiveness of the project. The MASA add-in was tested with a real industrial construction project which included 522 activities and 18 resources. The results of the project was presented to the academic world by proceedings, articles and a web site.

## İçindekiler

Önsöz.....	ii
Özet .....	iii
Abstract .....	iv
1.Giriş .....	1
2.Literatür Özeti.....	2
2.1 Kesin Yöntemlerle İlgili Literatür Çalışmaları.....	2
2.2 Sezgisel ve Üst-Sezgisel Yöntemlerle İlgili Literatür Çalışmaları.....	3
3. Doğrusal Tamsayılı Modeller Ve Ms Excel-GAMS/CPLEX Uygulaması.....	10
3.1 Doğrusal Tamsayılı Eniyileme (Optimizasyon) Modeli.....	10
3.1.1. Kümeler.....	10
3.1.2. Parametreler.....	10
3.1.3. Değişkenler.....	10
3.1.4. Modelleme.....	11
3.2 MS Excel ve GAMS/CPLEX Uygulaması.....	16
4. Dal Ve Sınır Algoritması.....	20
4.1. Dal ve Sınır Algoritmasının Temel Bölümleri.....	20
4.2. Kritik Yol Yöntemi İş Programlaması.....	21
4.2.1. İleri İşlem.....	24
4.2.2. Geri İşlem.....	24
4.3. Süre ilişkili programlama yöntemi (Temporal scheduling).....	25
4.4. Kesin Çözüm İçin Geliştirilen Dal ve Sınır Algoritması.....	26
4.4.1. Algoritmada Kullanılan Bazı Tanımlar.....	26
4.4.2. Algoritmada Kullanılan Bazı Fonksiyonlar.....	28
4.4.3. Dal ve Sınır Algoritmasının Şematik Anlatımı.....	30
4.4.4. Dal ve Sınır Algoritmasının Performansının Artırılması için İzlenen Stratejiler.....	36
4.4.5. Sezgisel Yöntemle İlk Üst Sınır Değeri Belirlemesi.....	38
4.4.6. Alt Sınır Değeri Hesaplamasında Yapılan İyileştirmeler.....	40
4.4.7. Sabit ve Dinamik Budak Cinsiyeti Seçimi.....	42
4.4.8. Budakların Birbirlerine Bağlanmasında Kullanılan Değere Göre Adresleme Yöntemi.....	44
5. Dal Ve Sınır Algoritmasının Performansının Belirlenmesi İçin Yapılan Testler.....	46
5.1. Testlerde Kullanılan Ölçütler.....	47
5.2. Gather vd. (2010)'daki Problemlerin Test Sonuçları.....	47

5.3. Reick vd. (2012) Problemleri ve Karelerin Toplamı ve Ortalamayı Aşan Kaynaklar Toplamı Ölçütleri için Yapılan Testler .....	49
5.4. Mutlu (2010) Problem Setleriyle ve Azami Kaynak Kullanımı ve Atıl Kaynak Günü (MRD+RID) Ölçütü ile Yapılan Testler.....	51
5.5. Rieck vd. (2012) Problemleri ve Azami Kaynak Kullanımı ve Atıl Kaynak Günü (MRD+RID) Ölçütü ile Yapılan Testler.....	52
5.6. Dal ve Sınır Algoritması ve Karma-Tamsayılı Modelleme Yöntemi ile Yapılan Testlerin Sonuçlarının Özeti.....	54
6. Üst Sezgisel Kaynak Dengeleme Algoritmaları.....	54
6.1. Genetik Algoritmanın Yapısı.....	54
6.1.1. KYY'ye Göre İş Programı Hazırlaması .....	56
6.1.2. Başlangıç Popülasyonunun Oluşturulması.....	56
6.1.3. Genetik İşlemlerin Uygulanması.....	56
6.2. Memetik Algoritma.....	57
6.2.1. TBMA'da Kromozomların Yapısı.....	57
6.2.2. Sezgisel İyileştirme.....	59
6.2.3. Tavlama Benzetimi.....	61
6.2.4. Çaprazlama, Mutasyon, ve Tavlama Benzetimi.....	62
7. Üst Sezgisel Kaynak Dengeleme Algoritmalarının Performansının Belirlenmesi İçin Yapılan Testler.....	64
7.1. Küçük Ölçekli Problem Setleriyle Yapılan Testler.....	64
7.2. Kolish ve Sprecher (1997) ile Debels ve Vanhoucke (2007) Problem Setleriyle Yapılan Testler.....	65
7.3. TBMA' nın Microsoft Project ve Primavera ile Karşılaştırılması için Yapılan Testler.....	69
7.4. TBMA' nın AGA ile Karşılaştırılması için Yapılan Testler.....	71
8. Geliştirilen Üst Sezgiselin Ms Project'e Entegrasyonu.....	72
8.1. TBMA Microsoft Project Professional 2013 Eklentisi.....	73
8.2. TEPE İnşaat Planlama Bölümü'nün görüşleri.....	77
8.3. TBMA Microsoft Project Eklentisinin Gerçek bir İnşaat Projesi ile Denenmesi.....	77
9. Sonuçlar Ve Öneriler.....	81
Kaynaklar.....	84
Akademik Çıktılar ve Yayınlar .....	90

## Tablolar Listesi

Tablo.1 Kesin yöntemlerle ilgili literatür çalışmaları.....	6
Tablo 2. Sezgisel ve üst-sezgisel yöntemlerle ilgili literatür çalışmaları.....	7
Tablo 3. Gather vd. (2010) problem setleriyle yapılan karelerin toplamı ölçütüne göre test.....	48
Tablo 4. Rieck vd. (2012) problem setleriyle ve karelerin toplamı ölçütü ile yapılan testlerin sonuçları.....	50
Tablo 5. Rieck vd. (2012) problem setleriyle ve ortalamayı aşan kaynaklar toplamı ölçütü ile yapılan testlerin sonuçları .....	50
Tablo 6. Mutlu (2010) Problem Setleriyle ve Azami Kaynak Kullanımı ve Atıl Kaynak Günü (MRD+RID) Ölçütü ile Yapılan Testlerin Sonuçları .....	52
Tablo 7. MRD+RID ölçütü için B&B ve DT yöntemlerinin sonuçları .....	53
Tablo 8. Küçük ölçekli problem seti ve KT ölçütü ile yapılan testlerin sonuçları.....	66
Tablo 9. Küçük ölçekli problem seti ve OKMSM ölçütü ile yapılan testlerin sonuçları.....	67
Tablo 10. Küçük ölçekli problem seti ve MRD+RID ölçütü ile yapılan testlerin sonuçları.....	68
Tablo 11. 1920 problem ile yapılan testlerin sonuçları.....	69
Tablo 12. TBMA'nın MS Project ve Primavera ile karşılaştırılması.....	70
Tablo 13. TBMA'nın AGA(Ponz-Tienda ve vd. 2013). ile karşılaştırılması.....	72
Tablo 14. İnşaat Projesine ait Kaynaklar.....	77

## Şekiller Listesi

Şekil 1. Problem bilgilerinin Excel'de girilmesi.....	17
Şekil 2. KYY iş programlama sonuçları ve çözüm için kullanıcı tercihleri.....	18
Şekil 3. MRD+RID için GAMS programının çalışma sonuçları.....	18
Şekil 4. El Rayes ve Jun (2009)'a ait problemin MRD+RID toplamını en aza indiren ölçüte göre çok kaynaklı (4 kaynak) çözümü.....	19
Şekil 5. Kesin çözüm algoritmasının temel içeriği.....	21
Şekil 6. Problem çözümünde kullanılacak veri tablosunun biçimi.....	22
Şekil 7. Algoritmanın dosya okuma işleminden sonraki veri tablosunun görünümü.....	23
Şekil 8. Veri tablosunun KYY iş programlama işleminden önceki görünümü .....	23
Şekil 9. Veri tablosunun KYY'ye göre programlama sonucundaki hali .....	25
Şekil 10. Süre ilişkili programlama yöntemine göre hazırlanmış bir şebeke diyagramı.....	25
Şekil 11. Easa (1989)'a ait problemin dal ve sınır algoritmasıyla mutlak sapma miktarının en aza indirilmesi amaç fonksiyonuna göre kesin çözümünün bulunuşu .....	31
Şekil 12. Easa (1989)'a ait problemin KYY iş programı bilgileri.....	32

Şekil 13. Dal ve sınır algoritması işlemi süresince aday anne budak zinciri, en iyi budak ve en iyi yaprak elemanları içerisindeki değişim.....	34
Şekil 14a. Dal budak şemasına ait ara kaynak kullanım profilleri (Kısım 1) .....	35
Şekil 14b. Dal budak şemasına ait ara kaynak kullanım profilleri (Kısım 2) .....	36
Şekil 15. İşlem ve hafıza gerektiren budakların alt sınır değerine göre dağılımı.....	37
Şekil 16. Üst sınır değeri hesaplaması için en iyi aktivite sıralama kriterinin seçimi .....	40
Şekil 17. Bir budağın alt sınır değeri hesabı aşamasında ara kaynak kullanım profilinin oluşumu.....	41
Şekil 18. Değere göre adresleme yöntemi.....	45
Şekil 19. İki kromozomun çaprazlamasıyla yeni kromozomlar elde edilmesi.....	55
Şekil 20. Kromozomu oluşturan genlerden birinin mutasyona uğratılarak yeni bir kromozom elde edilmesi.....	55
Şekil 21. Örnek problem.....	59
Şekil 22. Örnek probleme ait bir kromozom.....	59
Şekil 23. Kromozomun deşifre edilmesi.....	60
Şekil 24. İyileştirilmiş program.....	61
Şekil 25. İyileşmiş Programın Gen Kodu.....	61
Şekil 26. TBMA'nın akış diyagramı.....	63
Şekil 27. TBMA'a ait eklentinin kaynak dengeleme seçenekleri.....	73
Şekil 28. MRD+RID ölçütüne ait veri penceresi.....	74
Şekil 29. El-Rayes ve Jun (2009)' a ait problemin MRD+RID ölçütüne göre TBMA eklentisi ile dengelenmiş kaynak profili.....	75
Şekil 30. El-Rayes ve Jun (2009)' a ait problemin erken başlangıç kaynak profili.....	75
Şekil 31. El-Rayes ve Jun (2009)'a ait problemin OKMSM ölçütüne göre TBMA eklentisi ile dengelenmiş kaynak profili.....	76
Şekil 32. El-Rayes ve Jun (2009)'a ait problemin KT ölçütüne göre TBMA eklentisi ile dengelenmiş kaynak profili.....	76
Şekil 33. Fabrikasyon tank montaj yardımcısı için erken başlangıç kaynak profili.....	78
Şekil 34. Fabrikasyon tank montaj yardımcısı için TBMA eklentisi ile dengelenmiş kaynak profili.....	79
Şekil 35. Fabrikasyon tank kaynak ustası için erken başlangıç kaynak profili.....	79
Şekil 36. Fabrikasyon tank kaynak ustası için TBMA eklentisi ile dengelenmiş kaynak Profili.....	80
Şekil 37. Donatı ustası için erken başlangıç kaynak profili.....	80
Şekil 38. Donatı ustası için TBMA eklentisi ile dengelenmiş kaynak profili .....	81





## 1.GİRİŞ

Kaynak dengelenmesi problemi (KDP), kritik yol yöntemine göre hazırlanan proje bitiş süresi baz alınarak, kaynak çizelgesindeki dalgalanmaları en aza indirgeyen, faaliyet başlangıç ve bitiş zamanlarının bulunmasıdır. Kaynak dengelenmesi, inşaat yapım projelerinde kullanılan özellikle işgücü ve makine kaynaklarının verimli bir şekilde kullanılıp, projelerinin düşük maliyetle tamamlanması açısından çok önemlidir.

Kaynak çizelgesindeki dalgalanmalar eğer düzenlenmezse dört önemli problemin ortaya çıkmasına sebep olabilir. Bu problemler: (1) inşaat projesinde çalışacak personelin çok kısa süreli istihdam edilmesi, (2) kaliteli işgücünün (sürekli istihdam garanti edilemediğinden dolayı) cezbedilememesi, (3) öğrenme eğrisinde aksamaların gerçekleşmesi, ve (4) bazı personelin boş durması sebebiyle önemli verimlilik kayıplarının olmasıdır (El-Rayes ve Jun 2009). Bu sebeplerden ötürü kaynakların doğru planlanmaması proje maliyetlerinin artmasına sebep olmaktadır. Kaynak dengeleme probleminin bu önemine rağmen, inşaat projelerinin programlaması ve yönetiminde yaygın olarak tercih edilen Microsoft Project ve Primavera paket programlarında kullanılan yöntemler kaynak dengeleme problemi (KDP)'nin çözümünde yetersiz kalmaktadır

KDP'nin çözümü için literatürde yer alan yöntemler iki grupta toplanabilir. İlk grupta kesin yöntemler vardır ve bu yöntemlerin amacı KDP için optimal çözümün bulunmasıdır. Tamsayılı programlama, dinamik programlama ve dal ve sınır teknikleri KDP'nin çözümü için kullanılan kesin yöntemlerdendir. Kesin yöntemlerin kullanıldığı çalışmalar çok az sayıdadır ve genellikle küçük ölçekli problemlere yönelmiştir. KDP'nin çözümü için geliştirilmiş ikinci grup yöntemler ise sezgisel ve üst-sezgisel yöntemlerdir. Bu yöntemlerin amacı ise kısa süre içerisinde KDP için optimala yakın bir çözümün bulunmasıdır. Üst-sezgisel yöntemler arasında KDP için en yaygın olarak kullanılan teknik gen algoritmalarıdır. Gen algoritmaları dışında karınca kolonisi algoritması, kuş sürüsü algoritması, tavlama simülasyonu ve yapay sinir ağları da KDP'nin çözümü için önerilen üst-sezgisel yöntemler arasındadır. Sezgisel ve üst-sezgisel yöntemlerin kullanıldığı çalışmalarda genellikle optimal sonucu belli olmayan tek kaynaklı küçük problemler ele alınmıştır.

Bu projede, inşaat yapım projelerinde kaynak kullanımında ortaya çıkan dengesizlikleri azaltmaya yönelik yüksek performanslı kesin ve üst-sezgisel yöntemlerin geliştirilmesi hedeflenmiştir. KDP için geliştirilmesi hedeflenen kesin yöntem ile geliştirilecek olan üst-sezgisel yöntemin performansının test edilmesi ve literatürdeki mevcut kesin çözüm algoritmalarının performansının geliştirilmesi planlanmıştır. Geliştirilmesi planlan üst-sezgisel

yöntemle ise küçük, orta ve büyük ölçekli çoklu kısıtsız kaynaklı şebekeler için KDP'nin çözümünde mevcut sezgisel ve üst-sezgisel yöntemlerinin iyileştirilmesi hedeflenmiştir. Proje kapsamında geliştirilecek üst-sezgisel kaynak dengeleme algoritmasının Microsoft Project paket programı ile entegre edilmesi yoluyla projenin yaygın etkisinin artırılması amaçlanmıştır.

## 2.LİTERATÜR ÖZETİ

Kritik yol yöntemi (KYY) yaklaşık 60 yıldır inşaat projelerinin programlamasında ve kontrolünde kullanılmaktadır. Özellikle son yıllarda KYY'nin kullanımı inşaat sektöründe yaygınlaşmış ve birçok projenin programlamasında bir standart haline gelmiştir. Klasik KYY'ye göre hazırlanan iş programlarında kaynakların optimizasyonu dikkate alınmaması, birçok araştırmacının KDP'nin çözümüne yönelmesine sebep olmuştur. Bu konudaki literatür çalışmalarını kesin ve sezgisel yöntemler olarak iki ana grupta toplamak mümkündür.

### 2.1 Kesin Yöntemlerle İlgili Literatür Çalışmaları

KDP'nin kesin çözümü için literatürdeki ilk çalışmalar sayım yöntemi, tamsayılı programlama ve dinamik programlama tekniklerini önermiştir (Easa 1989, Karshenas ve Haber 1990, Bandelloni v.d. 1994, Younis ve Saad 1996, Mattila ve Abraham 1998, ve Son ve Mattila 2004). Ancak bu çalışmalarda genellikle tek kaynaklı çok küçük problemler kullanılmıştır ve önerilen tekniklerin hesaplama performansı dahil edilmemiştir. Neumann ve Zimmermann (2000) aktiviteleri ardaşık bir şekilde programlayarak, uygulanabilir çözüm kümesini azaltan bir dal ve sınır yöntemi önermiştir. Neumann ve Zimmermann (2000)'ın önerdiği yöntemin diğer kesin çözüm yöntemlerinden en önemli farkı bu yöntemle karelerin toplamı veya atıl kaynak günü ölçütünü kullanan farklı optimizasyon fonksiyonlarına sahip kaynak dengeleme problemlerini çözebilmesidir. Son ve Mattila (2004), aktivitelerin bölünmesine izin vererek bu yöntemle daha iyi sonuçlara ulaşılabildiğini göstermiştir. Gather v.d. (2010), dal ve sınır tekniğine yeni bir döküm yöntemi ekleyerek Kolisch v. d. (1999) tarafından önerilen rlp\_j10 ve rlp\_j20 test setleri, Franck v. d. (2001) tarafından oluşturulan ubo10 ve ubo20 test setleri ve son olarak Neumann ve Zimmermann (2000)'nin sunduğu rlp-15 test setlerini çözdürmüş ve bu testler için işlem sürelerini vermişlerdir. En çok 20 aktivite içeren bu örneklerin çözümü için işlem süresi olarak 10 saat gibi bir sürenin sınır olarak koyulduğu ve bu örneklerden bazılarının bu süre içerisinde çözülemediği görülmüştür. Mustafa Çağdaş Mutlu (2010) da dal ve sınır tekniğinde işlem gerektiren çözüm kümesini azaltan uygulamalarla algoritmanın daha hızlı ve daha verimli çalışmasını sağlamıştır. Rieck, Zimmermann ve Gather 2012'de

yayınlanan makalelerinde, KDP'nin kesin çözümü için doğrusal tamsayılı programlama yöntemiyle çeşitli modeller sunmuş ve bu modelleri geniş bir problem seti kullanarak test etmiştir. 10, 15, 20, 30 ve 50 aktivitelik tekli, üçlü ve beşli kaynaklı her biri 40 problemden oluşan setler olmak üzere toplam 600 problemi karelerin toplamı ve ortalamayı aşan kaynaklar toplamı ölçütüne göre çözdürülmüştür. Kesin yöntemlerle ilgili literatürde yer alan çalışmaların özeti Tablo 1'de verilmiştir.

## 2.2 Sezgisel ve Üst-Sezgisel Yöntemlerle İlgili Literatür Çalışmaları

Sezgisel yöntemler arasında en yaygın olarak kullanılan teknik Burgess ve Killebrew (1962)'un geliştirdiği öncelik sırasını baz alan tekniktir. Burgess ve Killebrew (1962)'den sonra önerilen sezgisel yöntemlerin (Harris 1990, Martinez ve Ioannou 1993, Neumann ve Zimmermann 1999) çoğu da öncelik sırası tekniğine dayanmaktadır. Woodworth v.d. (1975) çoklu kaynaklı çoklu projelerde iş programlaması yapmış ve bu programlarda kaynak dengeleme yaparak maximum günlük kaynak tüketimini azaltmayı hedefleyen bir sezgisel önermiştir.

Son yıllarda üst-sezgisel algoritmalar konusunda kaydedilen gelişmeler birçok araştırmacının KDP'nin çözümünde özellikle gen algoritmalarına (GA) yönelmesine sebep olmuştur. Oral v.d. (2003) ve El-Rayes ve Jun (2009) tek kaynaklı şebekelerde KDP'nin çözümü için gen algoritmalarının potansiyelini araştırmışlardır. Leu v.d. (2000) üç kaynaklı dokuz aktiviteli bir şebeke için, Hegazy (1999) ise altı kaynaklı yirmi aktiviteli bir şebeke için, gen algoritmalarının KDP'nin çözümündeki performansını incelemiştir. Zheng v.d. (2003) çoklu kaynaklı projeler için GA yöntemini kullanarak orta ve büyük ölçekli projeler için çözüm vaad eden algoritmalar üzerinde çalışmıştır. Gen algoritması kullanarak kaynak dengeleme ve kısıtlı kaynaklı iş programlama problemine (KKİPP) eş zamanlı bir yaklaşım Roca v.d (2008) tarafından sunulmuştur. Jun ve El-Rayes (2011), Roca v.d. (2008) gibi GA kullanarak KDP ve KKİPP'yi eş zamanlı bir yöntemle çözmeye çalışmıştır. Bettemir (2009), literatürde yer alan en büyüğü on üç aktiviteli altı kaynak dengeleme problemini kullanarak beş farklı yalın ve melez gen algoritmasının performansını kıyaslamıştır. Chan v.d. (1996), Leu v.d. (1999), ve Senouci v.d. (2004), GA tabanlı yöntemlerle iş programlama problemlerine bütünsel bir yaklaşım sergileyerek kaynak dengeleme, kısıtlı kaynaklı iş programlama ve zaman-maliyet ödünleşim problemlerini birlikte çözmeye çalışmışlardır.

El-Rayes ve Jun (2009) geliştirdikleri GA tabanlı algoritmada inşaat yapım projeleri için yeni bir ölçüt olan Azami Kaynak Kullanımı ve Atıl Kaynak Günü (MRD+RID) ölçütünü önermiştir.

El-Rayes ve Jun (2009) bu çalışmalarında önerdikleri MRD+RID ölçütününün kaynak eğrisinde atıl kaynak gününün en aza indirgenmesinde kaynak dengelenmesi için kullanılan diğer mevcut ölçütlere göre çok daha başarılı olduğunu göstermiştir.

Karınca kolonisi algoritması (Geng v.d. 2010), seri programlama jenerasyon şemasıyla birleştirilmiş karınca kolonisi optimizasyonu (Xiong ve Kuang 2006), kuş sürüsü algoritması (Qi v.d. 2007, Pang v.d. 2008, Guo v.d. 2009), tavlama simulasyonu (Son ve Skibniewski 1999) ve yapay sinir ağları (Savin v.d. 1997, Kartam ve Tongthong 1998) da KDP'nin çözümü için önerilen diğer üst-sezgisel yöntemler arasındadır. Bu gruptaki çalışmalarda kullanılan örneklerin hepsi tek kaynaklı olup, şebekelerin aktivite sayısı dört ile on üç arasında değişmektedir. Hegazy ve Erşahin (2001), toplam proje maliyetini azaltmak için tablolama tabanlı bir yöntem kullanarak iş programlama problemlerine genel bir yaklaşım sergilemiş, bu esnada algoritmalarında kaynak dengelemesine de yer vermişlerdir. Kaynak dengeleme problemini asgari moment yaklaşımıyla çözmeye çalışan araştırmacılardan (Hiyassat 2001 ve Christodoulou v.d 2010) Christodoulou v.d. daha iyi sonuç alabilmek için aktivitelerin sıkıştırılmasına ve gevşetilmesine izin vermiştir. Balestin v.d. (2007), popülasyon tabanlı bir yöntem için beş kaynaklı ve bin aktiviteye kadar problem içeren bir set denemişlerdir. Sezgisel ve üst-sezgisel yöntemlerin sonuçları incelenirken, kullanılan KDP problemlerinin optimal çözümünün olmaması yöntemlerin performansının değerlendirmesini güçleştirmektedir.

GA'lar önemli bir üst-sezgisel optimizasyon algoritmaları olmalarına rağmen, bazı problem için tepe tırmanma ve ince ayar konusunda çok başarılı olmayabilirler. Bu konulardaki yeteneklerinin sınırlı olması GA'ların optimal veya optimala yakın değerlere yakınsamasını bazı durumlarda zorlaştırmaktadır. Bu sebeple araştırmacılar GA'ların tepe tırmanma ve ince ayar yeteneklerinin geliştirilmesi için tavlama benzetimi yöntemi ile melezlenmesini önermektedir (Wang ve Zheng 2001, Chen v.d. 2005, Hwang ve He 2006, Han ve Sun 2006). Önder Halis Bettemir'in (2009) doktora tez çalışmasında, zaman-maliyet ödünleşim, kaynak dengeleme, ve kısıtlı kaynaklı iş programlaması problemlerinin çözümünde GA'ların tavlama benzetimi yöntemi ile melezlenmesinin, sonuçların geliştirilmesinde katkı sağladığı gözlemlenmiştir. Doulabi v.d. (2011) genetik algoritmaya onarım mekanizması ekleyerek GA'nın optimal değere yakınsamasını kolaylaştırmış, bu şekilde işlem süresini azaltmaya çalışmıştır. Alsayegh ve Hariga (2012) aktivite sürelerinin bölünebildiği ve bölünme maliyetlerinin de dahil edildiği KDP için kuş sürüsü (PSO) ve tavlama benzetimi algoritmalarının birleştirildiği bir melez algoritma önermiştir. KDP konusundaki en güncel çalışmada ise Ponz-Tienda v.d. (2013) adaptasyonlu bir genetik algoritma (AGA) geliştirmiş



ve bu algoritmayı, Sprecher, ve Drexl'in oluşturduğu (Kolisch v.d. 1995, Kolisch ve Sprecher, 1997) 480 adet 30 faaliyetli, 480 adet 60 faaliyetli, ve 480 adet 120 faaliyetli problemlerden meydana gelen toplam 1440 problemi kullanarak, karelerin toplamı ölçütü için test etmiştir. Sezgisel ve üst sezgisel yöntemlerle ilgili literatür çalışmaları Tablo 2' de özetlenmiştir.

İnşaat sektöründe proje programlaması ve yönetiminde en yaygın olarak kullanılan Microsoft Project ve Primavera paket programları (Liberatore v.d., 2001) kaynak dengeleme problemlerinin çözümünde öncelik sıralaması yöntemini kullanmaktadır. Kolisch (1999) çeşitli iş programlama yazılımlarının kullandığı öncelik sırasını baz alan sezgisel tekniklerin kaynak atanması konusundaki sınırlarını sunmuştur. Kastor ve Sirakoulis (2009) Primavera P6.0 ve Microsoft Project 2007'in KDP'nin çözümündeki yetersizliğini gerçek iki inşaat projesine ait verileri kullanarak göstermiştir.

KDP ile ilgili mevcut literatürde önemli boşluklar vardır. Araştırmaların çoğu tek kaynaklı küçük ölçekli şebekeler için KDP'nin çözümüne yönelmiştir. Çoklu kaynakların ele alındığı çalışmalarda ise genellikle faaliyet sayısı 20'yi geçmemektedir. Ayrıca sezgisel ve üst-sezgisel yöntemlerin performansının değerlendirilebilmesi için kritik yol yöntemine ait optimal çözümleri olan problemler (özellikle MRD+RID ölçütü için) yoktur. Yine mevcut iş programlama yazılımları KDP çözümünde son derece yetersiz olduğundan, bu konuda yapılacak iyileştirmeler özellikle inşaat sektörü açısından önem arz etmektedir.

Tablo 1. Kesin yöntemlerle ilgili literatür çalışmaları

Yıl	Yazar	Yöntem	Problem	Notlar
1989	Easa	Doğrusal Tamsayılı Programlama	Kaynak Dengeleme Problemi (KDP)	Ortalama kaynak kullanım düzeyinden sapmalar ve ardışık günlerin kaynak kullanımındaki sapmalar en aza indirilmiştir. KDP için kesin yöntemler önermiştir, ancak programlaması nispeten uzun zaman gerektirmektedir.
1990	Karshenas ve Haber	Doğrusal Tamsayılı Model	Kaynak Maliyeti İndirgeme	Proje optimizasyonu için bütünsel bir yaklaşım. Proje maliyetini, kaynak kullanımındaki dalgalanmaları (KDP) ve proje süresini azaltarak (zaman-maliyet ödünleşim problemi) indirgemeyi amaçlamaktadır.
1994	Bandelloni, Tucci ve Rinaldi	Seri olmayan dinamik programlama	Kaynak Dengeleme Problemi (KDP)	Belirli bir kaynak kullanım düzeyinden minimum sapma miktarını hesaplayan amaç fonksiyon kullanılarak küçük ve orta ölçekli problemler çözülmüştür.
1996	Younis ve Saad	Matematiksel model	Kaynak Dengeleme Problemi (KDP)	Çoklu kaynaklı projelerde optimal kaynak dengeleme konusu çalışılmıştır.
1998	Mattila ve Abraham	Doğrusal Tamsayılı Programlama	Kaynak Dengeleme Problemi (KDP)	Doğrusal iş programlarında kaynak dengeleme problemi çalışılmıştır. Bu amaçla bir otoban yol projesi örnek olarak sergilenmiştir. Çözülebilir problemin boyutunu belirlemek için kullanılacak değişken sayısı sunulmuştur.
2000	Neumann and Zimmermann	Dal ve Sınır	Kaynak Dengeleme Problemi (KDP) ve Net Hazır Değer Problemi (NHDP)	Kaynak kullanımındaki dalgalanmalar farklı amaç fonksiyonlarına göre indirgenmeye çalışılmıştır. KDP ve NHDP kısıtlı kaynaklı ve kısıtsız kaynaklı durumlar için çalışılmıştır. KDP, 20 aktiviteli ve 5 kaynaklı problemler için ilk defa çözülmüştür.
2004	Son ve Mattila	Doğrusal Programlama	Kaynak Dengeleme Problemi (KDP)	Aktivitelerin bölünmesinin izin verildiği iş programlarıyla uğraşan bu çalışmada bu yöntemle daha iyi sonuçlara ulaşılabileceği önerilmiştir.
2010	Gather, Zimmermann ve Bartels	Dal ve Sınır	Kaynak Dengeleme Problemi (KDP)	Dal ve sınır tekniğine yeni bir numaralandırma şeması eklenerek KDP çözülmeye çalışılmıştır. Kolisch v.d. (1999) tarafından sunulan <i>rlp_j10</i> ve <i>rlp_j20</i> , Franck v.d. (2001) tarafından oluşturulan <i>ubo10</i> ve <i>ubo20</i> , Neumann ve Zimmermann (2001) tarafında önerilen <i>rlp-15</i> test setleri çözdürülmüş ve bu testler için işlem süreleri verilmiştir
2010	Mutlu	Dal ve Sınır	Kaynak Dengeleme Problemi (KDP)	KDP'nin kesin çözümü için dal ve sınır algoritması kullanılmıştır. Bazı iyileştirmeler eklenerek algoritmanın problemleri daha kısa sürede çözmesi sağlanmıştır.
2012	Rieck, J., Zimmermann, J., and Gather, T.,	Doğrusal Tamsayılı Programlama	Kaynak Dengeleme Problemi (KDP)	Doğrusal tamsayılı programlama yöntemiyle çeşitli modeller sunulmuş ve bu modeller geniş bir problem seti kullanarak test edilmiştir. 50 aktivitel ve beş kaynağa kadar olan problemler karelerin toplamı ve ortalamayı aşan kaynaklar toplamı ölçütüne göre çözdürülmüştür.

Tablo 2. Sezgisel ve üst-sezgisel yöntemlerle ilgili literatür çalışmaları (1/3)

Yıl	Yazar	Yöntem	Problem	Notlar
1962	Burgess ve Killebrew	Sezgisel	Kaynak Dengeleme Problemi (KDP)	Kaynak kullanımındaki dalgalanmaları azaltmak için öncelik kuralını baz alan sezgisel bir yöntem önerilmiştir. Çeşitli dengeleme ölçütleri için uygulanabilir bir yöntemdir.
1975	Woodworth ve Willie	Sezgisel	Kaynak Dengeleme Problemi (KDP)	Çoklu kaynaklı çoklu projelerde sezgisel bir kaynak dengeleme algoritması işlenmiştir.
1990	Harris	Sezgisel	Kaynak Dengeleme Problemi (KDP)	Kaynak dengeleme için kaynak dağılım cetvelinin momentini azaltan PACK yöntemi tanıtılmıştır.
1993	Martinez ve Ioannou	Sezgisel	Kaynak Dengeleme Problemi (KDP)	Kaynak dengelemede Değiştirilmiş Asgari Moment Sezgiseli kullanılmıştır.
1996	Chan, Chua ve Kannan	Üst-Sezgisel (GA)	Kaynak Dengeleme Problemi (KDP) ve Sınırlı Kaynak Tahsisi	Hazır bulunan kaynak miktarından sapmayı en aza indirmeyi amaçlayan GA tabanlı bir yaklaşım sergilenmiştir. KDP ve sınırlı kaynak tahsisi için eş zamanlı genel bir model çalışılmıştır.
1997	Savin, Alkass ve Fazio	Üst-Sezgisel (Sinir Ağları)	Kaynak Dengeleme Problemi (KDP)	Kaynak dengelemede sinir ağları bazlı yöntem kullanılmıştır.
1998	Kartam ve Tongthong	Üst-Sezgisel (Sinir Ağları)	Kaynak Dengeleme Problemi (KDP)	İnşaat projelerinde kaynak dengelemesi yapmak için sinir ağları bazlı bir üst-sezgisel yöntem sunulmuştur.
1999	Neumann ve Zimmermann	Sezgisel	Kısıtlı ve Kısıtsız Kaynaklı Dengeleme Problemi (KDP)	Polinomial öncelik kuralı tabanlı bir üst-sezgisel yöntem geliştirilmiştir. Birçok farklı amaç fonksiyonlar işlenmiştir. Ampirik performans analiz sonuçları sunulmuştur.
1999	Leu ve Yang	Üst-Sezgisel (GA)	Zaman-Maliyet Ödünleşim Problemi, KKİPP, KDP	Zaman-maliyet ödünleşim problemi, KKİPP ve KDP'lerini eş zamanlı çözmeye kabiliyetine sahip GA programlayıcı çalışılmıştır.
1999	Hegazy	Üst-Sezgisel (GA)	Kaynak Dengeleme Problemi (KDP) ve Kaynak Tahsisi	Kaynak tahsisi sezgiselini iyileştirmek amacıyla rastgele aktivite önceliği tanıtılmıştır. Dengeleme sezgiselini değiştirmek için ikili moment yaklaşımı kullanılmıştır. GA modülü aracılığıyla kaynak tahsisi ve kaynak dengeleme problemleri için çoklu amaç optimizasyonu oluşturulmuştur.
1999	Son ve Skibniewski	Çoklu Sezgisel ve Üst-Sezgisel	Kaynak Dengeleme Problemi (KDP)	Kaynak dengelemesi, yerel bir optimizasyon (çoklu sezgisel yaklaşım) ve tavlama benzetimi modülü kullanılarak yapılmıştır. Amaç fonksiyonu olarak karelerin toplamı ölçütü kullanılmıştır.
2000	Leu, Yang ve Huang	Genetik Algoritma (GA)	Kaynak Dengeleme Problemi (KDP)	Ortalama kaynak kullanımı ve asıl kaynak kullanımı arasındaki sapma en aza indirilerek dengeleme yapılmıştır. Planlamacıların birçok senaryoyu dikkate alabildiği GA tabanlı bir sistem geliştirilmiştir.

Tablo 2. Sezgisel ve üst-sezgisel yöntemlerle ilgili literatür çalışmaları (2/3)

Yıl	Yazar	Yöntem	Problem	Notlar
2001	Hiyassat	Sezgisel	Kaynak Dengeleme Problemi (KDP)	Çoklu kaynaklı projelerde kaynak dengelemesi için değiştirilmiş asgari moment yaklaşımı kullanılmıştır.
2001	Hegazy ve Ersahin	Sezgisel	Eş zamanlı KDP, KKİPP, ve Zaman-Maliyet Ödünleşim Problemi	Zaman-maliyet ödünleşim analizi, kaynak tahsisi ve kaynak dengelemesi problemleri için toplam proje maliyetini azaltan amaç fonksiyonunu kullanan tablolama tabanlı bir yöntem tanıtılmıştır.
2001	Wang ve Zheng	Üst Sezgisel (GA)	İş Programlama Problemi	Genetik algoritma ve tavlama benzetimi yöntemleri paralel kullanılarak oluşturulan hibrid modelle GA'lar geliştirilmiştir.
2003	Oral, Laptalı Oral, Bozkurt ve Erdiş	Üst-Sezgisel (GA)	Kaynak Dengeleme Problemi (KDP)	GA tabanlı kaynak dengeleme modülü geliştirilmiştir. Ortalama kaynak kullanımından sapmalar en aza indirmeye çalışılmıştır.
2003	Zheng, Ng ve Kumaraswamy	Üst-Sezgisel (GA)	Kaynak Dengeleme Problemi (KDP)	Çoklu kaynaklı dengeleme problemleri için çok amaçlı GA tabanlı bir teknik kullanılmıştır.
2004	Senouci ve Eldin	Üst-Sezgisel (GA)	Kaynak Dengeleme Problemi (KDP) ve KKİPP eş zamanlı olarak	Proje programlama problemlerine bütünsel bir yaklaşım sergileyen GA tabanlı bir yöntem içerir. Eş zamanlı olarak KDP ve KKİPP problemleriyle ilgilenmiştir.
2005	Chen, Lee ve Park	Üst-Sezgisel (GA)	Mühendislik Problemleri	Tavlama benzetimli GA'ların yüksek performanslı bilgisayarda paralel çalıştırılmasıyla algoritmanın performansı iyileştirilmeye çalışılmıştır.
2006	Hwang ve He	Üst-Sezgisel (GA)	Mühendislik Problemleri	GA'ların tepe tırmanma ve lokal optimum çözümlere takılıp kalma gibi problemleri tavlama benzetimi kullanarak iyileştirilmeye çalışılmıştır.
2006	Han ve Sun	Üst-Sezgisel (GA)	Mühendislik Problemleri	GA'lara tavlama benzetimi eklenerek GA'ların daha hızlı işlem yapması ve lokal optimum noktalarında takılıp kalmaması sağlanmaya çalışılmıştır.
2006	Xiong ve Kuang	Üst-Sezgisel ve Çoklu Sezgisel	Kaynak Dengeleme Problemi (KDP)	Kaynak dengeleme problemi için seri programlama jenerasyon şeması ve karınca kolonisi tekniklerini kullanan hibrid bir yöntem önermiştir.
2007	Balestin, Schwindt ve Zimmermann	Sezgisel	Kaynak Dengeleme Problemi (KDP)	Poplasyon tabanlı kaynak dengeleme yöntemi sunulmuştur. Sipariş üzerine üretim sistemi düşünülerek üretim planlama problemi çalışılmış, daha sonra bu model KDP'ye uygulanmıştır.
2007	Qi, Wang ve Guo	Üst-Sezgisel (PSO)	Kaynak Dengeleme Problemi (KDP)	Kaynak dengeleme problemi için geliştirilmiş kuş sürüsü optimizasyonu (G-PSO) kullanılmıştır.
2008	Pang, Shi ve You	Üst Sezgisel (PSO)	Kaynak Dengeleme Problemi (KDP)	Tek kaynaklı dengelemeler için kısıtlama faktörlü kuş sürüsü optimizasyonu (PSO) kullanılmıştır. 9 aktivite bir iş programı örnek olarak çalışılmıştır.
2008	Roca, Pugnaghi ve Libert	Üst-Sezgisel (GA)	Kaynak Dengeleme Problemi (KDP) ve KKİPP eş zamanlı olarak	Kaynak dengeleme problemi (KDP) ve kısıtlı kaynaklı iş programlama problemini (KKİPP) eş zamanlı olarak çözebilmek için genetik algoritma tabanlı çok amaçlı bir optimizasyon önerilmiştir.



Tablo 2. Sezgisel ve üst-sezgisel yöntemlerle ilgili literatür çalışmaları (3/3)

Yıl	Yazar	Yöntem	Problem	Notlar
2009	Bettemir	Üst-Sezgisel (GA)	Eş zamanlı KDP, KKİPP, ve Zaman-Maliyet Ödünleşim Problemi	Zaman-maliyet ödünleşim, kaynak dengeleme, ve kısıtlı kaynaklı iş programlaması problemlerinin çözümünde GA'ların tavlama benzetimi yöntemi ile melezlenmesinin sonuçların geliştirilmiştir.
2009	Guo, Li ve Ye	Üst-Sezgisel (PSO)	Kaynak Dengeleme Problemi (KDP)	Çoklu kaynaklı çoklu projelerin çözümü için PSO tabanlı yöntem kullanılmıştır.
2009	El-Rayes ve Jun	Üst-Sezgisel (GA)	Kaynak Dengeleme Problemi (KDP)	Atıl kaynak günü ve işten çıkarma-işe alma sayısı şeklinde iki yeni ölçüt tanımlanmıştır. Bu ölçütler kullanılarak KDP'yi çözmek için GA tabanlı bir modül geliştirilmiştir.
2010	Christodoulou, Ellinas ve Kamenou	Sezgisel	Kaynak Dengeleme Problemi (KDP)	Entropi maksimizasyonunun kullanıldığı asgari moment yöntemi çalışılmıştır. Daha iyi kaynak dengelemesi yapabilmek için aktivite sıkıştırma ve gevşetmesine izin verilmiştir.
2010	Geng, Weng ve Liu	Üst-Sezgisel (KKO)	Kaynak Dengeleme Problemi (KDP)	Kaynak dengeleme problemini çözmek için yönlümel karınca kolonisi optimizasyonu kullanılmıştır.
2011	Doulabi, Seifi ve Shariat	Üst-Sezgisel (Onarım mekanizmalı GA)	Kaynak Dengeleme Problemi (KDP)	Kaynak dengeleme problemini çözmek için yerel arama sezgiseli ve onarım mekanizması içeren hibrid bir genetik algoritması önerilmiştir.
2011	Jun ve El-Rayes	Üst-Sezgisel (GA)	Kaynak Dengeleme Problemi (KDP) ve KKİPP eş zamanlı olarak	Kaynak dengeleme problemi (KDP) ve kısıtlı kaynaklı iş programlama problemini (KKİPP) eş zamanlı olarak çözebilmek için çok amaçlı bir GA sunulmuştur.
2012	Alsayegh ve Hariga	Üst Sezgisel (PSO ve TB)	Kaynak Dengeleme Problemi (KDP)	Aktivite sürelerinin bölünebildiği KDP için melez kuş sürüsü (PSO) ve tavlama benzetimi algoritması önerilmiştir.
2013	Ponz-Tienda v.d.	Üst-Sezgisel (GA)	Kaynak Dengeleme Problemi (KDP)	KDP için bir adaptasyonlu GA geliştirilmiş ve 120 aktivite ve dört kaynağa kadar olan problemler karelerin toplamı ölçütü için test edilmiştir.



### 3. DOĞRUSAL TAMSAYILI MODELLER VE MS EXCEL-GAMS/CPLEX UYGULAMASI

Kaynak dengelemesi problemi için kesin çözümleri olan kritik yol yöntemine göre çözülmüş problem setleri literatürde mevcut değildir. Gerek proje kapsamında geliştirilen dal ve sınır algoritmasının sonuçlarının kontrol edilebilmesi için, gerekse de özellikle 20 aktiviteden büyük şebekeler için dal ve sınır algoritmasına alternatif olması açısından, doğrusal tamsayılı model yaklaşımı ile kesin çözüm üreten bir uygulama geliştirilmiştir. Geliştirilen uygulama, en küçük mutlak fark, karaların toplamı, azami günlük kaynak kullanımı ve atıl kaynak günü ölçütleri için çok kaynaklı kaynak dengelemesi probleminin doğrusal programla yöntemi ile modellenmesine imkan vermektedir. Uygulama için geliştirilen modeller 3.1 nolu bölümde verilmiştir. Bu modeli baz alarak geliştirilen Excel ve GAMS uygulamasının ayrıntılarına 3.2 nolu bölümde değinilmiştir.

#### 3.1 Doğrusal Tamsayılı Eniyileme (Optimizasyon) Modeli

##### 3.1.1. Kümeler

$I$ : Proje aktiviteleri,  $i = 1, \dots, I$

$T$ : Projedeki günler,  $t = 0, \dots, T$

$K$ : Projede kullanılan kaynak çeşitleri,  $k = 1, \dots, K$

$N$ : Herhangi bir kaynağın bütün aktiviteler tarafından kullanılan toplam günlük miktarı,  
 $n = 1, \dots, N$

##### 3.1.2. Parametreler

$EST_i$  :  $i$  aktivitesinin en erken başlayabileceği gün.

$LST_i$  :  $i$  aktivitesinin en geç başlayabileceği gün.

$d_i$  :  $i$  aktivitesinin süresi.

$r_{i,k}$  :  $i$  aktivitesinin kullandığı  $k$  kaynak miktarı.

$a_{t,k}$  :  $t$  gününde kullanılması hedeflenen  $k$  kaynak miktarı.

$w_k$ :  $k$  kaynağının ağırlığı

$D$  : Projenin en geç bitirilebileceği gün sayısı.

$p_{i,j}$  : Eğer  $j$  aktivitesi  $i$  aktivitesinden önce bitirilmeliyse 1; böyle bir şart yoksa 0.

##### 3.1.3. Değişkenler

$z_1$  : Hedeflenen günlük kaynak kullanımından olan sapmaların mutlak değerlerinin ağırlıklı toplamı.

$z_2$  : Kaynakların toplam günlük kullanımlarının ağırlıklı toplamı.

$z_3$  : Kaynakların proje süresince en fazla kullanılan günlük miktarlarının ağırlıklı toplamı (azami günlük kaynak ihtiyacı).

$z_4$  : Kaynakların her gün için o günden önce ve sonra kullanılan en büyük günlük miktarları arasından daha küçük olanı ile kaynağın o gün kullanılan miktarı arasındaki farkların toplamı (toplam atıl kaynak günü sayısı).

$f_i$  :  $i$  aktivitesinin başlayacağı gün.

$u_{t,k}$  :  $t$  gününde kullanılan  $k$  kaynak miktarı.

$mxu_k$  :  $k$  kaynağının proje süresince bir günde en fazla kullanıldığı miktar.

$mx1u_{t,k}$  :  $k$  kaynağının  $t$  gününden önce bir günde en fazla kullanıldığı miktar.

$mx2u_{t,k}$  :  $k$  kaynağının  $t$  gününden sonra bir günde en fazla kullanıldığı miktar.

$mnu_{t,k}$  : Her kaynak ve her gün için  $mx1$  ve  $mx2$  arasından daha küçük olan değer.

$v_{t,k}$  :  $t$  gününde kullanılan  $k$  kaynak miktarının karesi.

$x_{t,k}$  :  $t$  gününde hedeflenen  $k$  kaynak miktarından ( $a_{t,k}$ ) fazla olarak kullanılmış kaynak miktarı.

$y_{t,k}$  :  $t$  gününde hedeflenen  $k$  kaynak miktarından ( $a_{t,k}$ ) az olarak kullanılmış kaynak miktarı.

$\lambda_{n,t,k}$  :  $t$  gününde  $k$  kaynağından toplam  $n$  kadar kullanılıyorsa 1; başka bir miktar kullanılıyorsa 0.

$\varphi_{t,i}$  : Eğer  $i$  aktivitesi  $t$  gününde yapılıyorsa 1; yapılmıyorsa 0.

$\sigma_{t,i}$  : Eğer  $i$  aktivitesi  $t$  gününde başlamışsa 1; başlamamışsa 0.

### 3.1.4. Modelleme

Kaynakların dengeli kullanımı için değişik ölçütler (amaç fonksiyonu) kurgulanabilir. Projede literatürde geçen dört ölçüt modellenmiştir. (1) numaralı ölçüt, kaynakların hedeflenen günlük kaynak kullanım miktarından olan günlük sapmalarının mutlak değerlerinin ağırlıklı toplamını en aza indirmektedir. (2) numaralı ölçüt, günlük kaynak kullanım miktarlarının karelerinin ağırlıklı toplamını en aza indirmektedir. (3) numaralı ölçüt (MRD – Maximum Resource Demand – Azami Kaynak Kullanımı), proje süresince kullanılan en büyük günlük kaynak kullanım miktarı en küçük değere sahip olacak şekilde çözüm bulmaktadır. (4) numaralı ölçüt (RID – Resource Idle Days - Atıl Kaynak Günü) ise projedeki her gün için o günden önce kullanılan en büyük kaynak miktarı ile o günden sonra kullanılan en büyük kaynak miktarı arasından daha küçük olanın o gün kullanılan kaynak miktarından olan farklarının toplamını en aza indirmektedir.



$$\min z_1 = \sum_t \sum_k w_k |u_{t,k} - a_{t,k}| \quad (1)$$

$$\min z_2 = \sum_t \sum_k w_k u_{t,k}^2 \quad (2)$$

$$\min z_3 = \sum_k w_k \max(u_{1,k}, \dots, u_{T,k}) \quad (3)$$

$$\min z_4 = \sum_t \sum_k w_k (\min(\max(u_{1,k}, \dots, u_{t,k}), \max(u_{t,k}, \dots, u_{T,k})) - u_{t,k}) \quad (4)$$

Yukarıda bahsedilen fonksiyonlar doğrusal olmadığı için modeller önce doğrusal olarak ifade edilmiştir. İlerleyen bölümlerde her modelin ölçütü ve diğer modellerden farklı olan kısıtları verildikten sonra ortak kısıtlar bir arada gösterilmiştir.

*Model 1:*

$$\min z_1 = \sum_t \sum_k w_k (x_{t,k} + y_{t,k}) \quad (5)$$

*Kısıtlar:*

$$u_{t,k} - a_{t,k} = x_{t,k} - y_{t,k} \quad \forall t \in T, \forall k \in K \quad (6)$$

$$x_{t,k}, y_{t,k} \in \mathbb{Z}_0 \quad \forall t \in T, \forall k \in K \quad (7)$$

Model 1'in ölçütü hedeflenen günlük kaynak kullanım miktarından olan sapmaların mutlak değerlerinin ağırlıklı toplamını en aza indirmektedir. Mutlak değer fonksiyonu doğrusal olmadığı için (6) nolu kısıtta  $u_{t,k} - a_{t,k}$  değerleri iki negatif olmayan değişkenin farkı olarak ifade edilerek model doğrusal hale getirilmiştir.  $x_{t,k}, y_{t,k}$  değişkenleri 0 veya daha büyük tamsayı değerleri alabilir.

*Model 2:*

$$\min z_2 = \sum_t \sum_k w_k v_{t,k} \quad (8)$$

*Kısıtlar:*

$$u_{t,k} = \sum_n n \lambda_{n,t,k} \quad \forall t \in T, \forall k \in K \quad (9)$$

$$v_{t,k} = \sum_n n^2 \lambda_{n,t,k} \quad \forall t \in T, \forall k \in K \quad (10)$$

$$\sum_n \lambda_{n,t,k} = 1 \quad \forall t \in T, \forall k \in K \quad (11)$$

$$v_{t,k} \in \mathbb{Z}_0 \quad \forall t \in T, \forall k \in K \quad (12)$$

$$\lambda_{n,t,k} \in \{0,1\} \quad \forall n \in N, \forall t \in T, \forall k \in K \quad (13)$$

Model 2'nin ölçütü günlük kaynak kullanım miktarlarının karelerinin ağırlıklı toplamını en aza indirmektedir. (9) kullanılan toplam  $k$  kaynak miktarını bulmaktadır. (10) bu değer karesine eşittir. (11) ise  $k$  kaynağının günlük kullanım toplamının sadece tek bir değere eşit olmasını sağlamaktadır.  $v_{t,k}$  değişkenleri 0 veya daha büyük tamsayı değerler alabilir.  $\lambda_{n,t,k}$  değişkenleri ikili değişkenlerdir.

*Model 3:*

$$\min z_3 = \sum_k w_k mxu_k \quad (14)$$

*Kısıtlar:*

$$u_{t,k} \leq mxu_k \quad \forall t \in T, \forall k \in K \quad (15)$$

$$mxu_k \in \mathbb{Z}_0 \quad \forall k \in K \quad (16)$$

Model 3'ün ölçütü kaynakların proje süresince kullanıldığı en büyük günlük miktarlarının ağırlıklı toplamını en aza indirmektedir. Her kaynağın kullanıldığı en büyük günlük miktar bir değişken ( $mxu_k$ ) ile ifade edildikten sonra her kaynağın günlük kullanım miktarının bu miktardan büyük olamayacağı koşulu eklenerek (15) model doğrusal hale getirilmiştir.  $mxu_k$  değişkenleri 0 veya daha büyük tamsayı değerler alabilir.

Model 4:

$$\min z_4 = \sum_t \sum_k w_k (mnu_{t,k} - u_{t,k}) \quad (17)$$

Kısıtlar:

$$u_{t',k} \leq mx1u_{t,k} \quad \forall t' \leq t, \forall t \in T, \forall k \in K \quad (18)$$

$$u_{t',k} \leq mx2u_{t,k} \quad \forall t' \geq t, \forall t \in T, \forall k \in K \quad (19)$$

$$mnu_{t,k} \leq mx1u_{t,k} \quad \forall t \in T, \forall k \in K \quad (20)$$

$$mnu_{t,k} \leq mx2u_{t,k} \quad \forall t \in T, \forall k \in K \quad (21)$$

$$mx1u_{t,k}, mx2u_{t,k}, mnu_{t,k} \in \mathbb{Z}_0 \quad \forall t \in T, \forall k \in K \quad (22)$$

Model 4'ün ölçütü, her gün için o günden önce ve sonra gerçekleşen en büyük kaynak kullanımları arasından daha küçük olan değer in o gün gerçekleşen kaynak kullanımından olan farklarının ağırlıklı toplamını en aza indirmektedir.  $mx1u_{t,k}$ , değişkenleri  $t$  günden önce olan en büyük kaynak kullanımınıdır.  $mx2u_{t,k}$ , değişkenleri ise  $t$  günden sonra olan en büyük kaynak kullanımınıdır. (18) ve (19)  $t$  günden önce ve sonra gerçekleşen kaynak kullanımlarının hiç bir kaynak için  $t$  gününe kadar olan ve  $t$  günden sonra olan en büyük kaynak kullanımından daha fazla olamayacağını söylemektedir. (20) ve (21) ise her gün ve her kaynak için  $t$  günden önce ve sonra gerçekleşen en büyük kaynak kullanımları arasından daha küçük olanı bulmaktadır.  $mx1u_{t,k}$ ,  $mx2u_{t,k}$ ,  $mnu_{t,k}$  değişkenleri 0 veya daha büyük tamsayı değerler alabilir.

Ortak Kısıtlar:

$$\sum_i r_{i,k} \varphi_{t,i} = u_{t,k} \quad \forall t \in T, \forall k \in K \quad (23)$$

$$p_{i,j} f_i \geq p_{i,j} (f_j + d_j) \quad \forall i, j \in I, i \neq j \quad (24)$$

$$\sum_{EST_i \leq t \leq LST_i} t \sigma_{t,i} = f_i \quad \forall i \in I \quad (25)$$



$$\sum_{EST_i \leq t \leq LST_i} \sigma_{t,i} = 1 \quad \forall i \in I \quad (26)$$

$$\varphi_{t,i} = \sum_{t=\max(EST_i, t-d_i+1)}^{\min(LST_i, t)} \sigma_{t,i} \quad \forall t \in T, \forall i \in I, EST_i \leq t \leq LST_i + d_i - 1 \quad (27)$$

$$\varphi_{t,i} = 0 \quad \forall t \in T, \forall i \in I, t < EST_i \quad (28)$$

$$\varphi_{t,i} = 0 \quad \forall t \in T, \forall i \in I, t > LST_i + d_i - 1 \quad (29)$$

$$f_1 = 0 \quad (30)$$

$$f_i \leq D \quad (31)$$

$$\sigma_{0,1} = 1 \quad (32)$$

$$u_{t,k} \in \mathbb{Z}_0 \quad \forall t \in T, \forall k \in K \quad (33)$$

$$f_i \in \mathbb{Z}_0 \quad \forall i \in I \quad (34)$$

$$\varphi_{t,i} \in \{0,1\} \quad \forall t \in T, \forall i \in I \quad (35)$$

$$\sigma_{t,i} \in \{0,1\} \quad \forall t \in T, \forall i \in I \quad (36)$$

Aktivitelerin zamanlaması ile ilgili kısıtlar bütün modellerde ortaktır. (23) günlük kaynak kullanım miktarını bulmaktadır. Aktivitelerin sadece aktif oldukları günlerde kaynak tüketmeleri bu kısıtla zorlanmaktadır. (24) aktivitelerin kendilerinden önce bitmesi gereken aktiviteler bitmeden başlamalarını engellemektedir. (25) aktivitelerin başladığı günü bulmaktadır. (26) her aktivitenin en erken ve en geç başlama günleri arasındaki sadece bir günde başlatılabileceğini belirtmektedir. (27) aktivitelerin hangi günler faal olduğunu bulmaktadır, ve faal olunan günlerin ardışık günler olmasını sağlamaktadır. (28) ve (29), bir aktivitenin üzerinde çalışılmasının mümkün olduğu günler dışında faal olamayacağını belirtmektedir. İlk ve son aktiviteler projenin başlangıcını ve sonunu belirlemek için oluşturulmuş yapay aktivitelerdir. (30) ilk aktivite olan 1 nolu yapay aktivitenin hemen başlayacağını belirtmekte, (31) ise bütün işlerin projenin en geç bitirilmesi gereken günden

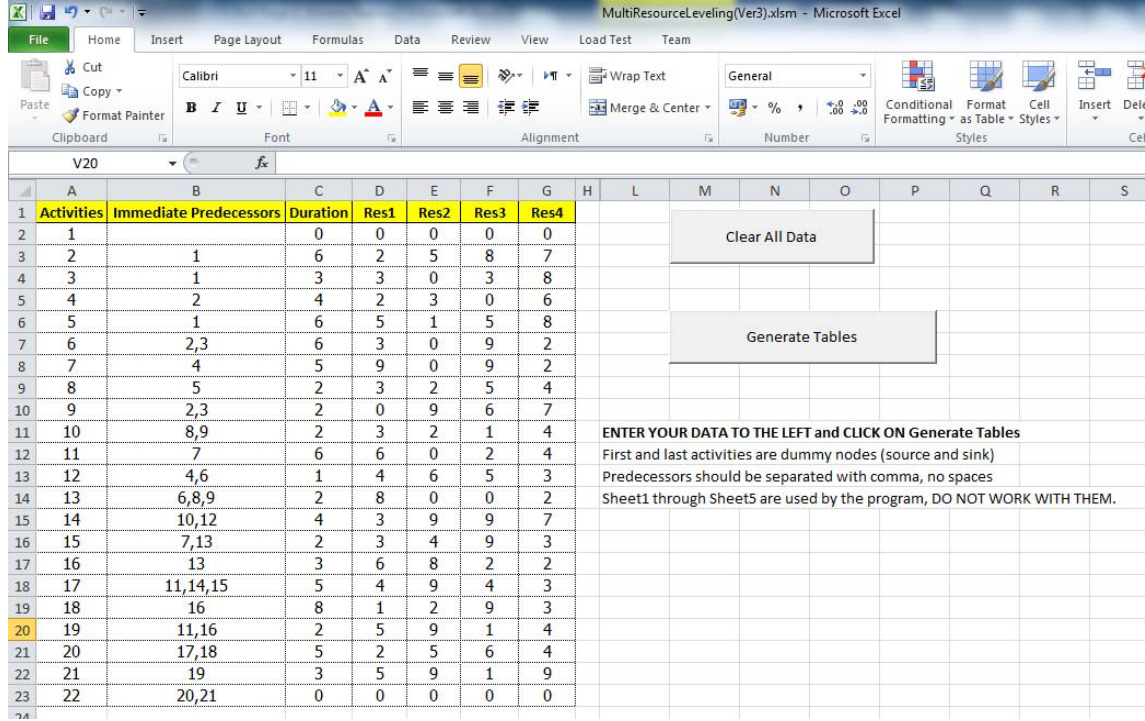


önce bitmesi gerektiğini söylemektedir. (32) birinci aktivitenin  $t = 0$  zamanında aktif olduğunu söylemektedir.  $u_{t,k}$ ,  $f_i$  değişkenleri 0 veya daha büyük tamsayı değerler alabilir.  $\varphi_{t,i}$ ,  $\sigma_{t,i}$  değişkenleri ise ikili değişkenlerdir.

### 3.2 MS Excel ve GAMS/CPLEX Uygulaması

Doğrusal tamsayılama yöntemiyle tanımlanan kesin çözüm uygulaması MS Excel'e girilen problem bilgileri kullanılarak çözümlenme modellerinin üretilmesi ve daha sonra eniyileme işlemlerinin GAMS adlı programa yaptırılmasıyla çalışır. GAMS içinde çözücü olarak ise CPLEX kullanılmaktadır. Şekil 1'de El Rayes ve Jun (2009)'a ait 22 aktivite (ilk ve son aktiviteler başlama ve bitişini temsil etmektedir) problemin geliştirilen uygulama ile çözdürülmesi örnek olarak verilmiştir. MS Excel'de açılan sayfada, aktivitelere ait numaralar, öncel aktiviteler, çalışma süreleri ve kaynak ihtiyaçları girilir. Çok kaynaklı problemler için her kaynak türü ayrı bir sütuna girilir. Literatürde hazırda bulunan örnek problemlerde genellikle aktivitelere ait ardıl aktivite bilgileri verildiğinden, MS Excel'de hazırlanan bir sayfada findpredecessors tablosu kullanılarak ardıl aktivite bilgilerinden öncel aktivite bilgileri elde edilebilir. Bu şekilde problem bilgileri girildikten sonra "Generate Tables" butonu tıklanarak probleme ait KYY iş programı bilgileri ve çözüm için kullanıcı tercihleri sayfası açılır (Şekil 2). Açılan sayfada kaç kaynaklı çözüm yapılacağı ve tercih edilen ölçüt, her kaynağın çözüm için kullanılacak ağırlıkları girilir. Şekil 1'de verilen MS Excel görüntüsünde 4. satır M sütunundaki hücrede çözüm için istenilen kaynak sayısı girilir. 5'ten küçük girilen sayılar için soldan itibaren girilen sayı kadar sütundaki kaynaklar çözüm için dikkate alınır. 6. satır M sütunundaki hücrede *MinAbsDev* (mutlak sapma miktarı) için "1", *MinSSQR* (karelerin toplamı) için "2", *weighted MRD+RID* (azami kaynak ihtiyacı ve atıl kaynak günlerinin ağırlıklı toplamı) için "3" rakamları girilerek ölçüt tercihi yapılır. Aynı sütunun 8 ve 9. satırlarında RID ve MRD için ayrı ayrı ağırlıklar girilir. Eğer sadece RID ölçütü kullanılmak isteniyorsa RID ağırlığı "1", MRD ağırlığı "0" olarak girilir, benzer şekilde sadece MRD ölçütü kullanılmak isteniyorsa MRD ağırlığı "1", RID ağırlığı "0" olarak girilir. Daha sonra "Run B-K Heuristic & GAMS" butonuna basılarak problem çözdürülür.





Activities	Immediate Predecessors	Duration	Res1	Res2	Res3	Res4
1		0	0	0	0	0
2	1	6	2	5	8	7
3	1	3	3	0	3	8
4	2	4	2	3	0	6
5	1	6	5	1	5	8
6	2,3	6	3	0	9	2
7	4	5	9	0	9	2
8	5	2	3	2	5	4
9	2,3	2	0	9	6	7
10	8,9	2	3	2	1	4
11	7	6	6	0	2	4
12	4,6	1	4	6	5	3
13	6,8,9	2	8	0	0	2
14	10,12	4	3	9	9	7
15	7,13	2	3	4	9	3
16	13	3	6	8	2	2
17	11,14,15	5	4	9	4	3
18	16	8	1	2	9	3
19	11,16	2	5	9	1	4
20	17,18	5	2	5	6	4
21	19	3	5	9	1	9
22	20,21	0	0	0	0	0

Şekil 1. Problem bilgilerinin Excel’de girilmesi

Uygulama çalıştırıldığında MS Excel’de doğrusal tamsayılı modele göre hazırlanan veriler GAMS programına aktararak problem çözdürülmeye başlanır (Şekil 3). GAMS problemin çözümünü bulduğu zaman ekranda beliren pencerede “Load Data” butonuna basılarak sonuçlar MS Excel’e alınır (Şekil 4). Sonuç sayfasında kesin çözüme ait 2. satır B sütununda verilen ölçüt değeri GAMS’ten alınan ve kaynak sayısına göre ağırlıklı ortalaması alınmamış sonuç değeridir. 1. satır F sütununda verilen ölçüt değeri ise kaynak sayısına göre ve kullanılan fonksiyon sayısına göre (4 farklı kaynak ve 2 farklı fonksiyon – MRD & RID, dolayısıyla GAMS’te bulunan sonuç 8’e bölünmüş) ağırlıklı ortalaması alınmış ölçüt değeridir. Sayfada başlama süreleri (3. satırdan itibaren B sütunu), ve proje süresi boyunca her kaynağa ait günlük kaynak kullanım miktarları (F, G, H ve I sütunları) verilmiştir.

S4											Optimize_ResourceLeveling_Temporal.xlsm - Microsoft Excel						
	A	B	C	D	E	F	G	H	I	J	L	M	N	O	P	Q	R
1	Activ	ties	EST	EFT	LST	LFT	Durat	Res	Res	Res	Res	Target	Target	Target	Target	Target	Rounded up for each activity and
2	1	0	0	0	0	0	0	0	0	0	0	Res1	Res2	Res3	Res4	Res5	then summed up, Rieck2012, p.28
3	2	0	6	0	6	6	2	5	8	7	Critical Path	Level	Level	Level	Level	Level	(Incorrect?) I use roundup after
4	3	0	3	4	7	3	3	0	3	8	Weights (Integer)	1	1	1	1	1	Change target levels if necessary
5	4	6	10	6	10	4	2	3	0	6	Number of resources to	4					Run B-K Heuristic & GAMS
6	5	0	6	5	11	6	5	1	5	8	be used	100					
7	6	6	12	7	13	6	3	0	9	2	Number of priority lists	3	1 - MinAbsDev; 2 - MinSSQR; 3 - Min(weighted MRD+RID)				
8	7	10	15	10	15	5	9	0	9	2	Objective function	60	Currently same for all resources, needed for OF 2				
9	8	6	8	11	13	2	3	2	5	4	Max daily total resource	1					
10	9	6	8	11	13	2	0	9	6	7	usage (estimate, can						
11	10	8	10	15	17	2	3	2	1	4	cause problem						
12	11	15	21	15	21	6	6	0	2	4	2 infeasibility if too small)						
13	12	12	13	16	17	1	4	6	5	3	Weight of MRD						
14	13	12	14	13	15	2	8	0	0	2	Weight of RID						
15	14	13	17	17	21	4	3	9	9	7							
16	15	15	17	19	21	2	3	4	9	3							
17	16	14	17	15	18	3	6	8	2	2							
18	17	21	26	21	26	5	4	9	4	3							
19	18	17	25	18	26	8	1	2	9	3							
20	19	21	23	26	28	2	5	9	1	4							
21	20	26	31	26	31	5	2	5	6	4							
22	21	23	26	28	31	3	5	9	1	9							
23	22	31	31	31	31	0	0	0	0	0							

Şekil 2. KYY iş programlama sonuçları ve çözüm için kullanıcı tercihleri

```

C:\Users\Huseyin\Desktop\Excel MinDev Optimal Solver\GAMS>
Fixed MIP status(1): optimal
Proven optimal solution.
MIP Solution:          200.000000    <236424 iterations, 9966 nodes>
Final Solve:          200.000000    <0 iterations>
Best possible:        200.000000
Absolute gap:         0.000000
Relative gap:         0.000000

--- Restarting execution
--- resourceLeveling.gms(406) 0 Mb
--- Reading solution for model resourceLeveling
--- resourceLeveling.gms(418) 3 Mb
--- Putfile out C:\Users\Huseyin\Desktop\Excel MinDev Optimal Solver\GAMS\resou
ceLeveling_z.txt
--- Putfile out1 C:\Users\Huseyin\Desktop\Excel MinDev Optimal Solver\GAMS\resou
rceLeveling_t.txt
--- Putfile out2 C:\Users\Huseyin\Desktop\Excel MinDev Optimal Solver\GAMS\resou
rceLeveling_u.txt
*** Status: Normal completion
--- Job resourceLeveling.gms Stop 03/11/13 09:38:32 elapsed 0:00:37.659
C:\Users\Huseyin\Desktop\Excel MinDev Optimal Solver\GAMS>

```

Şekil 3. MRD+RID için GAMS programının çalışma sonuçları

	A	B	C	D	E	F	G	H	I
1	B-K	Optimal	% Gap		Result =>	25			
2	249	200,00	24,50						
3	Activity	Optimal starting d	B-K starting day		Starting d.	Usage 1	Usage 2	Usage 3	Usage 4
4	1	0	0		0	2	5	8	7
5	2	0	0		1	5	5	11	15
6	3	1	1		2	5	5	11	15
7	4	6	6		3	5	5	11	15
8	5	4	4		4	7	6	13	15
9	6	6	6		5	7	6	13	15
10	7	10	10		6	10	4	14	16
11	8	10	6		7	10	4	14	16
12	9	10	10		8	10	4	14	16
13	10	13	8		9	10	4	14	16
14	11	15	15		10	15	11	29	15
15	12	12	12		11	15	11	29	15
16	13	12	12		12	21	6	14	7
17	14	17	13		13	20	2	10	8
18	15	15	17		14	18	10	12	8
19	16	14	14		15	15	12	13	9
20	17	21	21		16	15	12	13	9
21	18	17	17		17	10	11	20	14
22	19	21	21		18	10	11	20	14
23	20	26	26		19	10	11	20	14
24	21	23	23		20	10	11	20	14
25	22	31	31		21	10	20	14	10
26					22	10	20	14	10
27					23	10	20	14	15
28					24	10	20	14	15
29					25	9	18	5	12
30					26	2	5	6	4
31					27	2	5	6	4
32					28	2	5	6	4
33					29	2	5	6	4
34					30	2	5	6	4

Şekil 4. El Rayes ve Jun (2009)'a ait problemin MRD+RID toplamını en aza indiren ölçüte göre çok kaynaklı (4 kaynak) çözümü

Geliştirilen modeller ve MS Excel ve GAMS/CPLEX uygulaması bir sonraki bölümde anlatılan ve proje kapsamında geliştirilen dal ve sınır algoritmasının sonuçlarının doğruluğunun sınanmasında kullanılmıştır. Ayrıca MS Excel ve GAMS/CPLEX uygulamasının mutlak sapma miktarı ve karelerin toplamı ölçütleri için dal ve sınır algoritmasının çözemediği büyük ölçekli problemlerin kesin sonuçlarının elde edilmesinde kullanılmıştır.

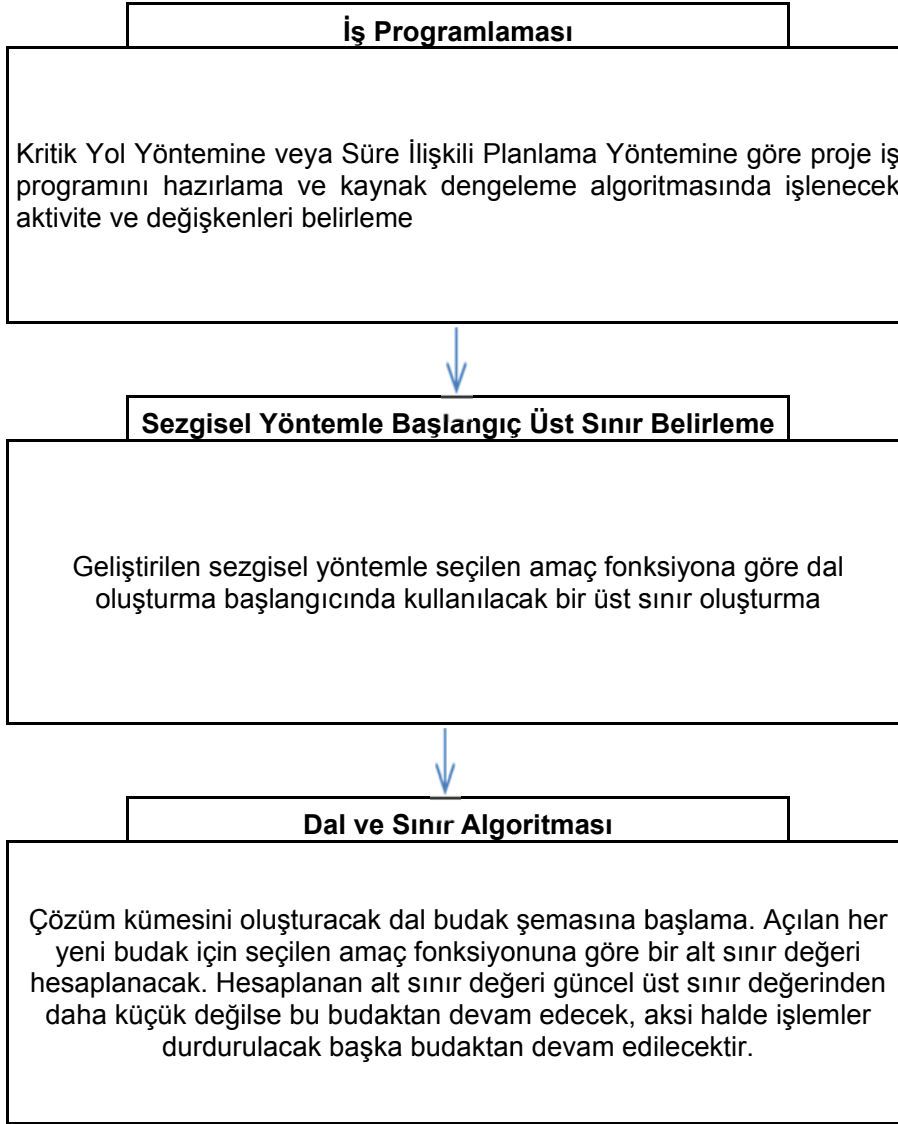


## 4. DAL ve SINIR ALGORİTMASI

Kaynak dengelemesi probleminin çözümünde dal ve sınır yönteminin en büyük avantajı her ölçüt (amaç fonksiyonu) için uygulanabilmesidir. Aynı zamanda, dal ve sınır yönteminin esnekliği, performans ve hız bakımından algoritmanın geliştirilmesini daha kolay ve mümkün kılmaktadır. Bununla birlikte literatürde yer alan mevcut dal ve sınır algoritmaları 20 aktivitelik KDP problem setlerinden bir kısmını 10 saatlik bir süre içerisinde çözememektedir (Gather v.d. 2010). Bu nedenle bu projede, KDP'nin kesin çözümü için mevcut dal ve sınır algoritmalarının performansını arttıracak iyileştirmeler geliştirilmesi amaçlanmıştır. Bu doğrultuda üç farklı iyileştirme stratejisi değerlendirilmiştir. Bu stratejilerin temel amacı algoritmanın hem daha az hafıza kullanmasını, hem de, daha hızlı çalışmasını sağlamaktır. Uygulanacak birinci strateji dal ve sınır algoritmasında kullanılacak ilk üst sınır değerinin iyileştirilmesidir. Bu doğrultuda projede geliştirilen dal ve sınır algoritmasına özel kısa sürede iyi sonuç verebilecek alternatif sezgisel geliştirilmiştir. İkinci strateji, Zamani (2001)'in kaynak kısıtlı iş programlaması probleminde kullandığı dal ve sınır algoritması yönteminin KDP'ye uyarlanmasıdır. Zamani (2001) geliştirdiği etkin dal ve sınır algoritması yönetimi ile kaynak kısıtlı iş programlaması problemi için kısa süre ve az hafıza gerektiren bir algoritma sunmuştur. Bu dal ve sınır yönteminin KDP için de gerek süre gerekse hafıza açısından iyileştirme potansiyeli değerlendirilmiştir. Üçüncü ve son strateji ise Neumann ve Zimmermann (2000)'in KDP için geliştirdiği alt sınır hesaplama yönteminin iyileştirilmesidir. Bu doğrultuda Çağdaş Mutlu'nun yüksek lisans tez çalışmasında sunulan alt sınır hesaplanması yöntemine (Mutlu 2010) ilaveten kaynak tahditlerinin güncellendiği, dinamik bir yöntem geliştirilmiştir. Ayrıca bu projede Zimmermann (2000)'in geliştirdiği alt sınır hesaplama yönteminine alternatif KDP için yeni bir alt sınır hesaplama yöntemi sunulmuştur.

### 4.1. Dal ve Sınır Algoritmasının Temel Bölümleri

Proje kapsamında KDP için geliştirilen dal ve sınır algoritması üç ana bölümden oluşmaktadır. İlk kısımda iş programını hazırlama ve kaynak dengeleme algoritmasında işlenecek aktivite ve değişkenleri belirleme işlemleri yapılmaktadır. İkinci kısımda ise proje kapsamında geliştirilen sezgisel yöntemle ilk üst sınır değeri belirlenmektedir. Üçüncü kısım ise budak açma, alt sınır belirleme ve budama işlemlerinin yapıldığı dal ve sınır algoritması işlemlerini kapsamaktadır. Dal ve sınır algoritmasının temel içeriği Şekil 5' de verilmiştir.



Şekil 5. Kesin çözüm algoritmasının temel içeriği

#### 4.2. Kritik Yol Yöntemi İş Programlaması

Kritik Yol Yöntemi (KYY) İş Programlaması kısmında öncelikle çözülecek problemin KYY'ye göre iş programı çizelgesi hazırlanarak, problemin çözümünde kullanılacak değişkenler kümesi oluşturulacaktır. KYY'ye göre hazırlanan iş programında kaynak dengelemesi yapabilmek için program içerisinde yer alan aktivitelerin toplam bollukları dahilinde başlama tarihleri değiştirilmelidir. Hangi aktivitelerin hangi zaman aralıklarında programlanabileceğini



belirlemek için öncelikle KYY iş programı çizelgesine ihtiyaç vardır. Problemin çözümü için algoritmada kullanılacak veri tablosu Şekil 6'da sunulmuştur.

no	süre	kaynak(k)	a	ardıl(a)	ö	öncel(ö)	EB	ET	GB	GT	TB	sa	sö
1													
2													
3													
4													
.													
n													

Şekil 6. Problem çözümünde kullanılacak veri tablosunun biçimi

no : aktivitelerin kimlik numarası

süre : aktivitelerin süresi

kaynak(k) : aktiviteye tahsis edilen türüne göre kaynak miktarı

k : kaynak türü

a : aktivitenin ardıl aktivite sayısı

ardıl(a) : aktivitenin ardıl aktiviteleri

ö : aktivitenin öncel aktivite sayısı

öncel(ö) : aktivitenin öncel aktiviteleri

EB : erken başlama

ET : erken bitiş

GB : geç başlama

GT : geç bitiş

TB : toplam bolluk

sa : serbest ardıl sayısı

sö : serbest öncel sayısı

n : toplam aktivite sayısı

Veri tablosu içeriğinin ilk beş kolonu (no, süre, kaynak, a, ardıl) kullanıcı tarafından girilecektir. Hazır problem setlerinin kolaylıkla kullanılabilmesi amacıyla, bu bilgiler algoritmaya bir *text* dosyasından otomatik olarak okutulabilmektedir. Veri tablosunun başlangıç görünümü Şekil 7' de sunulmuştur.

No	Süre	kaynak(k)	a	ardıl(a)	ö	öncel(ö)	EB	ET	GB	GT	TB	sa	sö
1	...	...	...	...									
2	...	...	...	...									
3	...	...	...	...									
4	...	...	...	...									
.	...	...	...	...									
.	...	...	...	...									
.	...	...	...	...									
n	...	...	...	...									

Şekil 7. Algoritmanın dosya okuma işleminden sonraki veri tablosunun görünümü

Veri tablosunun geri kalan içeriklerinden *öncel aktivite sayısı (ö)* ve *öncel aktiviteler (öncel(ö))* değişkenleri basit bir öncel belirleme işleminin ardından belirlenip tablo doldurulacaktır. Bu işlem salt aktivitelerin ardıllarına bakılarak öncelleri belirleme işlemi olup, dosya okuması aşamasında gerçekleştirilecektir. *Serbest ardıl sayısı (sa)* ve *serbest öncel sayısı (sö)* değişkenleri başlangıçta, ardıl sayısı (a) ve öncel sayısı (ö) değişkenlerin değerlerinin aynısına eşittir ( $sa=a$ ,  $sö=ö$ ). Bu değişkenler KYY iş programlama çizelgesini hesaplama esnasında dinamik değişken olarak kullanılacağı için tabloya dahil edilmişlerdir (Şekil 8). Tablo içeriğinde geriye kalan değişkenler KYY iş programlamasına ait bilgileri içermektedir. Dolayısıyla tablodaki bilgileri tamamlamak için KYY'ye göre iş programlaması yapılması gerekmektedir. KYY iş programlaması *ileri işlem* ve *geri işlem* olmak üzere iki algoritmayla yapılacaktır. İleri işlem algoritması aracılığıyla aktivitelere ait erken başlama (EB) ve erken bitiş (ET) zamanları hesaplanır. Geri işlem aracılığıyla aktivitelere ait geç başlama (GB) ve geç bitiş (GT) zamanları hesaplanır. TB, EB ve GB değerleri veya ET ve GT değerleri kullanılarak bulunabilir.

No	Süre	kaynak(k)	a	ardıl(a)	ö	öncel(ö)	EB	ET	GB	GT	TB	sa	sö
1	...	...	...	...	...	...						...	...
2	...	...	...	...	...	...						...	...
3	...	...	...	...	...	...						...	...
4	...	...	...	...	...	...						...	...
.	...	...	...	...	...	...						...	...
.	...	...	...	...	...	...						...	...
.	...	...	...	...	...	...						...	...
n	...	...	...	...	...	...						...	...

Şekil 8. Veri tablosunun KYY iş programlama işleminden önceki görünümü

#### 4.2.1. İleri İşlem

İleri işlem sırasında öncel sayısı sıfır olan aktiviteden başlanır, ki bu ilk aktivitedir. Bu aktivitenin EB tarihi sıfır olarak ayarlanır, ET tarihi de aktivitenin süresinin EB'ye eklenmesiyle ( $ET = EB + Süre$ ) bulunan tarih olarak ayarlanır. Böylece ilk aktivite programlanmış olur. Bu işlem tamamlandığında, söz konusu aktivitenin tüm ardıllarının serbest öncel sayıları 1 azaltılır. Bundan sonra programlanacak bir sonraki aktiviteye geçilir. Bu aktiviteyi seçerken yine öncel sayısı sıfır olan aktivite ele alınır. Eğer birden fazla serbest öncel aktivite sayısı sıfır olan aktivite varsa, o zaman aktivite numarası küçük olandan devam edilir. Bu aktivitenin EB tarihi, öncel aktivitelerinin ET tarihlerinden en geç olan olarak belirlenir. ET tarihi ise  $EB + Süre$  olarak belirlenir. Programlama bittikten sonra yine ardıl aktivitelerin serbest öncel sayısı bir azaltılır ve bu şekilde tüm aktiviteler programlanıncaya kadar aynı işlemler tekrar edilir. İleri işlemin sonucunda tüm aktivitelerin EB ve ET tarihleri bulunmuş olur. Projenin bitişini temsil eden son aktivitenin EB ve ET tarihleri aynı olup aynı zamanda projenin toplam süresine eşittir.

#### 4.2.2. Geri İşlem

Geri işlem ileri işlemin benzeri olup projenin bitiş tarihinden geriye doğru gidilerek yapılır. Burada serbest ardıl sayısının sıfır olduğu aktiviteden başlanır, ki bu başlangıçta son aktivitedir. Bu aktivitenin GT tarihi projenin bitiş tarihiyle aynı güne denk getirilir. GB tarihi ise GT tarihinden aktivite süresinin çıkarılmasıyla bulunur ( $GB = GT - Süre$ ). Bu aktivite programlandıktan sonra bu aktivitenin öncellerinin serbest ardıl sayıları 1 azaltılır ve bir sonraki aktiviteye geçilir. Burada serbest ardıl sayısı sıfır olan aktiviteden devam edilir. Eğer birden fazla serbest ardıl sayısı sıfır olan aktivite varsa aktivite numarası büyük olandan devam edilir. Bu şekilde seçilen sıradaki aktivite programlanırken GT tarihi ardıllarının en erken başlama tarihine denk getirilir. GB ise  $GT - Süre$  olarak bulunur. Programlama bittikten sonra öncellerinin serbest ardıl sayısı 1 azaltılır ve bu şekilde tüm aktiviteler programlanıncaya kadar aynı işlemler tekrar edilir. Geri işlemin sonucunda tüm aktivitelerin GB ve GT tarihleri bulunmuş olur. Geriye kalan aktivitelerin toplam bolluk (TB) süresi de  $TB = GB - EB$  ilişkisi kullanılarak hesaplanır.

KYY'ye göre proje programlaması yapıldıktan sonra KDP'nin çözümünde kullanılacak veri tablosu hazırlanmış olur. Bütün aktivitelerin serbest ardıl ve serbest öncel sayılarının programlama işlemi sonunda sıfır olmaları gerekmektedir. Bu tüm aktivitelerin erken ve geç



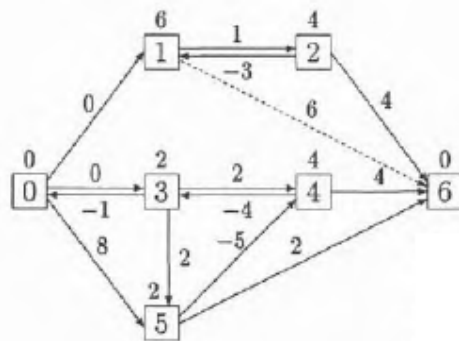
gerçekleşme tarihlerinin öncel ve ardıl aktiviteler arasındaki ilişkiye uyularak programlandığı anlamına gelmektedir (Şekil 9).

No	Süre	kaynak(k)	a	ardıl(a)	ö	öncel(ö)	EB	ET	GB	GT	TB	sa	sö
1	...	...	...	...	...	...	...	...	...	...	...	0	0
2	...	...	...	...	...	...	...	...	...	...	...	0	0
3	...	...	...	...	...	...	...	...	...	...	...	0	0
4	...	...	...	...	...	...	...	...	...	...	...	0	0
.	...	...	...	...	...	...	...	...	...	...	...	0	0
.	...	...	...	...	...	...	...	...	...	...	...	0	0
.	...	...	...	...	...	...	...	...	...	...	...	0	0
n	...	...	...	...	...	...	...	...	...	...	...	0	0

Şekil 9. Veri tablosunun KYY'ye göre programlama sonucundaki hali

### 4.3. Süre ilişkili programlama yöntemi (Temporal scheduling)

Kritik yol yönteminin daha genel hali olan süre ilişkili programlama (Temporal Scheduling) yönteminde aktiviteler arasında çift yönlü ilişkiler tanımlanabilmektedir. Buna göre, her aktivite kendisinden sonra gelen ardıl aktiviteyle başlama zamanı bakımından asgari ve azami süre ilişkisine sahip olabilir (Neumann v.d. 2003). Örneğin, bir ardıl aktivite kendisinden önce gelen öncül aktivite başladıktan asgari  $x$  gün sonra başlama koşuluna sahip olabileceği gibi azami  $y$  gün sonra başlama koşuluna da sahip olabilir. Burada,  $x$  sayısı ardıl aktivitenin öncül aktiviteyle olan asgari süre ilişkisini,  $y$  sayısı ardıl aktivitenin öncül aktiviteyle olan azami süre ilişkisini temsil eder.  $y$  sayısı, gösterim açısından öncülden ardıla doğru bir ok çekip ilişkiyi pozitif bir sayıyla göstermek yerine ardıldan öncüle bir ok çekip negatif sayıyla gösterilir. Böylece,  $x$  ve  $y$  ilişkilerinden hangisinin asgari ve azami süre ilişkisi olduğunun karışmaması sağlanır. Şekil 10' da verilen Neumann v.d. 2003'ten alınan örnekte olduğu gibi iki aktivite aynı anda birbirlerinin ardılı ve öncülü olabilir.



Şekil 10. Süre ilişkili programlama yöntemine göre hazırlanmış bir şebeke diyagramı

Literatürde kaynak dengeleme probleminin kesin çözümü için problem setlerinin kullanıldığı üç adet çalışma vardır (Neumann ve Zimmermann 2000 Gather v.d. 2010, Rieck v.d 2012) Bu çalışmalarda kullanılan problem setleri süre ilişkili programlama (Temporal Scheduling) yöntemine göre hazırlandıklarından proje kapsamında geliştirilen kesin yöntemlerde süre ilişkili programlama yöntemi de dahil edilmiştir.

#### 4.4. Kesin Çözüm İçin Geliştirilen Dal ve Sınır Algoritması

Kesin çözümü bulan dal ve sınır algoritmasını budaklardan dal açma stratejisi ve her bir budak için alt sınır hesaplama stratejisi olarak iki bölüme ayırabiliriz. Budaklardan dal açma stratejisi olarak henüz dal açılmamış budakların tümünden en iyi (en küçük) alt sınır değerine sahip budaktan dal açılması stratejisi kullanılmıştır. Alt sınır hesaplamasında, kritik yolda bulunan aktivitelere ait kaynaklar, kritik olmayan ve bir başlama tarihi belirlenmiş aktivitelere ait kaynaklar, ve kritik olmayan ve henüz programlanmamış aktivitelerin belirli günlerdeki (varsa) kaçınılmaz tükettiği kaynaklar, projenin kaynak kullanım profiline yerleştirildikten sonra geri kalan kaynakların izin verilen günlük azami kaynak kullanımı çerçevesinde amaç fonksiyonu (ölçüt) değerini en iyi verecek şekile yerleştirilmesi stratejisi kullanılmıştır. Bu bölümlere ait daha ayrıntılı bilgi ilerleyen bölümlerde verilmiştir.

##### 4.4.1. Algoritmada Kullanılan Bazı Tanımlar

**Budak:** Dal ve sınır algoritmasında kritik yolda olmayan aktivitelerden birinin programlandığı ve seçilen aktivite için belirlenen başlangıç tarihi bilgilerinin, hesaplanan sınır değerinin, kendisinden önceki budağın (anne budak) bilgisayar hafızasındaki fiziksel adresinin, kendisinden sonraki en iyi sınır değerine sahip budağın (sonraki en iyi budak) bilgisayar hafızasındaki fiziksel adresinin tutulduğu yapı birimidir.

**Kök Budak:** Dal ve sınır şemasının başlangıç budağı. Henüz hiç bir aktivitenin programlanmadığı, anne budağı olmayan budaktır.

**Ara Budak:** Dal ve sınır şemasında her hangi bir noktada, o noktada seçilmiş aktiviteye ait başlangıç tarihi olan, anne budağı olan ve budak oluşturma kapasitesi olan budaktır.

**Yaprak (Uç Budak):** Dal ve sınır şemasının en son budağıdır. Bu budağa gelince tüm aktivitelere ait bir başlangıç tarihi belirlenmiştir ve uygun bir iş programı elde edilmiştir.



**Anne Budak:** Kritik olmayan programlanmamış en az bir aktivitesi olan budaktır. Bu budak yeni budaklar oluşturma kapasitesine sahiptir. Yapraklar anne budak olamaz.

**Çocuk Budak:** Bir anne budaktan oluşturulan yeni budaklara verilen isimdir. Her bir çocuk budak kendisinin oluşturulduğu anne budağın adresini taşır.

**Başlama Günleri Haritası:** Algoritma işleyişi esnasında kullanılan, her hangi bir ara budaktan kök budağa ulaşana kadar işlenen tüm budakların tutmuş olduğu başlama tarihleri bilgisini anlık tutan hafıza birimidir. Programlanmış aktiviteler için belirlenmiş başlama tarihlerini, programlanmamış aktiviteler için boş değeri tutar.

**Kritik Kaynak Kullanım Profili:** Sadece kritik yolda olan aktivitelere ait kaynakların tahsisıyla elde edilen kaynak kullanım profilidir.

**Zemin Kaynak Kullanım Profili:** Kritik kaynak kullanım profili üzerine kritik olmayan ve başlama tarihi belirlenmiş aktivitelere ait kaynaklarla, kritik olmayan ve programlanmamış aktivitelere ait kaçınılmaz kaynakların eklenmesiyle elde edilen kaynak kullanım profilidir.

**Ara Kaynak Kullanım Profili:** Zemin kaynak kullanım profiline henüz tahsisi yapılmamış kaynakların amaç fonksiyonunu iyileştirecek şekilde yerleştirilmesiyle elde edilen kaynak kullanım profilidir.

**İzin Verilen Azami Günlük Kaynak Kullanım Profili:** Zemin kaynak kullanım profiline kritik olmayan ve programlanmamış aktivitelere ait henüz tahsisi yapılmayan kaynakların yerleşebileceği tüm günlerin eklenmesiyle elde edilen kaynak kullanım profilidir.

**Anne Budak Adayları Sırası:** Kendisinden yeni budaklar oluşturulabilecek tüm ara budakların (anne budakların) alt sınır değeri en iyi olandan en kötü olana (küçükten büyüğe) sıralanmasıyla elde edilen bir sıradır. Bir anne budaktan yeni budaklar oluşturulduktan sonra bu budak aday anne sırasından silinir. Sırada bulunan bir sonraki anne budak bu sıranın ilk elemanı olur. Anne budaktan oluşturulan her yeni çocuk budak alt sınır değerine göre bir sonraki çocuk budak oluşturma işlem için aday anne budak sırasına yerleştirilir.

**Budak Cinsiyeti:** Budakta programlanan kritik olmayan aktivitenin numarasıdır. Kök budağın cinsiyeti boştur.



**Üst Sınır:** Dal ve sınır algoritmasına başlamadan önce bilinen en iyi iş programına ait amaç fonksiyonu değeridir. Kesin çözüm algoritmasına başlamadan önce sezgisel yöntemle bulunacaktır.

**Üst Sınır Değeri Güncellemesi:** Budak oluşturma aşaması esnasında yeni oluşturulan budakların yaprak olması durumunda, bu yaprakların amaç fonksiyonu değeri var olan üst sınır değerinden daha iyi ise üst sınır değeri güncellenir ve en amaç fonksiyonu değerine sahip olan yaprak elde edilen en iyi çözüm olarak hafızaya kaydedilir.

**Alt Sınır:** Dal ve sınır algoritması uygulama esnasında oluşturulacak budaklarda hesaplanan ara kaynak kullanım profillerinin amaç fonksiyonuna göre değeridir. Ara kaynak kullanım profili, tahsisi yapılmamış kaynakları yerleştirirken amaç fonksiyonunu hep iyileştirecek şekilde yerleştirdiğinden bir anne budaktan oluşturulacak yaprakların amaç fonksiyonu değeri anne budağın alt sınır değerinden daha iyi olamaz. Yine alt sınır değeri hesaplama mantığı gereğince kök budaktan yaprağa ulaşana kadar alt sınır değeri eşit kalabilir veya artış gösterebilir ancak azalamaz.

**Budama:** Bir budağın alt sınır değeri, önceden belirlenmiş üst sınır değerine göre eşitse veya daha büyükse, bu budaktan oluşacak bir yaprağa ulaşana kadar bu dizinin var olan üst sınır değerinden (bilinen en iyi çözümden) daha iyi bir çözüm sunma kabiliyetini yitirmiştir. Bu budağın aday anne budak sırasına alınmayıp hafızadan silinme işlemine budama denir. Böylece, bu budaktan oluşturulabilecek muhtemel budaklar ve yapraklar da şematik olarak budanmış olur. Budama ayrıca, üst sınır değeri güncellemesi yapıldıktan sonra aday anne sırası boyunca güncel üst sınır değerinin üstünde alt sınır değerine sahip budakların sıradan alınması ve hafızadan silinmesi işlemini de kapsar.

#### 4.4.2. Algoritmada Kullanılan Bazı Fonksiyonlar

**Ara Profil Bulucu:** Her hangi bir budağın oluşturulma işlemi sırasında kendisi ve kendisinden önceki (anne) budakların bilgilerinin okunmasıyla elde edilen başlama günleri haritasına göre kaynak kullanım profilini oluşturan fonksiyondur. İki bölümden oluşur ve her üç amaç fonksiyonu için ayrıca oluşturulmuştur. Fonksiyonun birinci bölümü tüm amaç fonksiyonları için ortaktır ve kritik olan aktivitelere ait kaynakları, kritik olmayan ve başlama süreleri belirlenmiş aktivitelere ait kaynakları, kritik olmayan ve programlanmamış aktivitelere ait kaçınılmaz kaynakları hesaba katarak zemin kaynak kullanım profilini oluşturur. Fonksiyonun ikinci bölümü, çözüm için seçilen amaç fonksiyonuna göre en iyi (bu



fonksiyonlar için en küçük) sonucu verecek şekilde kritik olmayan ve programlanmamış aktivitelere ait henüz yerleştirilmemiş kaynakların izin verilen azami günlük kaynak kullanımı kısıtlamasına göre profil üzerinde dağıtılmasını sağlar. Böylece, işlem yapılan budağa ait en iyi alt sınır değerini verecek kaynak profili oluşturulmuş olur.

**Azami Profil Bulucu:** Herhangi bir budağın oluşturulma işlemi esnasında ara profil bulucu fonksiyonun içerisinde çağrılır ve izin verilen azami günlük kaynak kullanım profilinin hesaplamasında kullanılır. Kritik olmayan ve programlanmamış aktivitelere ait kaynakların tahsis edilebileceği tüm günler zemin kaynak profili üzerine eklenerek hesaplanır.

**Alt Sınır Değeri Hesaplayıcı:** Seçilen amaç fonksiyonuna göre ara kaynak kullanım profili elde edilen budağa ait amaç fonksiyonu değerini hesaplar.

**Aday Anne Budak Sıralayıcı:** Oluşturulan her yeni budağa ait hesaplanan alt sınır değerine göre anne budak adayları sırasını tarayıp yeni budağı bu sıra içinde uygun yere yerleştiren fonksiyondur. Yeni budakların bu sıraya yerleştirilmesi için bir yerleşim kriteri kullanılır. Yerleşim kriteri baştan sona doğru:

- En iyi (en küçük) alt sınır değerine sahip budak,
- En çok kritik olmayan programlanmış aktiviteye sahip budak,
- İşlem sırasında sonra gelen budak önceliklerini içermektedir.

Buna göre bir yerleşim kriteri iki budak için eşitse bir sonraki kriter gere göre yerleşim yapılmaktadır. Bu kriter, algoritmayı kesin çözüme ulaştıracak en muhtemel aday anne budaklarını sıranın en başına koymaktadır.

**Budak Cinsiyet Belirleyici:** Aday anne budaktan oluşturulacak çocuk budakların cinsiyetini belirleyen fonksiyondur. Yani, anne budaktan çocuk budaklar oluşturulurken yeni oluşturulacak budaklarda hangi aktivitenin programlanacağını belirleyen fonksiyondur. Fonksiyon budak cinsiyetini, kritik olmayan programlanmamış aktiviteler arasından bir seçim kriterine göre belirlemektedir. Seçim kriteri:

- En yüksek (aktivite kaynak ihtiyacı)\*(kaynak tahsisi yapılmamış gün sayısı) değeri
- En yüksek (aktivite kaynak ihtiyacı)
- İşlem sırasında sonra gelen aktivite önceliklerini içermektedir.

Buna göre bir seçim kriteri iki aktivite için eşitse bir sonraki kriter gere göre seçim yapılmaktadır. Bu kriterin uygulanılmasındaki amaç, alt sınır değeri hesaplaması esnasında amaç fonksiyonunu en çok etkileme kabiliyetine sahip aktivitelere öncelik vermektir. Böylece, bu budaktan yapraklara doğru devam ederken alt sınırın daha erken büyümesi sağlanacak



ve olası budama işlemleri öne çekilerek kesin çözüm vaad etmeyen budakların budaması erken yapılabilecektir.

**Yeni Oluşturulacak Budak Sayısı ve Budak Başlama Tarihleri Belirleyici:** Bu fonksiyon anne budaktan yeni budaklar oluşturulurken, kaç çocuk budak oluşturulacağını ve bu budaklarda programlanacak aktivitenin hangi tarihlere programlanacağını belirler. Bunun için, anne budağa ait başlama tarihlerinin haritası çıkartılır. Bu harita kullanılarak programlanmış aktiviteler sabitleşmiş kabul edilir. Sadece programlanmış aktiviteler üzerinde bir KYY ileri işlemi ve geri işlemi uygulanarak bu aktivitelere ait erken başlama, erken bitme ve geç başlama, geç bitme tarihleri güncellenir. Yeni oluşturulacak budaklarda programlanacak aktiviteye ait erken başlama tarihi ve bolluk süresi kayda alınır. Bu aktiviteye ait güncel bolluk süresinin bir fazlası yeni oluşturulacak çocuk budak sayısını belirler. Her bir çocuk budak için erken başlama tarihine sıfırdan başlayarak kullanılan bolluk günleri eklenerek programlanacak aktiviteye ait başlama tarihi belirlenir.

#### 4.4.3. Dal ve Sınır Algoritmasının Şematik Anlatımı

Dal ve sınır algoritmasının başlangıcı kök budaktır. İşlemler sonucunda kök budaktan yeni budaklar üretilerek yapraklara ulaşılmaya çalışılır. Yapraklar tüm aktivetelerin programlandığı uygun bir iş programına çözüm değerine sahiptir. Çoğunlukla ulaşılan ilk yaprak kesin çözüm olmaktadır. Ancak bu durum her zaman geçerli değildir. Şekil 11’de algoritmanın işleyişi esnasında kesin çözüme nasıl ulaşıldığı Easa (1989)’a ait problemle şematik olarak açıklanmaktadır. Şekil 11’ de Easa (1989)’a ait bir problem mutlak sapma miktarı ölçütü için geliştirilen dal ve sınır algoritması kullanılarak elde edilmiş çözümü verilmiştir. Easa’nın problemine ait KYY’ye göre iş programı bilgileri Şekil 12’de verilmiştir. 1. ve 7. aktiviteler projenin başlama ve bitiş aktiviteleridir. Şekil 12’deki tabloda sütunlar sırasıyla aktivitelere ait kimlik, süre, kaynak ihtiyacı, ardıl sayısı, ardıllar, öncel sayısı, önceller, erken başlama, erken bitirme, programlanan başlama (başlangıçta erken başlamaya eşit), programlanan bitirme (başlangıçta erken bitirmeye eşit), geç başlama, geç bitirme tarihleri ve son olarak bolluk süresi verilmiştir. Görüldüğü gibi 4, 5 ve 6 numaralı aktiviteler bolluk süreleri sıfırdan farklı olan yani kritik olmayan aktivitelerdir. Şekil 12’deki KYY bilgilerine göre 4, 5 ve 6 numaralı aktiviteler bolluk günleri içerisinde programlanarak amaç fonksiyonuna göre en iyi kaynak kullanım profilini veren iş programı bulunmaya çalışılır. Bu projede geliştirilen dal ve sınır algoritmasının Easa (1989)’nın örnek problemi çözümünde nasıl işlediğini aşağıda anlatılmıştır.

Own Address		
Mom		Next Best
ID	SD	lb

Şemada kullanılan değerlerin açıklamaları:

Own address : Budağın kendi adresi

Mom : Anne budak adresi

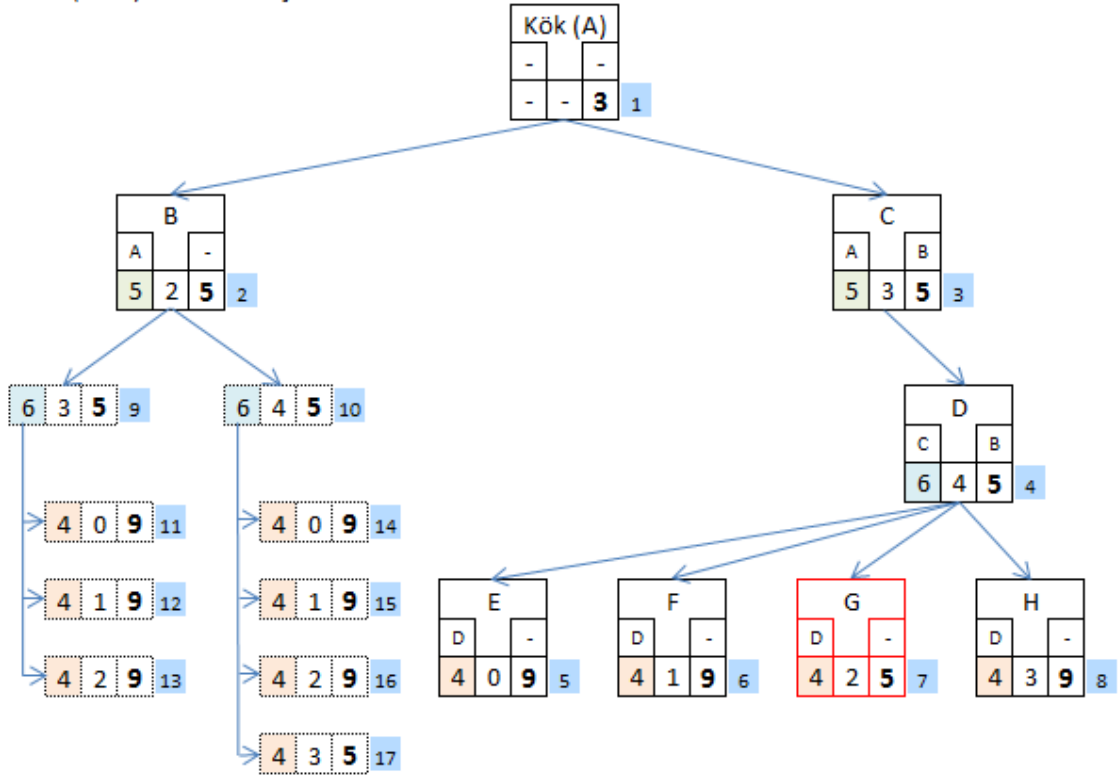
Next Best : Bir sonraki en iyi aday anne budak adresi

ID (Identity) : Budakta programlanan aktivite kimliği

SD (Start Date) : Budakta programlanan aktivitenin başlama tarihi

lb (lower bound) : Budağa ait kısmi iş programının ara kaynak kullanım profiline ait alt sınır değeri

Easa (1989) Dal ve Sınır Şeması



Şekil 11. Easa (1989)'a ait problemin dal ve sınır algoritmasıyla mutlak sapma miktarının en aza indirilmesi amaç fonksiyonuna göre kesin çözümünün bulunuşu

```

===== CPM Scheduling Results Are Shown Here =====
A Single Resource Project with 7 Activities
ID DUR Resources S Successors P Predecessors ES EF SD PD LS LF TF
-----
1 0 0 2 2 4 0 0 0 0 0 0 0
2 2 10 2 3 5 1 1 0 2 0 2 0
3 3 6 1 7 2 1 2 2 5 2 5 0
4 1 2 1 6 1 1 0 1 0 1 3 4 3
5 1 4 1 6 1 2 2 3 2 3 3 4 1
6 1 2 1 7 2 4 5 3 4 3 4 4 5 1
7 0 0 0 2 3 6 2 3 5 5 5 5 0

<<< CPM Duration: 5 >>>

Start Activity: 1
Finish Activity: 7
Critical Activities: 1 2 3 7
Non-critical Activities: 4 5 6

```

Şekil 12. Easa (1989)'a ait problemin KYY iş programı bilgileri

**1. Adım:** Sezgisel yöntemle üst sınır değeri belirleme – Geliştirilen sezgisel yöntemle başlangıç için üst sınır değeri 5 olarak bulunmuştur. Bu değere Easa (1989) problemi için aynı zamanda kesin çözümün değeridir. Burada problem oldukça küçük çapta ve basit olduğu için sezgisel yöntemin kesin çözümü bulması olağandır.

**2. Adım:** Kök budağın oluşturulması – Dal ve sınır algoritmasının en başında kök budak oluşturulur. Bu budakta tüm elemanların bilgileri boş bırakılır (anne budak yoktur, bir sonraki en iyi budak yoktur, henüz hiç bir aktivite programlanmamıştır). Ancak, bu budağa ait ara kaynak kullanım profili oluşturulabileceğinden bu profile ait alt sınır değeri bulunur ve hafızaya işlenir. Kök budak elde edildikten sonra, eldeki en iyi budak kök budaktır (Bakınız Şekil 12, A kodlu/1 numaralı budak). Aday anne budak zincirinde tek elemandır ve bir sonraki dal açma işleminde aday anne budak olarak kullanılacaktır.

**3. Adım:** Aday anne budaktan (bu adım için A budağı) oluşturulacak yeni budakların cinsiyetinin belirlenmesi – Budak cinsiyet belirleyici fonksiyon çağrılır ve 4, 5, 6 numaralı aktiviteler arasından seçim kriterine göre budak cinsiyeti belirlenir. Buna göre, 5 numaralı aktivite seçim kriterine göre en yüksek önceliğe sahiptir ve budak cinsiyeti olarak belirlenir.

**4. Adım:** Oluşturulacak çocuk budak sayısı ve başlama tarihleri belirleme – Burada yeni oluşturulacak budak sayısı ve başlama tarihleri belirleyici fonksiyon çağırılarak, 5. aktiviteye





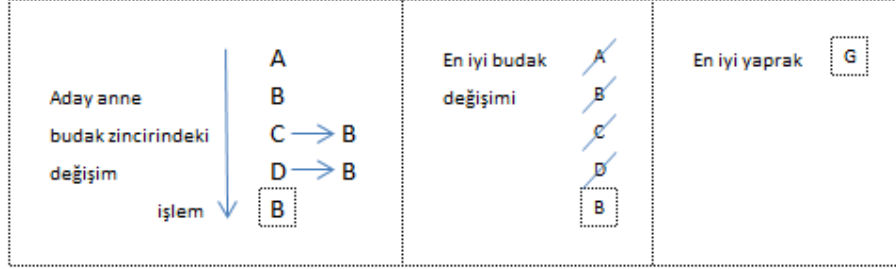
ait bolluk sayısı ve erken başlama tarihleri güncellenir (bu adım için bu aktivitenin bilgilerinde değişiklik olmamıştır). Bu güncellemeye göre 5. aktivitenin 1 bolluk günü bulunmaktadır, dolayısı ile 2 adet çocuk budak oluşturulabilir. Başlama tarihleri ise EB+0 ve EB+1 şeklinde (5 için erken başlama 2. günden sonradır) 2. ve 3. günlerdir.

**5. Adım:** Çocuk budak işlemleri – B kodlu budağa ait ara kaynak kullanım profili bulunur. Bu profile göre amaç fonksiyonun değeri hesaplanır. Bulunan değer budağın ilgili elemanın değeri olarak yazılır. B kodlu budağın annesi A kodlu budaktır. Bu adres ilgili elemanın değeri olarak yazılır. A kodlu budaktan yeni budaklar oluşturulduğu için artık aday anne budak zincirinden çıkartılır. B budağı tek budak olduğu için zincirin tek elemanı olur. Kendisinden sonra en iyi hiç bir budak yoktur, *next best* adresi boştur, ve en iyi budaktır (Şekil 13). C kodlu budağa ait ara kaynak kullanım profili bulunur. Bu profile göre amaç fonksiyonu değeri hesaplanarak hafızaya alınır. C budağı budak yerleşim kriterine göre aday anne budak zincirine yerleştirilir. B artık en iyi budak değildir, en iyi budak C olur. C'den sonra en iyi budak adresine B'nin adresi yazılır.

**6. Adım:** C'den çocuk budak üretme işlemi – Yeni oluşturulacak çocuk budakların cinsiyeti fonksiyon çağırılarak belirlenir. C için çocuk budakların cinsiyeti 6 numaralı aktivitedir. Bu aktivitenin erken başlama ve bolluk günleri anne budakların programlanmış başlama günlerine göre güncellenir. Şekil 11'de görüldüğü gibi 6 numaralı aktivitenin bolluk süresi 5. aktivitenin 3. güne programlanmış olması sebebiyle 0'a inmiştir. Dolayısıyla sadece bir çocuk budak oluşturulacaktır. Oluşturulan budağın başlama tarihi hafızaya işlenir, annesinin adresi olarak olarak C budağı yazılır. Ara kaynak profili bulunur, alt sınır değeri hesaplanır. Bu değere göre daha önce C->B şeklinde olan zincire bu budak yerleştirilir. Bu arada C budağı aday anne budak zincirinden çıkarıldığı için zincirin yeni şekli D->B şeklinde olur. En iyi budak artık D'dir.

**7. Adım:** D'den çocuk budak üretme işlemi – Yeni oluşturulacak çocuk budakların cinsiyeti belirlenir. Çocuk budakların cinsiyeti 4 numaralı aktivitedir. Bu aktivitenin erken başlama tarihi ve bolluk süresi güncellenerek çocuk budak sayısı ve çocuk budakların başlama tarihleri belirlenir. 4. aktivitenin bu adımda programlanmasıyla oluşturulacak budaklar artık yapraklılar. D budağından yeni budaklar oluşturulduğu için aday anne budak zinciri sırasından çıkarılır. Oluşturulan E, F, G, ve H yapraklarına ait ara kaynak profilleri ve bu profillere göre amaç fonksiyonu değerleri bulunur. E ve F yaprakları üst sınır değerinden daha büyük alt sınır değerlerine sahip oldukları için silinirler. G yaprağı alt sınır değeri üst sınır değerine eşittir. Bunun için bu yaprak en iyi yaprak olarak hafızaya alınır. H yaprağı da

aynı şekilde üst sınırdan daha kötü bir alt sınıra sahip olduğu için silinir. G yaprağı anne budak olamayacağından zincire alınmaz. Zincirde sadece B budağı kalmıştır. B budağının alt sınır değerine bakılarak daha iyi sonuç üretebilme yeteneği kontrol edilir. Burada B budağı alt sınır değeri 5'tir. Yani en iyi ihtimalle G ile aynı kalitede bir sonuç üretebilir. Algoritma amacı bütün optimal çözümleri bulmak değildir, sadece bir optimal çözüm bulmaktır. Dolayısıyla algoritma burada durdurularak G yaprağının sonucu kesin çözüm sonucu olarak alınır.

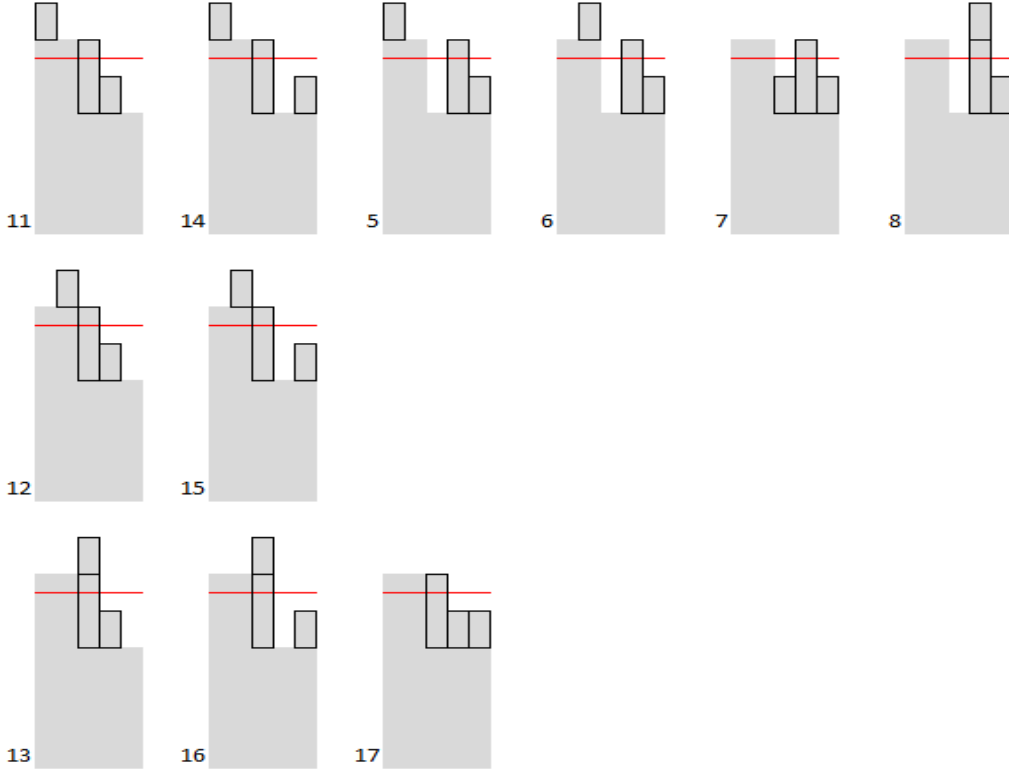


Şekil 13. Dal ve sınır algoritması işlemi süresince aday anne budak zinciri, en iyi budak ve en iyi yaprak elemanları içerisindeki değişim

Şekil 11'de ayrıca, B budağından yeni budak oluşturmaya devam edilseydi elde edilecek sonuçlar gösterilmiştir. 9 numaralı budaktan 17 numaralı yaprağa gelene kadar sadece bir tane kesin çözüme eşit sonuç bulunmuştur. Bu projedeki kesin çözümü bulmakla ilgili olan kısmın asıl amacı elde edilen kesin sonuçların üst-sezgisel yöntemlerin performansını değerlendirmede kullanılmasıdır. Bundan dolayı optimal çözümü veren sadece bir iş programının bulunması yeterlidir. B budağından devam ederek çözüm kümesinin diğer elemanlarına bakmaya gerek yoktur. 9'dan 17'ye kadar olan budakları oluşturmaya gerek yoktur. Problemin çözüm kümesi kısaltılarak kesin çözüme ulaşılmıştır.

Şekil 14a ve 14b'de görülen grafikler yukarıdaki dal ve budak şemasıyla çözümü gösterilen Easa (1989)'a ait her bir budağın ara kaynak kullanım profilidir. Kırmızı renkli doğru parçası hedeflenen kaynak kullanım miktarını göstermektedir. Kenarlıksız düz gri renkli alan kritik aktivitelere ait kaynak kullanımını göstermektedir. Kenarlıklı gri bloklar profilin ait olduğu budağa kadarki programlanan aktivitelerin kaynaklarına aittir ve ilgili aktivitelerin hangi günlere programlandığını da göstermektedir. Kırmızı renkli kareler kritik olmayan ve programlanmamış aktivitelere ait kaynakların amaç fonksiyonuna göre en iyi sonuç verecek şekilde dağıtımını göstermektedir. Bu şekilde alt sınır değeri hesaplamasının mutlak sapma miktarının en aza indirgenmesi amaç fonksiyonuna göre nasıl hesaplandığı görülebilir.



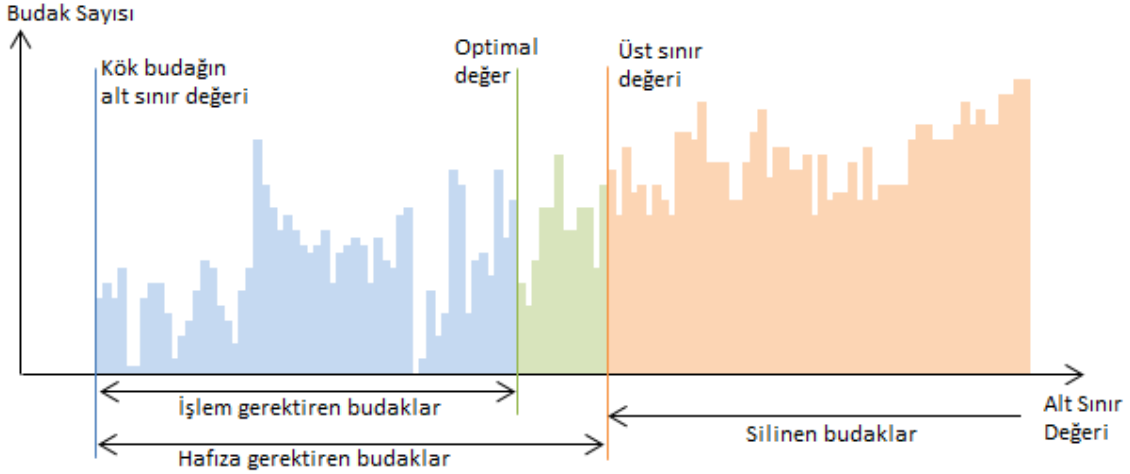


Şekil 14b. Dal budak şemasına ait ara kaynak kullanım profilleri (Kısım 2)

#### 4.4.4. Dal ve Sınır Algoritmasının Performansının Artırılması için İzlenen Stratejiler

Geliştirilen dal ve sınır yöntemiyle elde edilen çözüm incelendiğinde, çözüme ulaşmak için bir başlangıç noktasının (kök budak), çözüm kümesini oluşturan bir kümenin (yapraklar) ve çözüm kümesine ulaşırken kullanılan ara çözüm paketlerinin (budaklar) varlığından bahsedebiliriz (Bakınız Şekil 11). Tipik bir 30 aktivitelik, ortalama bolluk süreleri 30-40 gün olan problemde elde edilebilecek yaprak sayısı  $10^{40}$  gibi oldukça büyük bir sayı mertebesindedir. Dal ve sınır yönteminde kullanılan sınırlama (alt sınır) metoduyla ara çözüm paketleri henüz oluşturulurken hesaplanan alt sınır değerlerine göre dallardan çözüm vaat etmeyenler budanarak hedef çözüm kümesi küçültülmeye çalışılır. Kesin çözüme ulaşabilmek için, ya da diğer bir deyişle açılan ilk yapraklarda elde edilen çözümün kesin çözüm olduğunu söyleyebilmek için, anne budak adayları sırasında bulunan tüm budakların alt sınır değerlerinin yapraklarda bulunan en iyi değerden büyük veya eşit olması gerekmektedir. Bu durum, anne budak adaylarından hiç birinin artık daha iyi bir sonuç üretemeyeceğini, dolayısıyla elde bulunan en iyi yaprağın kesin çözüm olduğunu ifade eder.

Dal sisteminin çözümü esnasında üretilen budakların her biri için alt sınır değeri hesaplama fonksiyonu çalıştırılmaktadır, ki bu fonksiyon görece en uzun işlem gerektiren fonksiyonlar arasındadır. Bu da, üretilen budak sayısının alt sınır hesaplama fonksiyonun toplam çağırılıma sayısını ve dolayısıyla algoritma toplam süresini belirlemede en önemli etken olduğunu göstermektedir. Şekil 15’de tipik bir problemin çözümü esnasında üretilen budak sayısının alt sınır değerlerine göre dağılımını göstermektedir. Burada görülebileceği gibi, en düşük alt sınır değerine sahip budak kök budaktır. Yeni budaklar oluşturuldukça alt sınır değeri artmaktadır. Üst sınır değerinden daha büyük veya eşit alt sınır değerine sahip olan budaklar doğrudan silinmektedir. Optimal değerden daha büyük alt sınır değerine sahip olan budaklar çözüm esnasında optimal değer bilinmediğinden hafızada tutulmakta, ancak optimal çözüm bulunduktan sonra işleme alınmadığından performans açısından süreye bir etkisi olmayıp sadece hafıza kullanımını artırmaktadır. Son olarak, alt sınır değeri optimal değer altında kalan tüm budaklar işleme alınmak (yeni budaklar oluşturmak) zorundadırlar. Dolayısıyla, algoritmanın toplam işlem süresini alt sınır değeri optimal değer altında kalan budakların sayısı, toplam hafıza kullanımını ise alt sınır değeri üst sınır değerinin altında kalan budakların sayısı birincil derecede belirlemektedir.



Şekil 15. İşlem ve hafıza gerektiren budakların alt sınır değerine göre dağılımı



Buradan yola çıkarak performans arttırımı için yapılabilecek üç temel yöntem belirlenmiştir:

- Hafıza performansını arttırmak için ilk üst sınır değerini kesin çözüme yaklaştırma
- İşlem süresi ve hafıza performansını arttırmak için alt sınır değerini olabildiğince büyük sonuç verecek şekilde yapılandırma
- İşlem süresi performansını arttırmak için alt sınır değeri fonksiyonu ve diğer uzun süre işlem yapan fonksiyonları basit ve sade olarak (hızlı çalışacak şekilde) tasarlama

İleriki bölümlerde bu yaklaşım çerçevesinde yapılan geliştirmelere ayrıntılı olarak değinilecektir.

#### **4.4.5. Sezgisel Yöntemle İlk Üst Sınır Değeri Belirlemesi**

Dal ve sınır yönteminde başlangıç üst sınır değeri özellikle hafıza kullanımı itibariyle son derece önemlidir. Bu anlamda, dal ve sınır hesaplamalarına başlamadan önce probleme iyi bir üst sınır değeri belirlemek, işlem için hafızada tutulan ara çözüm paketlerinin (budak) kümesini daraltarak kullanılacak hafıza miktarını azaltmaktadır. Basit ama başarılı bir sezgiselle KDP çözümlenerek dal ve sınır algoritması için iyi bir üst sınır değeri hesaplanabilir. Bu proje dahilinde dal ve sınır algoritmasından esinlenilerek yeni bir sezgisel yöntem geliştirilmiştir. Bu bölümde bu yöntemden bahsedilecektir.

Sezgisel algoritmada açılan her bir yeni budakta kesin çözüm algoritmasında olduğu gibi bir sınır değeri hesaplaması yapılmaktadır. Bu hesaplama, kesin yöntemdekinden farklı olarak boştan doldurmalı kaynak profiline göre hesaplanmaktadır. Programlaması yapılan aktiviteye ait kaynaklar, kaynak kullanım profilinde yerleştirdikten sonra henüz programlaması yapılmamış aktivitelere ait kaynakların profil üzerindeki olası etkileri eklenerek elde edilen profilin ölçütüne göre değeri sınır değeri olarak alınmaktadır. Programlaması yapılmamış aktivitelere ait kaynakların profil üzerindeki olası etkileri, aktivitenin toplam çalışma süresince kullanacağı kaynakların (süre\*kaynak), aktivitenin aktif olabileceği toplam süre aralığına (erken başlama tarihinden en geç tamamlanabileceği tarihe kadar olan süre) bölünüp bu süre içerisindeki günlere eklenmesiyle hesaplanmaktadır. Böylece, henüz programlaması yapılmamış aktivitelere ait kaynakların, programlaması yapılmakta olan aktivitenin bolluk süresi içerisindeki seçeneklerine olan etkisinin sınır değeri hesabına katkısı olabildiğince göz önünde bulundurulmaya çalışılmaktadır. Algoritma başlangıcında bolluğu bulunmayan aktiviteler (kritik aktiviteler) belirlenerek bu aktivitelere ait kaynaklar en baştan programlanarak bir kritik kaynak kullanım profili elde edilmektedir. Daha sonra bollukları olan

aktiviteler bir sıraya konularak, bu sıraya göre sırası gelen aktiviteye ait kaynaklar boşluklara yerleştirilerek ve henüz sırası gelmeyen aktivitelerin olası etkileri kaynak profiline eklenerek iyi bir kaynak kullanım profili oluşturulmaya çalışılmaktadır. Bu biçimde sınır hesaplaması yapılarak en son aktivite de programlandığında elde edilen uygun bir iş programıyla iyi bir sonuç elde edilmesi hedeflenmektedir. Oluşturulan bu profil üzerinde kaynakların hali hazırda sahip oldukları bolluk günleri içerisinde yerleri değiştirilerek bu profil iyileştirilmeye çalışılmaktadır. Bunun için bir kaç ek dal ve sınır oluşturma işlemi daha uygulanmaktadır. İşlem sırasındaki her bir ara budak kullanılabilir bir iş programına sahiptir. Yapılan işlem, önceden belirlenmiş olan aynı aktivite sıralamasına göre, aktivitelerin sırayla diğer aktiviteler programlanmış iken sahip olduğu bolluk süreleri içerisinde denenerek daha iyi sonuç verebileceği başlama sürelerinin belirlenmesidir.

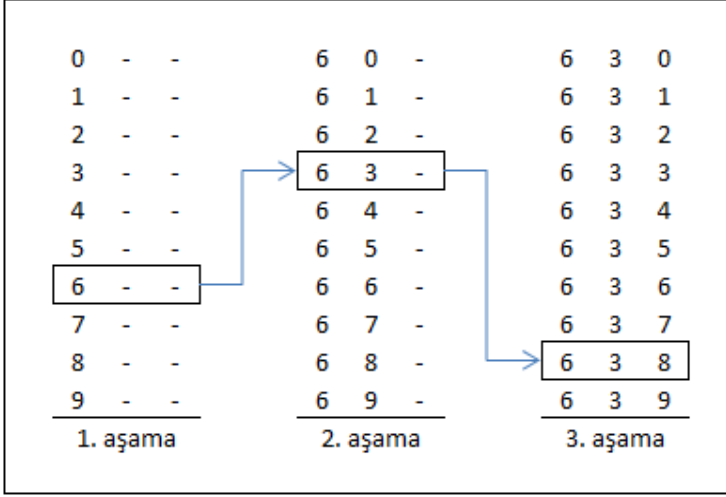
Bollukları bulunan aktivitelerin diziliş sırasının çözüm sonucunu farklı biçimlerde etkilediği görülerek, sezgisel yöntemle aktivite sıralayıcı sistem eklenmiştir. Buna göre, aktiviteler öncelikle 10 ayrı kritere göre sıralanarak her bir kriter için problem sezgiselle çözdürülmektedir. Daha sonra, en iyi sonucu veren sıralama kriteri sabit tutularak bu kritere göre eşit önceliğe sahip aktiviteler arasında ikinci bir 10 kriter sıralaması yapıp problem tekrar sezgiselle çözdürülmektedir. Yine aynı şekilde, bu aşamada en iyi sonucu veren kriter sabitlenerek üçüncü bir 10 kriter sıralaması yapılmakta elde edilen en iyi sonuç üst sınır değeri olarak aktarılmaktadır (Şekil 16). Burada, belirtilen 10 kriter, aktivitelerin;

- (toplam süre)\*(kaynak),
- (kaynak yerleştirilmemiş süre)\*(kaynak),
- (kaynak),
- (bolluk süresi),
- (aynı günlere denk gelen aktivite kalabalığı)

kriterlerine göre artan veya azalan biçiminde sıralanmasıdır.

Bazı aktiviteler birinci aşamaya göre sıralanırken aynı değere sahip olanları kendi aralarında sıralamak için ikinci, daha sonra üçüncü aşama uygulanmıştır. En son hala eşit değere sahip olan aktiviteler mevcut ise işlem esnasında önce gelen aktiviteye öncelik verilmektedir. Bu yöntemle, her problem için en iyi sonuç verecek sıralama kriteri algoritma tarafından belirlenmektedir.

Yapılan pilot denemelerde hesaplanan üst sınır değerleri genellikle kesin çözüme yakın olmuştur. Kesin çözüm algoritmasının küçük bir bölümünü oluşturan dal ve sınır sezgiseli olabildiğince basit ve sade tutulmuştur. Böylece, üst sınır değeri hesabının süre olarak kesin çözüm algoritması içindeki payı önemsenmeyecek kadar küçük olmuştur. Üst sınır değeri



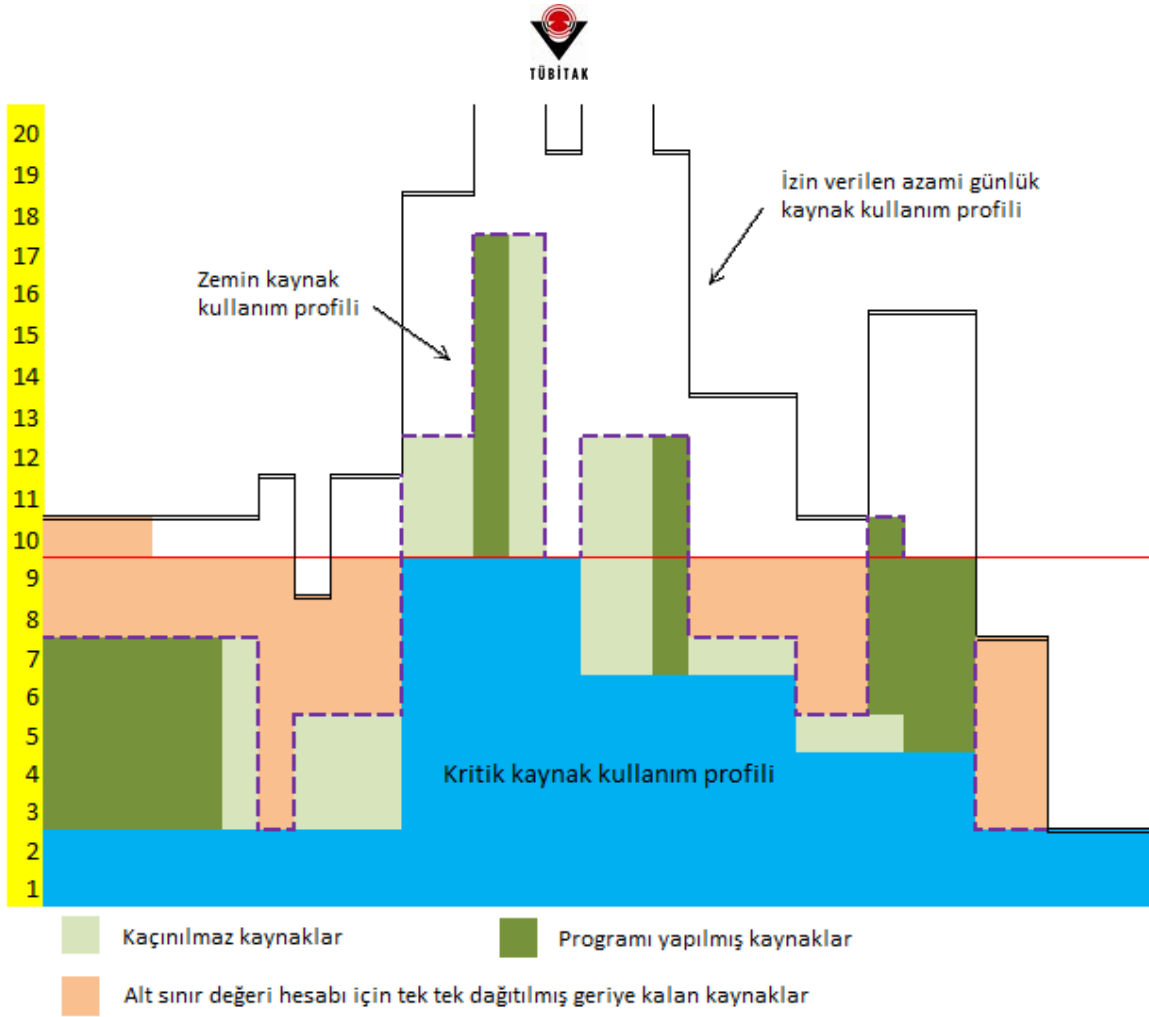
Şekil 16. Üst sınır değeri hesaplaması için en iyi aktivite sıralama kriterinin seçimi

hesaplamasında yapılabilecek daha ileri geliştirmeler mevcut olmakla birlikte, üst sınır değeri yaklaşımıyla algoritmanın hafıza performansının iyileştirilmesinde nispeten doyuma ulaşılmıştır.

#### 4.4.6. Alt Sınır Değeri Hesaplamasında Yapılan İyileştirmeler

Kesin çözüme ulaşana kadar işleme alınacak budak sayısını azaltmak için budaklarda hesaplanan alt sınır değerinin olabildiğince büyük olması gerekmektedir. Algoritmada kullanılan alt sınır değeri hesabı, kritik aktiviteler ve kritik olmayan programlanmış aktivitelere ait kaynaklar ve kaçınılmaz kaynaklar tahsis edildikten sonra, geriye kalan yerleştirilmemiş kaynakların izin verilen azami günlük kaynak kullanım miktarını aşmadan, zemin kaynak kullanım profili üzerinde en iyi profili verecek şekilde tek tek dağıtılmasıyla yapılır. Şekil 17’de görüldüğü gibi kaçınılmaz kaynaklarla, programı yapılmış kaynakların eklenmesiyle elde edilen profil zemin kaynak kullanım profilidir (Mavi, koyu yeşil ve açık yeşil bölge). İzin verilen azami kaynak kullanım profilini aşmamak kaydı ile programı yapılmamış geriye kalan kaynakların zemin kaynak kullanım profili üzerinde alt sınır değerini en iyi verecek şekilde tek tek dağıttıktan sonra elde edilen profil ara kaynak kullanım profilidir (Mavi, koyu yeşil, açık yeşil ve turuncu bölge). Proje kapsamında alt sınır hesaplanması için Neumann ve Zimmermann (2000) önerdiği alt sınır hesaplama yöntemine ilaveten kaynak tahditlerinin dikkate alındığı ve sürekli güncellendiği dinamik bir alt sınır yöntemi geliştirilmiştir. Ancak bu yöntemle elde edilen alt sınır değerlerinin algoritma performansında istenilen iyileştirmenin yapılabilmesi için çoğu zaman yeterince dar bir aralıkta olmadığı gözlemlenmiştir. Bu sebeple





Şekil 17. Bir budağın alt sınır değeri hesabı aşamasında ara kaynak kullanım profilinin oluşumu

daha dar bir aralıkta değişim gösterecek alt sınır değeri hesabı üzerinde çalışılmış ve yeni bir alt sınır hesap yöntemi bulunmuştur.

Yeni alt sınır değeri hesabında, bir önceki yöntem gibi kritik aktiviteler ve kritik olmayan programlanmış aktivitelere ait kaynaklar ve kaçınılmaz kaynaklar tahsis edilmektedir. Yani, zemin kaynak kullanım profili her iki hesap yönteminde aynıdır. Geriye kalan henüz tahsisi yapılmamış kaynakların dağıtılması ile ilgili farklı bir yola başvurulmuştur. Burada, programlaması yapılmamış her aktivite kendi bolluk günleri içerisinde denenerek zemin kaynak kullanım profili üzerindeki asgari etkisi hesaplanıp hafızaya alınmaktadır. Bu işlem yapılırken, programlanmamış aktivitelerden birinin hangi günde başlamasıyla asgari etkiyi verdiği durum, diğer bir programlanmamış aktivitenin asgari etki hesabına aktarılmamaktadır. Yani, programlanmamış her bir aktivitenin zemin kaynak kullanım profili üzerindeki etkisi birbirinden bağımsız hesaplanmaktadır. Bu durum, dar bir alt sınır değeri hesaplaması



yapmak için bir kayıp olduğu gibi hesaplanan değerin alt sınır olabilme koşulunu sağlaması amacıyla ve aynı zamanda alt sınır değeri hesabında işlem süresini kısa tutmak amacıyla uygulanmaktadır.

Asgari etki hesaplı alt sınır değeri, kaynakları tek tek dağıtarak hesaplanan alt sınır değerine göre çok farklı bir gelişim göstermektedir. Örneğin, asgari etki hesaplı alt sınır değeri algoritmanın başlarında kaynakları tek tek dağıtarak hesaplanan alt sınır değerine göre genelde çok daha küçük değerler bulmaktadır. Ancak, bu yöntemin artışı çok hızlı bir biçimde artış göstererek alt sınır değerleri aralığını çok çabuk bir şekilde daraltmasıdır. İki alt sınır değeri hesabı, ayrı ayrı kullanıldığı zaman algoritmanın problem seti çözümedeki performans dağılımı farklı olmaktadır. Örneğin, belli bir problemi çözerken birinci hesap iyi bir performans gösterirken ikinci hesap çok kötü bir performans sergileyebilmekte ya da tam tersi olmakta, birincisi kötü bir performans gösterirken ikincisi iyi bir performans sergilemektedir. Her iki alt sınır değeri hesabının farklı güçlerinin olması iki hesabı aynı anda kullanma fikrini doğurmuştur. İki hesap yöntemi birlikte kullanıldığında algoritmanın aynı problem setini çözümedeki performansı çok iyileşmiştir. Genel olarak, algoritmanın 10-100 kat daha hızlı çalıştığı gözlemlenmiştir.

Bu iki alt sınır değeri hesabı birlikte çalıştırıldığında alt sınır değeri hesabı için harcanan süre artmıştır. Bu süre alt sınır değeri hesabı yapılan budak sayısı ile çarpıldığında algoritmanın toplam süresinin %70-80 gibi bir payını harcadığı görülmüştür. Bunun için her iki hesap yöntemi için yazılan kodlarda değişikliğe gidilmiştir. Bunun için, anne budaktan çocuk budaklar oluşturulurken veraset yöntemiyle çocuk budaklardan birinde yapılan alt sınır değeri hesaplama verileri diğer çocuklara miras edilerek alt sınır değeri hesaplama fonksiyonlarının süresi oldukça kısaltılmıştır. Yani, bir anne budaktan 40 adet çocuk budak oluşturulduğu varsayıldığında alt sınır hesaplama fonksiyonu 40 budak için ayrı ayrı 40 defa çalıştırmak yerine, ilk çocuk budak için çalıştırıp hesaplama verilerini kardeş budaklara aktarmak, ve kardeş budaklarda sadece gerekli değişiklikleri yaparak, alt sınır değerini hesaplamada süre açısından daha verimli bir sistem olmuştur. Bu düzenleme de algoritmanın işlem süresi performansını önemli ölçüde arttırmıştır.

#### **4.4.7. Sabit ve Dinamik Budak Cinsiyeti Seçimi**

Anne budaktan çocuk budak oluştururken çocuk budağın cinsiyetinin yani bu budakta programlaması yapılacak aktivitenin seçiminin alt sınır değeri artış biçimini önemli ölçüde etkilemektedir. Bu doğrultuda budak cinsiyeti için biri sabit, diğeri dinamik olan iki fonksiyon



geliştirilmiştir. Sabit budak cinsiyeti seçici fonksiyonda, önceden belirlenmiş bir aktivite sıralama kriterine göre sırası gelen aktiviteden çocuk budaklar oluşturulmaktadır. Buna göre her kuşak anneden oluşacak çocukların cinsiyeti önceden belirlenmiştir. Örneğin, 5. kuşak anne budaktan oluşturulacak çocuk budaklarda önceden belirlenmiş kriterlere göre yapılan sıralamada 6. sıraya denk gelen aktivite programlanır (yani 6. kuşağın cinsiyeti sıralamada 6. sıraya denk gelen aktivite kimliğidir). Sabit budak cinsiyeti seçimi için denemelerde en iyi sonuç verdiği görülen aşağıdaki sıralama kriteri kullanılmıştır.

- En yüksek (aktivite kaynak ihtiyacı)
- En yüksek (aktivite kaynak ihtiyacı)\*(kaynak tahsisi yapılmamış gün sayısı) değeri
- En küçük (bolluk günleri sayısı)
- İşlem sırasında sonra gelen aktivite

Buna göre bir sıralama kriteri iki aktivite için eşitse bir sonraki kriterlere göre sıralama yapılmaktadır. Bu kriterin uygulanmasıyla, alt sınır değeri hesaplaması esnasında ölçütünü en çok etkileme kabiliyetine sahip aktivitelerin önce programlanması sağlanarak alt sınır değeri aralığının daha çabuk daralması amaçlanmıştır.

Dinamik budak cinsiyeti seçiminde, çocukları oluşturulacak her anne budak için henüz programlanması yapılmamış aktivitelerin ortalama etki değerleri hesaplanır ve bu değerler arasından en büyük değere sahip olan aktivite çocuk budakların cinsiyeti olarak seçilir. Dolayısıyla, aynı kuşaktan olan anne budakların çocuk budaklarının cinsiyeti birbirinden farklı olabilir. Burada, henüz programlanmamış aktiviteler için hesaplanan ortalama etki değerleri bu aktivitelerin çocuk budaklarda programlanması yapılması durumunda alt sınır değerinde yaratacakları ortalama artış miktarını belirtmektedir. Ortalama etki değeri, her aktivitenin bolluk günleri içerisinde programlanması durumunda zemin kaynak kullanım profili üzerinde yarattığı etkilerin toplanıp bolluk günleri sayısına bölünmesiyle elde edilmektedir. Bu hesap çoğu zaman asgari etki hesaplı alt sınır değerinde yapılan hesaplarla ortak işlemler içerdiği için bu işlemlerin bu fonksiyona miras edilmesiyle yapılmaktadır. Dinamik budak cinsiyeti seçimi, her anne budaktan oluşturulacak çocuk budaklarda alt sınır değerini en çok arttıracak aktivitenin seçilmesi duyarlılığını oldukça artırmıştır. Bu yöntem de alt sınır değerleri aralığının daha çabuk daralmasını sağlayarak kesin çözüme ulaşmak için gerekli süreyi kısaltmıştır.



#### 4.4.8. Budakların Birbirlerine Bağlanmasında Kullanılan Değere Göre Adresleme Yöntemi

Bir anne budaktan çocuk budakların oluşturulması sırasında yeni oluşturulan budakların aday anne budak zincirinde nasıl yerleştirileceği dal ve sınır algoritmasının önemli bir kalemını oluşturmaktadır. Bir önceki rapor döneminde de belirtildiği üzere bunun için tüm budaklar bir yerleşim kriterine göre sıraya dizilmekteydi. Bu yerleşim kriteri baştan sona doğru:

- En iyi (en küçük) alt sınır değerine sahip budak,
- En çok kritik olmayan programlanmış aktiviteye sahip budak,
- İşlem sırasında sonra gelen budak önceliklerini içermekteydi.

Yeni oluşturulan çocuk budağın zincire yerleştirilmesi, zincirin en başındaki budaktan başlayarak yukarıda belirtilen kriterlere göre sırayla her budakla karşılaştırılıp uygun yere yerleştirilmesi esasına dayanıyordu. Bu arada, bir yerleşim kriteri iki budak için eşitse bir sonraki kriterlere göre yerleşim yapılmakta bu şekilde budaklar birbirlerine bağlanarak sıraya dizilmekteydi. Yeni oluşturulan çocuk budakların zincire yerleştirilmesi için uygun yer bulununcaya kadar tek tek budak zincirindeki tüm budakların adreslerine gidip budakların alt sınır bilgisi okunmaktaydı. Bu okuma işleminin algoritma performansını büyük oranda düşürdüğü görülmüştür. Budak zincirinde kimi zaman yüz milyonlara varan budak sayısı olduğu düşünüldüğünde, yeni oluşturulan her bir budağın uygun yere yerleştirilebilmesi için yüz milyonlara varan adres tarama işlemi yapılması gerekmektedir.

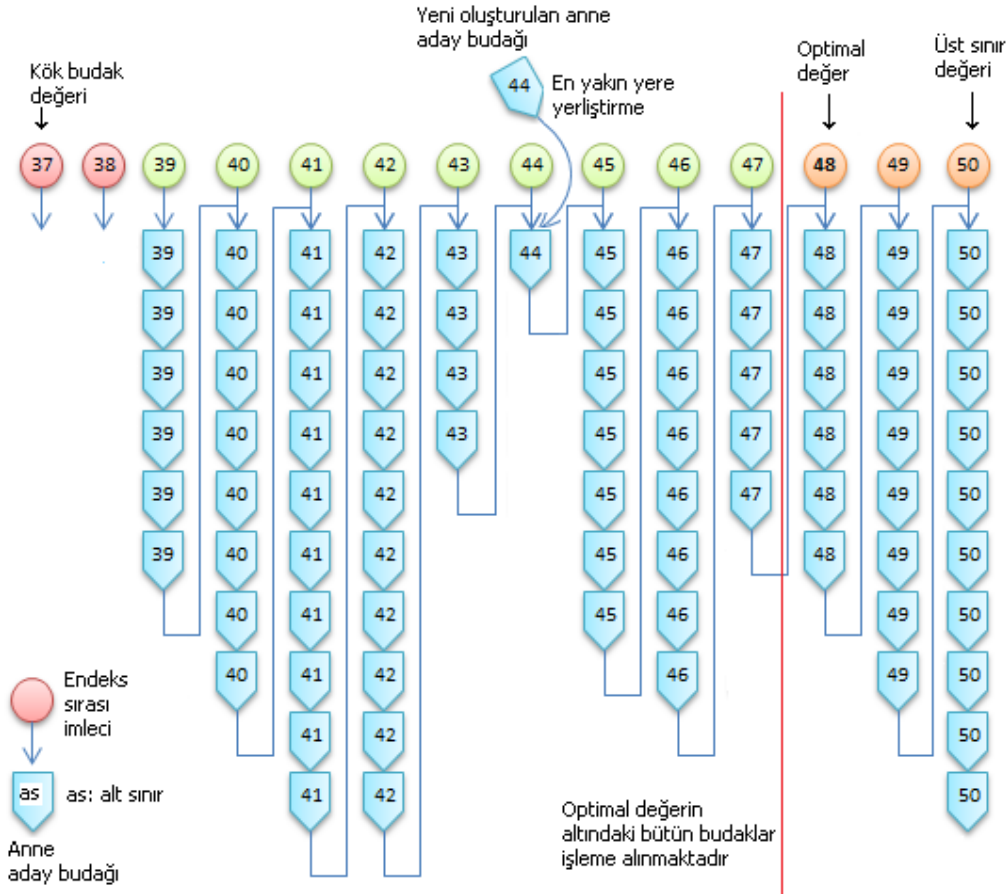
Budak zincirinde, her budak kendinden sonra gelen budağın adresini taşımakta ve işleme alınacak anne budak da budak zincirinin en başındaki budağın zincirden koparılıp işleme alınmasıyla seçilmektedir. Bu sistem aynı şekilde kullanılmaya devam edilmekle birlikte yeni oluşturulan budakların zincire yerleştirilmesi yönteminde değişikliğe gidilmiştir. Budakların alt sınır değerine göre adresleme yöntemi kullanılarak yeni budaklar zincire yerleştirilmeye başlanmıştır.

Bölüm 4.4.4'de, kesin çözümün bulunabilmesi için optimal değer altında alt sınır değerine sahip olan tüm budakların işleme alınmasının kaçınılmaz olduğunu belirtmiştik. Dolayısıyla yerleşim kriteri baştan sona doğru:

- En iyi (en küçük) alt sınır değerine sahip budak,
- İşlem sırasında sonra gelen budak

şekline indirgenerek kullanılmıştır. Oluşturulacak budak zincirinde, en küçük alt sınır değeri kök budakta bulunan değer iken en büyük alt sınır değeri üst sınır değeri olmaktadır (alt sınır

değeri üst sınır değerinden büyük veya eşit olan budaklar zincire eklenmeden hafızadan silinmektedir). Geriye kalan tüm budaklar bu iki değer arasında bir alt sınır değerine sahip olacaktır. Yeni yerleştirme yönteminde, boyutu, üst sınır değeri ile kök budak alt sınır değerinin farkı büyüklüğünde olan bir adres hafıza dizisi oluşturulur. Kullanılacak ölçüte bağlı olarak en küçük alt sınır değeri ve üst sınır değeri arasındaki değerler bir endeks dönüştürücü işlemle bu hafıza dizisinin endeksleri haline getirilir. Adres hafıza dizisindeki her eleman, endeksinin karşılık geldiği alt sınır değerine sahip ilk budağın bilgisayar hafızasındaki adresini tutar. Oluşturulan yeni çocuk budakların alt sınır değeri hesaplandıktan sonra bu değere karşılık gelen endeks bulunur. Adres hafıza dizisinde bu endeksin tuttuğu adrese gidilir, ve yeni oluşturulan budak bu adresteki budağın önüne yerleştirilerek gerekli bağlantılar yapılır. Böylece, yeni budak kendisinden daha küçük alt sınır değerine sahip budaklardan sonra, kendisiyle eşit veya kendisinden daha büyük alt sınır değerine sahip budaklardan önce gelen bir yere yerleştirilmiş olur. Değere göre adresleme yöntemi Şekil 18’de gösterilmiştir.



Şekil 18. Değere göre adresleme yöntemi



Bu yöntemle, zincirde yüz milyon budak olsa bile her hangi bir budak taraması yapılmadan yeni budak doğrudan alt sınır değerine karşılık gelen endeks kullanılarak budak zincirinde doğru yere yerleştirilmektedir. Değere göre adresleme yöntemiyle yeni budakların budak zincirinde doğru yere yerleştirilmesi işleminde algoritma performansı artırılmış, bu işlemin payı genel algoritmada çok küçük yüzdelere indirilmiştir.

## **5. DAL ve SINIR ALGORİTMASININ PERFORMANSININ BELİRLENMESİ İÇİN YAPILAN TESTLER**

Literatürde kaynak dengeleme probleminin kesin çözümü için problem setlerinin kullanıldığı üç adet çalışma vardır. Bu çalışmalardan ilki Neumann ve Zimmermann (2000)'in önerdiği başlangıç zamanı sayımlı (start time enumeration) dal ve sınır algoritmasıdır. Gather, Zimmermann ve Bartels 2010 yılında yine dal ve sınır algoritmalarının kullanıldığı köprü tabanlı yöntem (bridge-based approach) ve ağaç adaptasyonu (tree adaptation) yöntemlerini önermişlerdir (Gather vd. 2010). Rieck, Zimmermann ve Gather 2012 yılında çeşitli doğrusal tamsayılı modeller önererek kaynak dengeleme problemini iki farklı ölçüt için çözdürmüştür. Bu proje önerisinde Neumann ve Zimmermann (2000)'in ve Gather, Zimmermann ve Bartels 2010'in geliştirdiği dal ve sınır algoritmalarından daha iyi sonuç veren bir algoritma geliştirilmesi hedeflenmiştir. Bununla birlikte geliştirilen kesin algoritmalar hem Neumann ve Zimmermann (2000)'in ve Gather vd. (2010)'in geliştirdiği dal ve sınır algoritmaları ile kıyaslanmış, hem de proje önerisinden sonra yayımlanan Zimmermann ve Gather (2012)'in geliştirdiği doğrusal tamsayılı modeller ile kıyaslanmıştır. Bu çalışmalarda kullanılan problem setleri kritik yol yönteminin daha genel hali olan süre ilişkili programlama (Temporal Scheduling) yöntemine göre hazırlandıklarından testlerde öncelikle bu iş programlama yöntemi kullanılmıştır.

Gather vd. (2010)'de yapılan testler, Neumann ve Zimmermann (2000)'in geliştirdiği başlangıç zamanı sayımlı (start time enumeration) yöntemin güncel sonuçlarını da kapsamaktadır. Bu projede testler, 64 bit'lik bir platformda 3.33 GHz işlemci hızına sahip bilgisayarda yapılmıştır. Gather vd. (2010) ve Rieck Zimmermann ve Gather (2012)'daki testler 2.67 GHz işlemci hızına sahip bilgisayarda yaptığından bu test süreleri ile projede geliştirilen dal ve sınır algoritmasının test süreleri kıyaslanırken projedeki test süreleri 3.33/2.67 oranında artmıştır. Proje kapsamında geliştirilen dal ve sınır algoritması Visual Studio 2010 yazılımı ile C++ dili kullanılarak kodlanmıştır.

### 5.1. Testlerde Kullanılan Ölçütler

Neumann ve Zimmermann (2000) ve Gather vd. (2011)'deki problemler sadece karelerin toplamı ölçütüne göre çözdürülmüştür. Karelerin ağırlıklı toplamını en aza indirme ölçütü, proje süresi boyunca kullanılan her kaynağa ait günlük kaynak kullanım miktarlarının karelerinin toplamalarının ağırlıklı bileşimini en aza indirmeyi amaçlar. Rieck, Zimmermann ve Gather (2012)'deki problemler ise hem karelerin toplamı, hem de, ortalamayı aşan kaynakların ağırlıklı toplamını en aza indirme ölçütüne göre çözdürülmüştür. Ortalamayı aşan kaynakların ağırlıklı toplamını en aza indirme ölçütü, proje süresi boyunca kullanılan her kaynağa ait günlük kaynak kullanım miktarlarının her kaynak için hesaplanan ortalamaları aşmış kısımlarının toplamalarının ağırlıklı bileşimini en aza indirmeyi amaçlar. Bu ölçüt, ortalama kaynak kullanımından olan sapmaların mutlak değerlerinin toplamını en aza indirmeyi amaçlayan ölçütle aynı işi yapmakta olup sapmalardan sadece ortalama üstünde kalanları (yani pozitif sapmaları) dikkate almaktadır.

Proje kapsamında geliştirilen dal ve sınır algoritmasının performansının belirlenmesi için yapılan testlerde literatürdeki çalışmalarda kullanılan karelerin toplamı ve ortalamayı aşan kaynakların ağırlıklı toplamını en aza indirme ölçütleri kullanılmış, bunlara ilave olarak El-Rayes ve Jun (2009) inşaat yapım projeleri için geliştirdikleri ve 2.2. Bölümde değinilen Azami Kaynak Kullanımı ve Atıl Kaynak Günü (MRD+RID) ölçütü de kullanılmıştır. Bu ölçüt azami kaynak kullanımı ve atıl kaynak günü toplamını (MRD+RID) en aza indirmeyi hedeflemektedir. Testlerde kullanılan ölçütlerin ayrıntıları 3.1.4. bölümde verilmiştir.

### 5.2. Gather vd. (2010)'daki Problemlerin Test Sonuçları

İlk olarak geliştirilen dal ve sınır algoritması ile Gather vd. (2010)'deki 10 aktivitelik 270 problemde oluşan bir set (*rlpj\_10*), 10 aktivitelik 90 problemde oluşan bir set (*ubo\_10*), ve 20 aktivitelik 90 problemde oluşan iki set (*rlpj\_20* ve *ubo\_20*) olmak üzere toplam 540 problem karelerin toplamı ölçütüne göre çözdürülmüş ve sonuçlar Tablo 3'de verilmiştir. Tablo 1'de verilen Start Time Enu. yöntemi Neumann ve Zimmermann (2000)'in geliştirdiği başlangıç zamanı sayımlı (start time enumeration) dal ve sınır algoritmasıdır. Tablo 3'de kıyaslamalarda kullanılan bridge-based ve tree adaptation yöntemleri ise Gather vd. (2011)'de bahsi geçen dal ve sınır algoritması tabanlı yöntemlerdir. Bu projede geliştirilen dinamik budak cinsiyeti dal ve sınır algoritması Prj 111M140 ismiyle Tablo 3' de yer almaktadır.

Tablo 3. Gather vd. (2010) problem setleriyle yapılan karelerin toplamı ölçütüne göre test sonuçları

<i>rlpj_10</i>		Kullanılan Yöntem	Ortalama Süre	< 1 s	< 10 s
270 problem	Prj 111M140	Dinamik Budak Cinsiyeti	0.017 s	270	270
	Gather vd.(2011)	Bridge based	0.02 s	270	270
		Tree adaption	0.44 s	215	268
	Neumann ve Zimmermann (2000)	Start Time Enu.	0.66 s	241	263

<i>ubo_10</i>		Kullanılan Yöntem	Ortalama Süre	< 1 s	< 10 s
90 problem	Prj 111M140	Dinamik Budak Cinsiyeti	0.034 s	90	90
	Gather vd.(2011)	Bridge based	0.19 s	86	90
		Tree adaption	5.29 s	43	80
	Neumann ve Zimmermann (2000)	Start Time Enu.	0.62 s	78	89

<i>rlpj_20</i>		Kullanılan Yöntem	Ortalama Süre	< 10 s	< 1000 s	< 36000 s
90 problem	Prj 111M140	Dinamik Budak Cinsiyeti	22.87 s	86	89	90
	Gather vd.(2011)	Bridge based	373.38 s	35	85	90
		Tree adaption	6069.97 s	15	41	90

<i>ubo_20</i>		Kullanılan Yöntem	Ortalama Süre	< 10 s	< 1000 s	< 36000 s
90 problem	Prj 111M140	Dinamik Budak Cinsiyeti	28,83 s	76	90	90
	Gather vd..(2011)	Bridge based	22.28 s	27	53	74
		Tree adaption	5549.42 s	6	16	41

Tablo 3'de bahsi geçen dal ve sınır algoritmalarının 10, 1000 ve 36000 saniyelik süreler içerisinde çözdüğü problem sayısı verilmiştir. Tablo 3 incelendiğinde bu projede geliştirilen dal ve sınır algoritmasının (Prj 111M140), gerek Neumann ve Zimmermann (2000)'ın geliştirdiği start time enumeration yönteminden gerekse, Gather vd. (2011)'in geliştirdiği bridge based ve tree adaption yöntemlerinden çok daha yüksek performansla çalıştığı görülmektedir. Proje kapsamında geliştirilen dal ve sınır algoritması 10 aktivitelik setlerdeki tüm problemleri 1 saniye içerisinde, 20 aktivitelik setlerdeki problemleri ise birisi dışında 1000 saniyelik süre içerisinde çözebilmiştir. Literatürdeki diğer üç dal ve sınır yöntemi bu performansın çok gerisinde kalmıştır. Yine tüm setlerdeki ortalama algoritma süresinde



mevcut yöntemlerin ortalamasından çok daha iyi sonuçlar elde edilmiştir. Prj 111M140 algoritması için ubo\_20 problem seti için hesaplanan ortalama süreye bütün 90 problemin süreleri dahil edilmiştir. Ancak bridge based algoritmasında 36000 saniyede çözilemeyen problemlerin süresi ortalama hesabına katılmadığı için bu algoritmanın ortalama süresi Prj 111M140'a göre biraz daha küçüktür. Ancak ubo\_20 problem seti için 1000 saniye içerisinde çözülen problem sayısı incelendiğinde Prj 111M140 algoritmasının bridge based algoritmasına göre çok daha hızlı olduğu görülmektedir. Prj 111M140 algoritması 1000 saniye içerisinde 90 problemin hepsini çözmüş, bridge based algoritması için sadece 53 problemi çözebilmiştir. Sonuç itibari ile proje kapsamında geliştirilen dal ve sınır algoritması ile literatürdeki mevcut dal ve sınır algoritmalarının performansının önemli ölçüde önüne geçilmiş ve proje hedeflerinden bir tanesine ulaşılmıştır.

### **5.3. Reick vd. (2012) Problemleri ve Karelerin Toplamı ve Ortalamayı Aşan Kaynaklar Toplamı Ölçütleri için Yapılan Testler**

Rieck, Zimmermann ve Gather 2012'de bu projenin önerisinden sonra yayınlanan makalelerinde, KDP'nin kesin çözümü için doğrusal tamsayılı programlama yöntemiyle çeşitli modeller sunmuş ve bu modelleri geniş bir problem seti kullanarak test etmiştir. 10, 15, 20, 30 ve 50 aktivite tekli, üçlü ve beşli kaynaklı her biri 40 problemden oluşan setler olmak üzere toplam 600 problemi karelerin toplamı ve ortalamayı aşan kaynaklar toplamı ölçütüne göre çözdürülmüştür. Bu proje kapsamında geliştirilen dal ve sınır tabanlı kesin çözüm algoritması aynı seti kullanılarak 30 aktivite problem kadar çözdürülmüştür. 10, 15 ve 20 aktivite tüm problem setlerinde proje kapsamında geliştirilen dal ve sınır tabanlı kesin çözüm algoritmasının performansı Rieck, Zimmermann ve Gather'in modellerine göre daha iyi olmuştur. Ancak, 30 aktivite problem setlerinde karelerin toplamı ve ortalamayı aşan kaynaklar toplamı ölçütleri için doğrusal tamsayılı yönteminden daha yüksek bir performans elde edilememiştir. Bu sebeple 50 aktivite problem setleri denenememiştir. Sonuçlar, Tablo 4' ve Tablo 5'de verilmiştir.

Tablo 4 ve Tablo 5 'de proje kapsamında geliştirilen dal ve sınır algoritmasına ait sonuçlar sabit budak cinsiyeti (SBC) ve dinamik budak cinsiyeti (DBC) yöntemleri için verilmiştir. Rieck vd.(2012)'e ait sonuçlar ise iki ayrı doğrusal tamsayılı yöntemi (karelerin toplamı ölçütü için M1 ve M2, ortalamayı aşan kaynaklar toplamı ölçütü için M3 ve M4) için verilmiştir. Tablo 4 ve Tablo 5'de aynı zamanda Gather vd. (2010)'in geliştirdiği tree adaption dal ve sınır tabanlı algoritmasının (Tree-based B&B) da sonuçları dahil edilmiştir. Testlerde, M1, M2, M3, ve M4

Tablo 4. Rieck vd. (2012) problem setleriyle ve karelerin toplamı ölçütü ile yapılan testlerin sonuçları

	Prj 111M140				Rieck vd. (2012)					
	SBC		DBC		M1		M2		Tree adaption B&B	
	Süre (s)	✓	Süre (s)	✓	Süre (s)	✓	Süre (s)	✓	Süre (s)	✓
<i>rlp_10_1</i>	0.012	40	0.011	40	0.029	40	0.029	40	0.028	40
<i>rlp_10_3</i>	0.013	40	0.014	40	0.153	40	0.108	40	0.033	40
<i>rlp_10_5</i>	0.021	40	0.024	40	0.199	40	0.206	40	0.049	40
<i>rlp_15_1</i>	0.032	40	0.035	40	0.154	40	0.154	40	0.146	40
<i>rlp_15_3</i>	0.048	40	0.043	40	0.291	40	0.356	40	25.722	40
<i>rlp_15_5</i>	0.072	40	0.061	40	0.513	40	1.194	40	29.397	40
<i>rlp_20_1</i>	0.190	40	0.282	40	0.336	40	0.691	40	1792.086	38
<i>rlp_20_3</i>	1.259	40	0.468	40	2.410	40	10.994	40	1540.256	35
<i>rlp_20_5</i>	1.445	40	0.968	40	2.410	40	41.639	40	3392.972	31
<i>rlp_30_1</i>	292,833	37	308,919	39	9.983	40	-	-	-	-
<i>rlp_30_3</i>	146,638	39	54,427	40	18.389	40	-	-	-	-
<i>rlp_30_5</i>	338,914	33	401,748	36	61.198	40	-	-	-	-

Tablo 5. Rieck vd. (2012) problem setleriyle ve ortalamayı aşan kaynaklar toplamı ölçütü ile yapılan testlerin sonuçları

	Prj 111M140				Rieck vd. (2012)					
	SBC		DBC		M3		M4		Tree adaption B&B	
	Süre (s)	✓	Süre (s)	✓	Süre (s)	✓	Süre (s)	✓	Süre (s)	✓
<i>rlp_10_1</i>	0.010	40	0.014	40	0.031	40	0.014	40	0.022	40
<i>rlp_10_3</i>	0.018	40	0.019	40	0.140	40	0.027	40	0.023	40
<i>rlp_10_5</i>	0.027	40	0.027	40	0.298	40	0.051	40	0.038	40
<i>rlp_15_1</i>	0.027	40	0.030	40	0.309	40	0.096	40	9.733	40
<i>rlp_15_3</i>	0.048	40	0.052	40	0.975	40	0.071	40	20.043	40
<i>rlp_15_5</i>	0.077	40	0.070	40	14.390	40	0.112	40	23.640	40
<i>rlp_20_1</i>	0.102	40	0.166	40	1.236	40	0.191	40	1566.697	38
<i>rlp_20_3</i>	0.826	40	0.568	40	16.086	39	1.172	40	2283.954	37
<i>rlp_20_5</i>	0.847	40	1.506	40	48.214	40	1.523	40	2751.409	31
<i>rlp_30_1</i>	58.128	40	182.689	40	-	40	12.529	-	-	-
<i>rlp_30_3</i>	149.008	39	144.958	40	-	40	14.155	-	-	-
<i>rlp_30_5</i>	308,185	36	415.695	36	-	40	101.538	-	-	-

yöntemlerinin kullandığı modellerin çözümü için problem başına en fazla 3 saat, Tree adaption B&B algoritması içinse problem başına en fazla 6 saat işlem süresi sınırı

konulmuştur. Bu proje kapsamında geliştirilen algoritmalar içinse problem başına en fazla 9000 saniye işlem süresi sınırı konulmuştur.

Tablo 4 ve Tablo 5'teki sonuçlar Gather vd. (2010) dal ve sınır tabanlı (Tree adaption B&B) algoritmasına göre bu projede geliştirilen dal ve sınır algoritmasının performansının çok daha iyi olduğu bir kez daha teyid edilmiştir. Aynı zamanda proje kapsamında geliştirilen dal ve sınır algoritmalarını 20 aktiviteye kadar olan tüm problem setlerinde Rieck vd. (2012)'in doğrusal tamsayı yöntemlerinden daha iyi sonuç vermiştir. Bununla birlikte 30 aktiviteli problemlerde Zimmermann ve Gather'in modelleri daha iyi sonuç vermiştir.

#### **5.4. Mutlu (2010) Problem Setleriyle ve Azami Kaynak Kullanımı ve Atıl Kaynak Günü (MRD+RID) Ölçütü ile Yapılan Testler**

Kaynak dengeleme problemi için önerilen fakat henüz kullanımı yeni olan azami kaynak kullanımı ve atıl kaynak günü toplamını en aza indiren ölçüte göre literatürde yapılmış kapsamlı bir problem seti ve test sonuçları bulunmamaktadır. Bunun için, ilk olarak Çağdaş Mutlu (2010)'nun Yüksek Lisans tezi için geliştirdiği problem setleri kullanılmıştır. Testlerde projede geliştirilen dinamik budak cinsiyeti dal ve sınır algoritması ile en büyüğü 20 aktivitelik olan tekli ve çoklu kaynaklı 21 problemden oluşan set çözdürülmüştür. Aynı problemler Çağdaş Mutlu (2010)'nun azami kaynak kullanımı ve atıl kaynak günü toplamını en aza indiren ölçüt için uyarladığı başlangıç zamanı sayımlı dal ve sınır yöntemi-Start Time Enum. (Neumann ve Zimmermann, 2000) algoritması ile çözülmüştür. Sonuçlar Tablo 6'da verilmiştir. Sonuçlar karşılaştırıldığında, proje kapsamında geliştirilen dal ve sınır algoritması hem tekli hem çoklu kaynaklı durum için problem setindeki bütün problemleri 1 saniyenin altında bir sürede çözmüştür. Diğer bir tarafta, başlangıç zamanı sayımlı dal ve sınır yöntemi algoritması (Neumann ve Zimmermann, 2000) aynı seti tekli kaynaklar için ortalama 56 saniyelik bir sürede, çoklu kaynaklar içinse ortalama 471 saniyelik bir sürede çözebilmiştir. Tablo 6 incelendiğinde bu projede geliştirilen dal ve sınır algoritmasının (Prj 111M140), Neumann ve Zimmermann (2000)'in geliştirdiği start time enumeration yönteminden karelerin toplamı ölçütü için olduğu gibi azami kaynak Kullanımı ve atıl kaynak günü (MRD+RID) ölçütü için de çok daha kısa sürede çözüm elde ettiği görülmektedir.

Tablo 6. Mutlu (2010) Problem Setleriyle ve Azami Kaynak Kullanımı ve Atıl Kaynak Günü (MRD+RID) Ölçütü ile Yapılan Testlerin Sonuçları

Problem	Akt.	İşlem Süreleri				
		Prj 111M140		Start Time Enu.		
		TEKLİ KAYNAK	ÇOKLU KAYNAK	TEKLİ KAYNAK	ÇOKLU KAYNAK	
1	El Rayes ve Jun (2009)	22	0,467	0,546	39	3414
2	Stevens (1990 Sayfa 97)	17	0,430	4,719	493	4009
3	Stevens (1990 Sayfa 172)	21	0,406	1,238	257	854
4	Harris (1990)	13	0,234	0,266	0	1
5	Leu (2000)	15	0,336	2,186	216	261
6	Son and Skibniewski (1999)	15	0,298	0,338	28	144
7	Son and Skibniewski (1999)	13	0,214	0,240	0	1
8	Generated II (Mutlu 2010)	8	0,182	0,185	0	0
9	Generated III (Mutlu 2010)	16	0,478	0,423	0	0
10	Generated IV (Mutlu 2010)	20	0,416	0,436	0	0
11	Generated V (Mutlu 2010)	15	0,321	0,354	1	2
12	Generated VI (Mutlu 2010)	16	0,373	0,452	3	86
13	Generated VII (Mutlu 2010)	18	0,398	0,547	49	228
14	Mubarak (2004 Sayfa 61)	16	0,358	0,536	72	835
15	Mubarak (2004 Sayfa 67)	13	0,213	0,235	0	0
16	Mubarak (2004 Sayfa 217)	10	0,152	0,157	0	0
17	Newitt (2005 Sayfa 82)	18	0,237	0,545	5	17
18	Newitt (2005 Sayfa 121)	14	0,119	0,128	0	0
19	Hinze (2004 Sayfa 152)	17	0,212	0,218	17	45
20	Demeulemeester (2002)	12	0,250	0,264	0	0
21	Easa (1989)	7	0,094	0,126	0	0
Ortalama Süre			0,295	0,674	56,19	471,29

### 5.5. Rieck vd. (2012) Problemleri ve Azami Kaynak Kullanımı ve Atıl Kaynak Günü (MRD+RID) Ölçütü ile Yapılan Testler

Proje kapsamında geliştirilen dal ve sınır algoritmasının azami kaynak kullanımı ve atıl kaynak günü toplamını en aza indiren ölçüt için, doğrusal tamsayı yöntemine göre performansının belirlenebilmesi için, proje kapsamında geliştirilen dal ve sınır algoritmasının, yine proje kapsamında geliştirilen doğrusal tamsayılı eniyileme modeli ile karşılaştırılması Rieck vd. (2012)'nin test setleri kullanılarak yapılmıştır. Testler, 64 bit'lik bir platformda 3.33 GHz işlemci hızına sahip bilgisayarda yapılmış, ve testler 9,000 saniyede sonlandırılmıştır.

Proje kapsamında geliştirilen dal ve sınır algoritması (B&B-NGD3) ve yine proje kapsamında geliştirilen karma-tamsayılı doğrusal modeli (KTDM) yöntemlerinin 9,000 saniye içerisinde çözdüğü problem sayısı ve çözülen problemlerin ortalama çözüm süreleri Tablo 1' de verilmiştir. Bütün problem setlerinde toplam 40 adet problem vardır. KTDM yöntemi rlp\_10\_1, rlp\_10\_3, rlp\_10\_5, ve rlp\_15\_1 test setleri için B&B-NGD3 yöntemine göre çok daha kötü bir performans sergilediğinden, bu yöntem diğer test setleri için kullanılmamıştır. Tablo 7'de verilen sonuçlardan proje kapsamında geliştirilen dal ve sınır algoritmasının 15 faaliyet ve beş kaynağa kadar olan problemleri ortalama 1.5 saniyeden daha az bir süre içerisinde çözdüğü görülmektedir. Bununla birlikte geliştirilen dal ve sınır algoritması 30 faaliyet ve beş kaynağa kadar olan problemleri MRD+RID ölçütü için çözmüştür. Buna karşın karma-tamsayılı doğrusal modeli, MRD+RID ölçütü için dal ve sınır algoritmasına göre kötü bir performans sergilemiştir. KTDM yöntemi 10 faaliyet ve üç kaynaklı dört problemi 9,000 saniyelik bir işlem süresinde çözememiş, çözebildiği problemleri için ise dal ve sınır algoritmasına göre çok daha fazla bir ortalama işlem süresi elde etmiştir. Proje kapsamında geliştirilen dal ve sınır algoritmasının literatürde ilk kez çözdüğü 30 faaliyet ve beş kaynağa kadar olan problemlerin MRD+RID ölçütü için çözümleri proje sonuçlarının yayınlanması için geliştirilen <http://www2.ce.metu.edu.tr/~sonmez/research/111M140> adresli web sitesinde araştırmacıların erişimine sunulmuştur.

Tablo 7. MRD+RID ölçütü için B&B ve DT yöntemlerinin sonuçları

	B&B-NG3		KTDM	
	Süre (s)	✓	Süre (s)	✓
<i>rlp_10_1</i>	0.011	40	42.809	40
<i>rlp_10_3</i>	0.137	40	78.778	36
<i>rlp_10_5</i>	0.051	40	405.6	31
<i>rlp_15_1</i>	0.156	40	324.492	32
<i>rlp_15_3</i>	0.228	40		
<i>rlp_15_5</i>	1.461	40		
<i>rlp_20_1</i>	34.839	40		
<i>rlp_20_3</i>	21.919	40		
<i>rlp_20_5</i>	234.198	40		
<i>rlp_30_1</i>	146.151	32		
<i>rlp_30_3</i>	457.275	35		
<i>rlp_30_5</i>	391.001	25		

## 5.6. Dal ve Sınır Algoritması ve Karma-Tamsayılı Modelleme Yöntemi ile Yapılan Testlerin Sonuçlarının Özeti

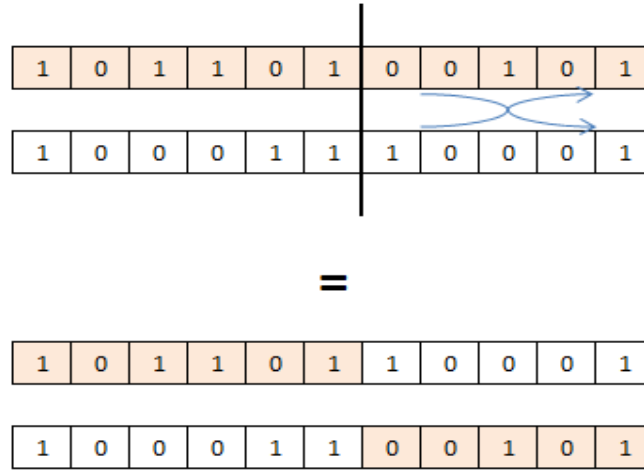
Kesin çözüm yöntemleriyle yapılan test sonuçları, geliştirilen dal ve sınır algoritmasının 20 faaliyete kadar olan küçük ölçekli problemlerin karaların toplamı ve ortalamayı aşan kaynaklar toplamı gibi geleneksel ölçütler için KDP'nin çözümünde performansının mevcut bütün yöntemlere göre daha iyi olduğunu göstermiştir. Ancak, 20 faaliyetten büyük problemlerin aynı ölçütler için çözümünde karma-tamsayılı doğrusal modelleme yönteminin (DTDM) dal ve sınır yöntemine göre daha başarılı olmuştur. Bununla birlikte proje kapsamında geliştirilen dal ve sınır algoritması mevcut dal ve sınır algoritmalarından gerek Neumann ve Zimmermann (2000)'in geliştirdiği start time enumeration yönteminden gerekse, Gather vd. (2011)'in geliştirdiği bridge based ve tree adaption yöntemlerinden bütün problemler için çok daha yüksek performansla optimal sonucu elde edebilmektedir. Son yıllarda El-Rayes ve Jun (2009) tarafından önerilen, atıl kaynak günü ve bununla birlikte azami kaynak kullanım miktarının (MRD+RID) eniyileştirilmesi, inşaat projeleri için pratik açıdan çok önemli bir ölçüt olarak ortaya çıkmıştır. Yapılan testlerde proje kapsamında geliştirilen dal ve sınır algoritması, karma-tamsayılı modelleme yöntemine göre MRD+RID için çok daha iyi bir performans sergilemiştir. Bu ölçüt için geliştirilen dal ve sınır algoritması 30 aktiviteye kadar olan problemleri çözebilmiştir. Sonuçlar proje kapsamında geliştirilen dal ve sınır algoritması yönteminin önemini irdeler niteliktedir.

## 6. ÜST SEZGİSEL KAYNAK DENGELEME ALGORİTMALARI

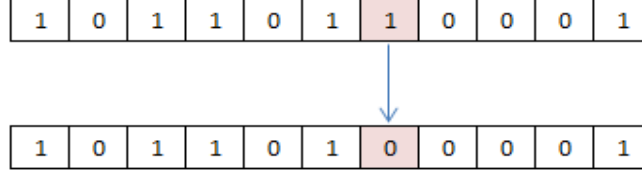
Gen algoritmaları (GA) kaynak dengeleme yönteminin çözümünde en yaygın olarak kullanılan üst sezgisel yöntemlerdir. Proje kapsamında melez üst sezgisel algoritmanın performansının test edilmesi amacıyla ilk olarak yalın bir GA geliştirilmiştir. Yalın gen algoritmasının kaynak dengeleme problemi için performansının artırılması için iki farklı strateji geliştirilmiştir. Bu stratejilerden ilki gen algoritmalarının tavlama benzetimi algoritması ile melezlemesidir. İkinci strateji ise melez algoritmadan elde edilen popülasyonun kaynak kullanımının bir sezgisel kullanılarak iyileştirilmesi yoluyla bir memetik algoritma elde edilmesidir. Proje kapsamında kaynak dengeleme problemi için geliştirilen yalın genetik algoritma (GA) ve tavlama benzetimli memetik algoritmanın (TBMA) ayrıntılarına ve performans testlerine takip eden bölümlerde değinilmiştir.

## 6.1 Genetik Algoritmanın Yapısı

Genetik algoritmalar, doğal seçim kanunlarından esinlenerek oluşturulmuş popülasyon tabanlı tarama stratejileridir. Bir GA'da, öncelikle doğal seçilime uğrayacak kromozomlar topluluğu meydana getirilir. Her kromozom, optimize edilmeye çalışılan problem için muhtemel bir çözüm olabilecek bir dizi sayısal veri içerir. Kromozomları oluşturan daha küçük veri birimlerine gen denir. Kromozomları oluşturan genler üzerinde değişiklikler yapılarak iki farklı biçimde yeni kromozomlar elde edilir. Şekil 19'da görüldüğü gibi iki ebeveyn kromozomun çaprazlaması ile genetik değişim sonucu veya Şekil 20'de görüldüğü gibi kromozomu oluşturan genlerden birini mutasyona uğratarak, yeni bir kromozom elde edilir. Her bir yeni kromozom üretme işleminden sonra üretilen kromozomun performansını belirlemek için bir uygunluk fonksiyonu kullanılır. Uygunluk fonksiyonu kullanılarak yeni kromozomlar üretmek üzere topluluktan kromozom seçim işlemi, GA'yı ön plana çıkaran özelliğdir (Blum ve Roli, 2008). Bu şekilde en uygun kromozoma en yüksek hayatta kalma şansı verilir. Bir çok yeni nesil üretme işleminden sonra kromozom topluluğunun optimal çözüme yakınsaması beklenir. Proje kapsamında melez üst sezgisel algoritmanın performansının test edilmesi amacıyla geliştirilen yalın GA 3 temel aşamadan oluşmaktadır. Bu aşamalar: 1) KYY'ye göre iş programı hazırlaması, 2) Başlangıç popülasyonunun oluşturulması, 3) Genetik işlemlerin uygulanması aşamalarıdır. Bu aşamaların ayrıntılarına bir sonraki bölümde değinilmiştir.



Şekil 19. İki kromozomun çaprazlamasıyla yeni kromozomlar elde edilmesi



Şekil 20. Kromozomu oluşturan genlerden birinin mutasyona uğratılarak yeni bir kromozom elde edilmesi

### 6.1.1. KYY'ye Göre İş Programı Hazırlaması

GA'ya başlamadan önce KYY'ye göre iş programı hazırlanarak, kritik yolda olan aktiviteler, kritik olmayan aktiviteler, kritik olmayan aktivitelere ait erken başlama (EB), erken tamamlama (ET), geç başlama (GB), geç tamamlama (GT) ve toplam bolluk (TB) süreleri bulunur. Bunlar GA'da kullanılacak kromozom topluluklarının oluşturulmasında girdi bilgileri olarak kullanılacaktır.

### 6.1.2 Başlangıç Popülasyonunun Oluşturulması

Bu aşamada önceden belirlenmiş bir sayı ( $N_k$ ) kadar kromozom başlangıç çözüm popülasyonu olarak oluşturulur. Bunun için:

- $N_k$  kadar kromozom gelişigüzel üretilir. Her kromozom, kritik olmayan aktiviteler için rastgele belirlenmiş başlama sürelerini tutan genlerden oluşur.
- Başlangıç popülasyonunu meydana getiren her bir kromozomun amaç fonksiyonuna göre değeri hesaplanır ve daha sonrada uygunluk değerleri hesaplanır.

### 6.1.3 Genetik İşlemlerin Uygulanması

Bir sonraki neslin topluluğunu oluşturmak için çaprazlama ve mutasyon gibi genetik işlemler uygulanır. Bu işlemler sırasıyla:

- Önceden belirlenmiş bir genetik çaprazlama oranıyla ( $P_c$ ) bir önceki nesilden bazı kromozomlar gelişigüzel seçilir. Daha sonra bir sonraki neslin kromozomlarını oluşturmak üzere tek-nokta genetik çaprazlama işlemi uygulanır.
- Yeni nesli oluşturmak için seçilecek kromozomlar seçkinci rulet-tekerleği yöntemine göre belirlenir. Bu yöntemde en iyi amaç fonksiyonu değerine sahip kromozom bir sonraki nesil için seçilir. Diğer kromozomlar geriye kalan kromozomlar arasından uygunluk değerlerine göre gelişigüzel seçilir. Bu yöntemle, daha iyi uygunluk





değerine sahip kromozomların daha yüksek hayatta kalma şansı bulunmaktadır. Seçilemeyen kromozomlara bir sonraki nesilde yer verilmez.

- Sonraki aşamada mutasyon işlemi, önceden belirlenmiş bir mutasyon olasılığına ( $P_m$ ) göre uygulanır.  $P_m$  değerine göre topluluk arasından belirli bir sayıda kromozom seçilir ve seçilen kromozomlara mutasyon işlemi uygulanır.

İkinci ve üçüncü işlemler önceden belirlenmiş bir çözüm kromozomları sayısına ( $N_s$ ) ulaşılan kadar (veya belli bir işlem süresine ulaşılan kadar) tekrar edilir. Son aşamada en son üretilen neslin en iyi amaç fonksiyonuna sahip olan kromozomu elde edilen en iyi sonuç olarak belirtilir.

GA'lar önemli bir üst-sezgisel taratma algoritmaları olmalarına rağmen tepe tırmanma ve ince ayar konusunda genellikle çok başarılı değillerdir. Bu konulardaki yeteneklerinin sınırlı olması GA'ların optimal çözüm veya optimal çözüme yakın değerlere yakınsamasını bazı durumlarda zorlaştırmaktadır. Bu sebeple proje kapsamında kaynak dengelemesi için ilk olarak oluşturulan genetik algoritmanın geliştirilmesi için tavlama benzetimi ve memetik algoritma integrasyon stratejileri kullanılmıştır. Bu doğrultuda geliştirilen melez üst sezgisel algoritma, tavlama benzetimli memetik algoritma (TBMA) olarak adlandırılmıştır.

## 6.2. Memetik Algoritma

Literatürde yer alan son yıllarda yapılan çalışmalarda, büyük ölçekli optimizasyon problemleri için üst-sezgisel algoritmalarla, sezgisel algoritmaların birleşiminden oluşan ve memetik algoritma olarak adlandırılan optimizasyon yöntemi önerilmiştir. Memetik algoritmaların, aralarında mühendislik, ekonomi, bilişim gibi bilim dallarının da olduğu çok farklı konularda başarılı uygulamaları vardır (Nguyen v.d. 2009, Pishvae v.d. 2010, Chica v.d. 2012, Neri v.d. 2012). Bu çalışmalarda memetik algoritmaların yerel taramalar için bireysel öğrenme yöntemini kullandıklarından, birçok farklı optimizasyon problemi için yalın sezgisel veya üst-sezgisel yöntemlere göre yüksek kaliteli çözümlere çok daha etkin bir şekilde ulaştıkları belirtilmiştir. Bu çalışmalar, memetik algoritmaların KDP'nin çözümünde önemli bir katkısı olabileceğini işaret etmektedir. Literatürdeki son çalışmaların ışığında, gen algoritması ile elde edilen kromozomlardaki kaynak kullanımının verimsiz olduğu kısımların azaltılması için proje önerisinde belirtilen Debels ve Vanhoucke'nin (2007) kısıtlı kaynaklı iş planlaması probleminin optimizasyonu için kullandığı çaprazlama operatörü yerine, memetik algoritmaların yerel taramalar için bireysel öğrenme özellikleri kullanılmıştır. Bu yöntem ilk



kromozom popülasyonunun ve çaprazlama ve mutasyon yolu ile elde edilen bütün çocuk kromozomların iyileştirilmesinde öncelik-temelli bir sezgiselin kullanılmasını içermektedir.

### 6.2.1 TBMA'da Kromozomların Yapısı

TBMA'da optimizasyon probleminin her bir sonuç adayı GA'da olduğu gibi kromozomlarla temsil edilmektedir. Bu şekilde, her bir sonuç onu temsil eden kromozomun içindeki gen adlı parametrelerin deşifre edilmesiyle bulunur. TBMA'da genler kritik olmayan faaliyetlerin başlangıç zamanları temsil etmektedir. Bu genler, 0 ve 1 arasında değer alan gerçek sayılardan oluşmaktadır. Genlerin aldıkları bu değerlerin deşifre edilmesiyle faaliyetlerin erken başlama ve geç başlama zamanları arasındaki başlama zamanı alternatiflerinin birini belirlenmektedir. Örneğin, bir genin değeri 0'a ne kadar yakınsa, o genle ilgili faaliyetin başlama zamanı o faaliyetin erken başlama zamanına o kadar yakındır. Benzer şekilde, bu değer 1'e ne kadar yakınsa faaliyet başlangıç zamanının onun geç başlama zamanına o kadar yakındır. Son ve Skibniewski (1999)'nin oluşturduğu örnek, kromozomların yapısını, genlerin kodlama ve deşifre işlemlerinin anlatılması için Şekil 21'de verilmiştir. Şekil 21'deki kısaltmaların açıklaması aşağıdaki gibidir:

*ID* : faaliyet numarası

*RR*: kaynak kullanım miktarı

*S* : faaliyet süresi

*EB*: erken başlama zamanı

*ET*: erken tamamlama zamanı

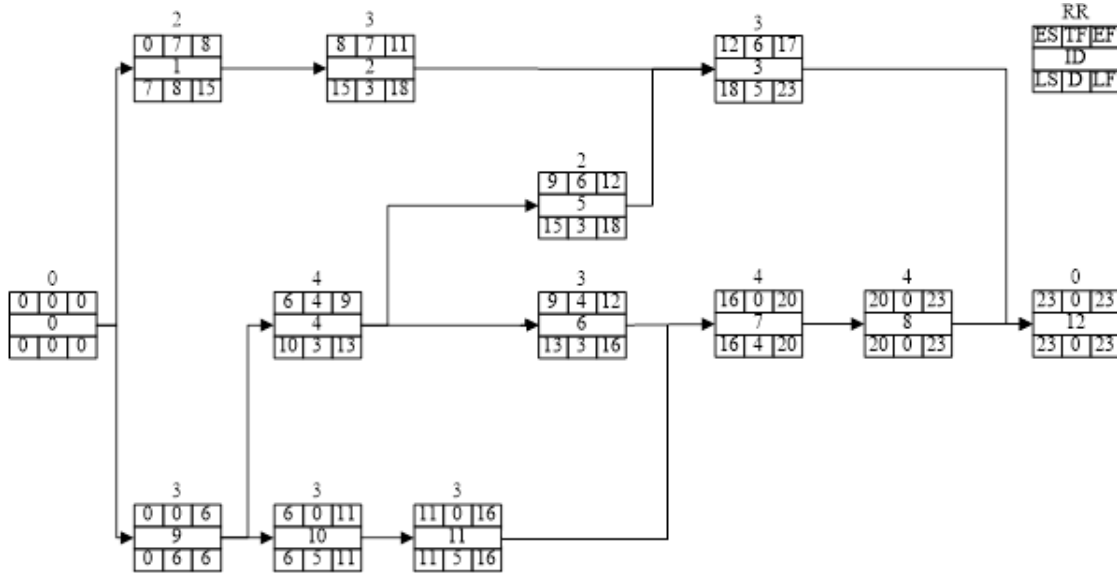
*GB*: geç başlama zamanı

*GT*: geç tamamlama zamanı

*TB*: toplam bolluk

Bu 13 faaliyetlik problemin, kritik olmayan 6 faaliyeti olduğundan, problem için 6 genlik bir kromozom oluşturulmuştur. Şekil 22'de rastgele seçilmiş genlerden oluşan bir kromozom yapısı gösterilmiştir. TBMA programlaması mümkün olan (öncelleri tamamlanmış) ve kritik olmayan faaliyetler listesindeki faaliyetleri, artan faaliyet numarası sırasına göre birer birer programlar. Örneğin, programlaması mümkün olan ve kritik olmayan aktiviteler listesinde ilk olarak 1. ve 4. aktiviter yer almaktadır. Bu faaliyetlerden 1. faaliyetin faaliyet numarası daha küçük olduğu için ilk olarak programlanır. Faaliyet 1'in toplam bolluğu 7 gün olduğundan, toplam 8 adet başlama günü alternatifi vardır. Bu yüzden 0'la 1 arasında, 8 aralık oluşturulur, ve her aralığın boyu  $0,125$  ( $1/8$ ) olur. Birinci faaliyetin başlama zamanını belirleyen genin

değeri 0,240 olduğundan, ve bu değer ikinci aralığa denk geldiğinden bu faaliyetin ikinci başlangıç zamanı seçilmektedir. Böylece, 1. faaliyetin programlanmış başlama zamanı (PB) 1. gün ve programlanır ve tamamlama zamanı (PT) 9. gün olarak belirlenir. Faaliyet 1 programlandıktan sonra, tüm kritik olmayan aktivitelerin erken başlama ve tamamlanma zamanları ve bolluk değerleri güncellenir, faaliyet 1 programlama sıralaması listesinden çıkarılır, ve faaliyet 2 (öncel faaliyeti programlandığından) programlaması mümkün olan ve kritik olmayan faaliyetler listesine eklenir. Daha sonra listeden sıradaki faaliyet seçilip benzer şekilde programlanır. Bu işleme tüm kritik olmayan faaliyetler programlanana kadar devam edilir. Şekil 23, tüm kritik olmayan aktivitelerin Şekil 22' deki gen değerlerine göre deşifre edilip programlanmış halini verilmektedir. Bu iş programından karelerin toplamı (KT) ölçütü için 985 değeri elde edilmiştir.



Şekil 21. Örnek problem

0.240	0.631	0.719	0.853	0.402	0.363
-------	-------	-------	-------	-------	-------

Şekil 22. Örnek probleme ait bir kromozom



Sezgisel iyileştirme işlemi tamamlandıktan sonra, iyileştirilmiş programın başlangıç günlerine göre genler yeniden kodlanıp, kromozom güncelleştirilir. Bunun için iyileştirilmiş programa göre, her kritik olmayan faaliyet için 0'la 1 aralığı, o faaliyetin başlama günü olan alternatiflerinin sayısına bölünerek, ilgili başlama gününe denk gelen aralığın orta noktası alınır. Örneğin, faaliyet 2 için iyileşmiş programda 7 adet başlama günü alternatifi olduğundan, ve aralıkların boyu 0,143 (1/7) olacağından, son başlama günü (15. gün) 0,857-1,000 aralığına denk geleceğinden ve bu aralığın orta noktası 0,929 olacağından, 2. genin değeri 0.929 olarak güncellenir. İyileşmiş programa ait gen kodlaması Şekil 25'de gösterilmiştir

		Günler																														
	EB	GB	PB	PT	S	KK	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23			
1	0	7	1	9	8	2		2	2	2	2	2	2	2	2																	
2	9	15	15	18	3	3															3	3	3									
3	18	18	18	23	5	3																			3	3	3	3	3	3		
4	6	10	10	13	3	4											4	4	4													
5	13	15	13	16	3	2														2	2	2										
6	13	13	13	16	3	3														3	3	3										
7	16	16	16	20	4	4																	4	4	4	4						
8	20	20	20	23	3	4																						4	4	4		
9	0	0	0	6	6	3	3	3	3	3	3	3																				
10	6	6	6	11	5	3							3	3	3	3	3															
11	11	11	11	16	5	3														3	3	3	3	3								
Günlük KK							3	5	5	5	5	5	5	5	5	3	7	7	7	8	8	11	7	7	7	7	7	7	7	7		
Günlük KK'nın karesi							9	25	25	25	25	25	25	25	25	9	49	49	49	64	64	121	49	49	49	49	49	49	49	49	49	49
KKKT 957																																

Şekil 24 İyileştirilmiş program

0.188	0.929	0.500	0.900	0.165	0.500
-------	-------	-------	-------	-------	-------

Şekil 25. İyileşmiş Programın Gen Kodu

### 6.2.3 Tavlama Benzetimi

Tavlama, metallerin fiziksel ve mekanik özelliklerinin iyileştirilmesi için yüksek sıcaklıktan başlayarak azalan bir sıcaklık zarfında metal kristallerinin düzgün dağılımının sağlanması işlemidir. Bu işlemde esinlenerek araştırmacılar tavlama benzetimi optimizasyon algoritmaları geliştirmiştir (Kirkpatrick vd.1983, Cerny 1985). GA makro ölçekte çözümü barındıran uzayı yüzeysel olarak tarayabilmekte ancak mikro ölçekte yakınsamada başarılı

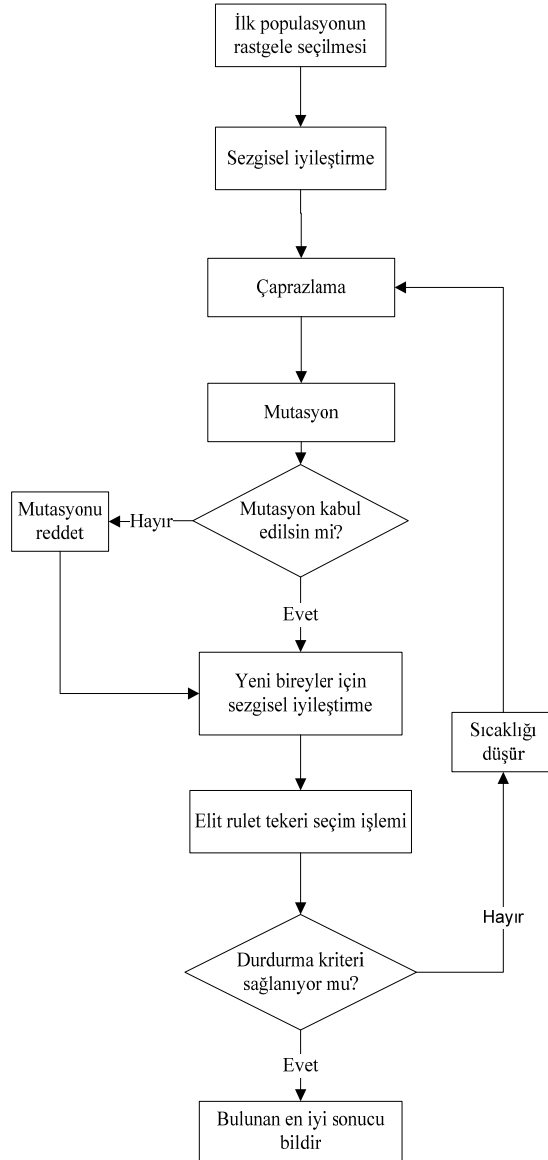
olamamaktadır. Tavlama benzetimi yöntemi ise makro ölçekte çözüm uzayını tarayamamakta ancak mikro ölçekte son derece başarılı sonuçlar elde edebilmektedir. GA ve tavlama benzetiminin ortak kullanımı ile elde edilen melez yöntem hem çözümü içeren uzayı çok iyi biçimde tarayabilmekte, hem de çok başarılı yakınsama yaparak iyi sonuçlar elde edebilmektedir. Bu sebeple araştırmacılar GA'ların tepe tırmanma ve ince ayar yeteneklerinin geliştirilmesi için tavlama benzetimi yöntemi ile melezlenmesini önermektedir (Wang ve Zheng 2001, Chen vd. 2005, Hwang ve He 2006, Han ve Sun 2006, Bettemir 2009). Bu araştırmalar doğrultusunda öncelik-temelli sezgisel algoritma ile melezlenmiş GA ikinci iyileştirme stratejisi olarak tavlama benzetimi (TB) yöntemi ile melezlenmiştir. Tavlama benzetimi (TB) yönteminin memetik algoritma (MA) ile melezlenmesi için Hwang ve He (2006)'in geliştirdiği yöntem baz alınmıştır. Tavlama benzetimli memetik algoritma (TBMA) her iyi mutasyonu (mutasyon sonrasında uygunluk fonksiyonu değerinin iyileşmesi) kabul eder. Mutasyon öncesine göre daha olumsuz sonuçlar veren kötü mutasyonların kabul edilip edilmemesi ise bir karar fonksiyonunca belirlenir. Böylece iterasyonların sonuna doğru kötü olan mutasyonların kabul edilme oranı azaltılarak problemin iyi bir sonuca yakınsaması amaçlanmaktadır. Tavlama benzetimi algoritması MA'nın çözüm uzayı tarama işlemini kontrol ederek, elde edilen popülasyonun işlem sonlarına doğru daha az kötü mutasyona uğramasını, dolayısı ile işlem sonlarına doğru bulunan iyi çözümlerin daha çok muhafaza edilmesini sağlar.

#### 6.2.4 Çaprazlama, Mutasyon, ve Tavlama Benzetimi

TBMA'da çaprazlama ve mutasyon operatörleri kullanılarak yeni bireyler üretilmektedir. Bu doğrultuda kritik olmayan faaliyet sayısı 10 ve altında olan problemler için tek noktalı, kritik olmayan faaliyet sayısı 10'un üzerinde olan problemler için çift noktalı çaprazlama uygulanmaktadır. Mutasyon işlemi, rastgele bir kromozom ve onun içinden bir rastgele gen seçilip, değerinin rastgele belirlenmiş bir değerle değiştirilmesiyle yapılmaktadır. TBMA'da mutasyon işlemi, tavlama benzetimi yöntemi kullanılarak ve adaptif mutasyon oranı esaslı bir soğutma programına dayanarak yapılmaktadır. Bu şekilde mutasyondan elde edilen bireyler daha iyi bir amaç fonksiyon değerine sahip olduğu zaman direkt kabul edilmektedir. Ancak, yeni bireyin amaç fonksiyon değeri iyileşmemişse, 37 nolu formüle göre kabul edilir veya red edilir.

$$r \leq e^{-\frac{(f_i - f_i^t) \cdot B}{f_i \cdot t}} \quad (37)$$

Bu formülde  $r$ , 0 ve 1 arasında rastgele seçilen bir sayıdır.  $f_i$  mutasyon yapılmadan önceki amaç fonksiyonu değeri,  $f_i'$  mutasyon yapıldıktan sonraki amaç fonksiyonu değeri,  $b$  Boltzman sabiti,  $t$  ise sıcaklıktır ve algoritmada iterasyon sayısının tersine eşittir. Adaptif mutasyon oranının asıl amacı yalın bir genetik algoritmaya göre daha etkili bir tarama yapmak ve algoritmanın lokal bir optimal değere takılmamasını sağlamaktır. Formül 37'deki tavlama değeri, iterasyon sayısı arttıkça küçülmektedir. Böylece iterasyonların sonuna doğru kötü olan mutasyonların kabul edilme oranı azaltılarak problemin iyi bir sonuca yakınsamasının sağlanması hedeflenmektedir. TBMA'nın akış diyagramı Şekil 26'da gösterilmiştir.



Şekil 26. TBMA'nın akış diyagramı

## 7. ÜST SEZGİSEL KAYNAK Dengeleme Algoritmalarının Performansının Belirlenmesi İçin Yapılan Testler

Proje kapsamında geliştirilen tavlama benzetimli memetik algoritma (TBMA)'nın performansının belirlenmesi için dört farklı test yapılmıştır. İlk iki testte, hem GA üzerinde proje kapsamında yapılan geliştirmelerin sonucunu değerlendirmek, hem de TBMA'ın performansını belirlemek amacıyla TBMA, yalın GA ve iyileştirilmiş B&K (Burgess ve Killebrew 1962) sezgiseli kullanılmıştır. İlk testte 30 faaliyete kadar olan ve optimal çözümü proje kapsamında geliştirilen kesin çözüm yöntemleri ile elde edilen 36 dört kaynaklı problem kullanılmıştır. İkinci testte ise 300 faaliyete kadar olan toplam 1920 dört kaynaklı problem kullanılmıştır. Üçüncü testte, TBMA'nın performansı MS Project 2010 ve Primavera 6.7 yazılımlarının kullandığı toplam 9 sezgiselle kıyaslanmıştır. Son testte ise KDP konusundaki en güncel ve kapsamlı çalışmalardan olan Ponz-Tienda v. d. (2013) tarafından geliştirilen adaptif genetik algoritmasının (AGA) performansı ile TBMA'nın performansı karşılaştırılmıştır.

Proje kapsamında geliştirilen yalın GA ve TBMA Visual Studio 2010 yazılımı ile C# dili kullanılarak kodlanmıştır. C# dili Microsoft Project yazılımı ile entegrasyona imkan tanıdığı için tercih edilmiştir. Tüm testler, 3.00 GHz işlemci hızına ve 6 GB RAM kapasitesine sahip, Intel Core2 E8400 CPU çekirdekli ve 64 bit işletim sistemli bir bilgisayarda gerçekleştirilmiştir.

### 7.1. Küçük Ölçekli Problem Setleriyle Yapılan Testler

TBMA'nın performansının belirlenmesi için yapılan ilk testte 5.4. Bölümde değinilen Çağdaş Mutlu (2010)'nun Yüksek Lisans tezi için geliştirdiği 21 problem ve bu problemlere ilaveten Kolisch, Sprecher, ve Drexl'in oluşturduğu dört kaynaklı problemlerden (Kolisch v.d. 1995) rastgele seçilen 15 adet 30 faaliyetli (J30) problemden oluşan toplam 36 adet küçük ölçekli problem seti kullanılmıştır. Testlerde TBMA birlikte yalın GA ve iyileştirilmiş B&K (Burgess ve Killebrew 1962) sezgiseli de dahil edilmiş, TBMA'nın 500,000 iş programı sonucunda elde ettiği sonuç verilmiş ve karelerin toplamı (KT), ortalama kaynak miktarından mutlak sapma miktarı (OKMSM), ve azami kaynak kullanımı ve atıl kaynak günü (MRD+RID) ölçütleri kullanılmıştır. Testlerde kullanılan ölçütlerin ayrıntıları 3.1.4. bölümde verilmiştir. Kıyaslamaların eşit koşullarda yapılabilmesi için testlerde iyileştirilmiş B&K (Burgess ve Killebrew 1962) sezgiseli (B&K2) kullanılmıştır. Standard B&K algoritması sadece tek bir rastgele oluşturulmuş popülasyon için çalışmaktadır. Testlerde kullanılan B&K2 algoritması belirli bir süre içerisinde oluşturulan çoklu rastgele popülasyonlardan (faaliyet numarası



öncelikli) elde edilen en iyi değeri vermektedir. Yalın GA ve B&K2 algoritmaları için ise her problem için, TBMA'nın 500,000 iş programı denemesi için kullanılan işlem süresi kullanılmıştır. Kıyaslamalarda problemler için elde edilen optimal değerden sapma miktarları (OS %) kullanılmıştır. KT ve OKMSM ölçütleri için optimal çözümleri proje kapsamında geliştirilen doğrusal tamsayılı eniyileme modeli ve MS Excel-GAMS/CPLEX uygulaması kullanılarak çözdürülmüştür. MRD+RID ölçütü için doğrusal tamsayılı modellerine göre daha iyi sonuç veren ve proje kapsamında geliştirilen dal ve sınır algoritması kullanılmıştır. Dal ve sınır algoritması ile 5 saatlik süre içerisinde 36 problemden 28'i için optimal çözüm elde edilmiştir, kıyaslamalarda MRD+RID ölçütü için bu 28 problem kullanılmıştır. Problemlere ait optimal çözümler ve TBMA, yalın GA ve B&K2 sezgiselinin elde ettiği sonuçlar <http://www2.ce.metu.edu.tr/~sonmez/research/111M140> adresli web sitesinde araştırmacıların erişimine sunulmuştur. Tablo 8, Tablo 9, ve Tablo 10' da ise üç yöntemin her problem için optimal değerden sapma miktarları (OS %) karelerin toplamı (KT), ortalama kaynak miktarından mutlak sapma miktarı (OKMSM), ve azami kaynak kullanımı ve atıl kaynak günü (MRD+RID) ölçütleri için verilmiştir.

KT ölçütü için her üç yöntemde de küçük ölçekli problemler için optimal veya optimala yakın sonuç elde etmiştir (Tablo 8). Bununla birlikte 36 problem için ortalama %0.2 optimalden sapma değeri ile TBMA en başarılı sonucu elde etmiştir. Aynı problemler için yalın GA ve B&K2 sezgiseli %0.8 ve % 1.7 ortalama optimalden sapma değerlerini elde etmiştir.

Tablo 9 ve Tablo 10 da verilen OKMSM ve MRD+RID ölçütlerine ait sonuçlar TBMA'nın küçük ölçekli KDP problemlerini çözümedeki başarısını teyit etmektedir. TBMA 2 ile 3 saniye arasındaki bir işlem süresi içerisinde KT ölçütünde olduğu gibi OKMSM ve MRD+RID ölçütleri içinde küçük ölçekli KDP problemler için optimal veya optimala çok yakın çözümler bulmuş, ve %1 civarında ortalama optimalden sapma değerlerini elde etmiştir. Aynı problemler için TBMA ile aynı işlem süresi dahilinde yalın GA ve B&K2 sezgiseli %2 ile %5 arasında bir ortalama optimalden sapma değerleri elde edebilmiştir.

## **7.2. Kolish ve Sprecher (1997) ile Debels ve Vanhoucke (2007) Problem Setleriyle**

### **Yapılan Testler**

TBMA'nın performansının belirlenmesi için yapılan ikinci testte ise 480 adet 30 faaliyetli, 480 adet 60 faaliyetli, 480 adet 120 faaliyetli (Kolish ve Sprecher 1997), ve 480 adet 300 (Debels ve Vanhoucke 2007)' faaliyetli problemden meydana gelen toplam 1920 dört kaynaklı

Tablo 8. Küçük ölçekli problem seti ve KT ölçütü ile yapılan testlerin sonuçları

No	Problem	Aktiv.	Süre(Saniye)	MA OS(%)	GA OS(%)	B&K2 OS(%)
1	El Rayes ve Jun (2009)	22	2.3	0.0	0.4	0.0
2	Stevens (1990 Sayfa 97)	17	2.3	0.0	0.1	0.4
3	Stevens (1990 Sayfa 172)	21	2.1	0.0	0.6	0.8
4	Harris (1990)	13	1.3	0.0	0.0	0.0
5	Leu (2000)	15	1.8	0.0	0.2	0.1
6	Son & Skibniewski (1999)	15	3.2	0.0	0.0	0.0
7	Son & Skibniewski (1999)	13	1.8	0.0	0.0	0.1
8	Generated II (Mutlu 2010)	8	1.7	0.0	0.0	0.0
9	Generated III (Mutlu 2010)	16	2.0	0.0	0.0	0.0
10	Generated IV (Mutlu 2010)	20	2.0	0.0	0.0	0.0
11	Generated V (Mutlu 2010)	15	1.7	0.0	0.0	0.0
12	Generated VI (Mutlu 2010)	16	2.0	0.0	0.0	0.4
13	Generated VII (Mutlu 2010)	18	2.4	0.0	0.6	0.0
14	Mubarak (2004 Sayfa 61)	16	2.4	0.0	0.0	0.0
15	Mubarak (2004 Sayfa 67)	13	2.1	0.0	0.0	0.0
16	Mubarak (2004 Sayfa 217)	10	0.0	0.0	0.0	0.0
17	Newitt (2005 Sayfa 82)	18	2.2	0.0	0.0	0.6
18	Newitt (2005 Sayfa 121)	14	0.0	0.0	0.0	0.0
19	Hinze (2004 Sayfa 152)	17	1.7	0.0	0.0	0.0
20	Demeulemeester (2002)	12	1.6	0.0	0.0	0.0
21	Easa (1989)	7	0.0	0.0	0.0	0.0
22	j301_8	32	2.7	0.0	8.5	2.2
23	j305_5	32	3.2	0.4	1.7	7.5
24	j306_5	32	3.2	0.5	0.5	0.6
25	j307_7	32	3.1	3.5	4.7	0.8
26	j3010_2	32	3.3	0.7	2.0	7.7
27	j3010_9	32	3.3	0.8	1.7	5.4
28	j3011_8	32	3.4	0.1	0.1	9.8
29	j3013_7	32	3.4	0.3	1.6	1.7
30	j3018_2	32	2.8	0.0	0.0	0.0
31	j3023_6	32	2.9	0.0	0.0	0.7
32	j3028_10	32	3.3	0.1	1.4	3.3
33	j3035_3	32	2.8	0.0	0.0	2.1
34	j3036_1	32	3.1	0.0	1.7	8.5
35	j3039_7	32	3.2	0.1	1.6	8.9
36	j3044_7	32	3.3	0.0	0.0	0.3
Ortalama:			2.3	0.2	0.8	1.7

Tablo 9. Küçük ölçekli problem seti ve OKMSM ölçütü ile yapılan testlerin sonuçları

No	Problem	Aktiv.	Süre(Saniye)	MA OS(%)	GA OS(%)	B&K2 OS(%)
1	El Rayes ve Jun (2009)	22	2.7	0.0	0.0	0.0
2	Stevens (1990 Sayfa 97)	17	2.5	0.0	0.0	1.8
3	Stevens (1990 Sayfa 172)	21	2.3	0.0	0.0	1.8
4	Harris (1990)	13	1.6	0.0	0.0	0.0
5	Leu (2000)	15	1.6	0.0	0.0	1.6
6	Son & Skibniewski (1999)	15	2.6	0.0	0.0	0.0
7	Son & Skibniewski (1999)	13	1.7	0.0	0.0	2.0
8	Generated II (Mutlu 2010)	8	1.9	0.0	0.0	0.0
9	Generated III (Mutlu 2010)	16	1.9	0.0	0.0	0.0
10	Generated IV (Mutlu 2010)	20	2.1	0.0	0.0	0.0
11	Generated V (Mutlu 2010)	15	1.6	0.0	0.0	0.0
12	Generated VI (Mutlu 2010)	16	2.0	0.0	0.0	0.0
13	Generated VII (Mutlu 2010)	18	2.4	0.0	0.0	0.0
14	Mubarak (2004 Sayfa 61)	16	2.4	0.0	0.0	0.0
15	Mubarak (2004 Sayfa 67)	13	2.0	0.0	0.0	0.0
16	Mubarak (2004 Sayfa 217)	10	0.0	0.0	0.0	0.0
17	Newitt (2005 Sayfa 82)	18	2.1	0.0	0.0	1.0
18	Newitt (2005 Sayfa 121)	14	0.0	0.0	0.0	0.0
19	Hinze (2004 Sayfa 152)	17	1.8	0.0	0.0	0.0
20	Demeulemeester (2002)	12	1.6	0.0	0.0	5.3
21	Easa (1989)	7	0.0	0.0	0.0	0.0
22	j301_8	32	2.8	0.0	0.0	2.8
23	j305_5	32	3.3	0.0	1.1	15.0
24	j306_5	32	3.5	0.0	6.4	17.7
25	j307_7	32	3.2	10.4	12.8	32.2
26	j3010_2	32	3.4	10.7	10.7	25.2
27	j3010_9	32	3.4	2.8	15.0	15.0
28	j3011_8	32	3.7	1.3	11.6	11.6
29	j3013_7	32	3.5	11.0	12.3	20.6
30	j3018_2	32	2.9	0.0	0.0	2.6
31	j3023_6	32	3.0	0.0	0.0	1.8
32	j3028_10	32	3.6	0.6	3.2	29.5
33	j3035_3	32	2.9	0.0	0.0	0.3
34	j3036_1	32	3.2	0.0	0.0	6.5
35	j3039_7	32	3.3	1.1	1.4	2.0
36	j3044_7	32	3.4	0.0	0.0	1.8
		Ortalama:	2.4	1.1	2.1	5.5

Tablo 10. Küçük ölçekli problem seti ve MRD+RID ölçütü ile yapılan testlerin sonuçları

No	Problem	Aktiv.	Süre(Saniye)	MA OS(%)	GA OS(%)	B&K2 OS(%)
1	El Rayes ve Jun (2009)	22	4.17	0.0	9.0	6.5
2	Stevens (1990 Sayfa 97)	17	3.73	0.0	0.0	21.0
3	Stevens (1990 Sayfa 172)	21	3.50	0.0	0.0	9.9
4	Harris (1990)	13	2.29	0.0	0.0	13.8
5	Leu (2000)	15	2.54	0.0	2.3	2.3
6	Son & Skibniewski (1999)	15	4.00	0.0	0.0	0.0
7	Son & Skibniewski (1999)	13	2.81	0.0	0.0	0.0
8	Generated II (Mutlu 2010)	8	2.51	0.0	0.0	0.0
9	Generated III (Mutlu 2010)	16	2.90	0.0	0.0	2.8
10	Generated IV (Mutlu 2010)	20	3.06	0.0	0.0	0.0
11	Generated V (Mutlu 2010)	15	2.65	0.0	1.3	1.3
12	Generated VI (Mutlu 2010)	16	3.42	0.0	0.0	6.7
13	Generated VII (Mutlu 2010)	18	3.97	0.0	0.7	14.0
14	Mubarak (2004 Sayfa 61)	16	4.06	0.0	0.0	1.3
15	Mubarak (2004 Sayfa 67)	13	3.67	0.0	0.0	0.0
16	Mubarak (2004 Sayfa 217)	10	0.01	0.0	12.1	0.0
17	Newitt (2005 Sayfa 82)	18	3.59	0.0	2.9	0.0
18	Newitt (2005 Sayfa 121)	14	0.01	0.0	8.8	0.0
19	Hinze (2004 Sayfa 152)	17	2.59	0.0	0.0	0.0
20	Demeulemeester (2002)	12	2.76	0.0	0.0	4.3
21	Easa (1989)	7	0.01	0.0	0.0	0.0
22	j307_7	32	4.80	17.4	1.7	9.8
23	j3010_2	32	4.87	6.4	13.2	17.7
24	j3018_2	32	4.44	0.0	0.0	0.0
25	j3023_6	32	4.54	0.3	16.0	3.5
26	j3035_3	32	4.45	0.0	1.3	15.7
27	j3039_7	32	4.23	6.0	6.0	6.0
28	j3044_7	32	4.87	0.0	4.6	10.5
Ortalama:			3.23	1.1	2.9	5.3

problem kullanılmıştır. Testlerde 7.1. bölümde olduğu gibi TBMA birlikte yalın GA ve iyileştirilmiş B&K (Burgess ve Killebrew 1962) sezgiseli de dahil edilmiş, TBMA'nın 500,000 iş programı sonucunda elde ettiği sonuç verilmiş ve karelerin toplamı (KT), ortalama kaynak miktarından mutlak sapma miktarı (OKMSM), ve azami kaynak kullanımı ve atıl kaynak günü (MRD+RID) ölçütleri kullanılmıştır. Yalın GA ve B&K2 algoritmaları için ise her problem için, TBMA'nın 500,000 iş programı denemesi için kullanılan işlem süresi kullanılmıştır. Kıyaslamalarda problemler için elde edilen en iyi değerden (üst sınırdan) sapma miktarları (%) kullanılmıştır. B&K2 sezgiseli, yalın GA ve TBMA'nın 1920 problem için bulunduğu çözüm, her problem için işlem süreleri, ve üst sınırdan sapma (ÜSS) değerleri

<http://www2.ce.metu.edu.tr/~sonmez/research/111M140> adresli web sitesinde araştırmacıların erişimine sunulmuştur. Tablo 11’de ise testlerin sonuçları özetlenmiştir. Tablo 11’de verilen karelerin toplamı ölçütü ve 30, 60, ve 120 faaliyetli problemlerde TBMA’nın 500,000 iş programı denemesi için kullanılan işlem süresinin 4. Dönem Raporu’nda verilen sürelerle göre daha kısa olmasının sebebi, son dönemde TBMA’nın hızlanması için yapılan iyileştirmelerdir.

Tablo 11’de 30, 60, 120 ve 300 faaliyetli her problem seti için B&K2 sezgiseli, yalın GA ve TBMA’nın karelerin toplamı (KT), ortalama kaynak miktarından mutlak sapma miktarı (OKMSM), ve azami kaynak kullanımı ve atıl kaynak günü (MRD+RID) ölçütleri için ortalama üst sınırdan sapma (ÜSS) değerleri ve ortalama işlem süreleri verilmiştir. Tablo 11’de verilen sonuçlar değerlendirildiğinde proje kapsamında geliştirilen TBMA üst sezgisel algoritmasının B&K2 sezgiseli ve yalın GA’ya göre KDP’nin çözümünde performansının çok daha iyi olduğu görülmekte, ve proje hedeflerinden bir tanesine daha ulaşıldığı teyit edilmektedir. TBMA yapılan testlerde bütün problem setleri ve ölçütler için en iyi ortalama sonucu vermiş ve % 0.0 ile %1.6 arasında bir ortalama üst sınırdan sapma (ÜSS) değerleri elde edebilmiştir. Bununla birlikte, B&K2 sezgiselinin ÜSS değeri bazı problem setlerinde %23.0’a kadar çıkmakta, benzer şekilde yalın GA’nın ÜSS değeri %18.2’ye ulaşmaktadır.

Tablo 11. 1920 problem ile yapılan testlerin sonuçları

Ölçüt	Set	ÜSS (%)	ÜSS (%)	ÜSS (%)	Süre (Sn.)
		B&K2	GA	TBMA	
KT	J30(480)	3.4	0.8	0.0	3.2
	J60(480)	3.1	1.7	0.0	6.1
	J120(480)	2.1	2.2	0.1	11.0
	J300(480)	0.4	0.4	0.0	37.1
OKMSM	J30(480)	9.7	3.0	0.1	3.3
	J60(480)	9.8	6.4	0.3	6.2
	J120(480)	8.5	8.3	0.5	11.1
	J300(480)	13.5	15.3	1.6	38.0
MRD+RID	J30(480)	23.0	11.4	1.1	4.9
	J60(480)	22.6	17.7	1.2	8.3
	J120(480)	21.4	18.2	1.2	13.5
	J300(480)	13.2	7.1	0.4	40.5

### 7.3. TBMA’ nın Microsoft Project ve Primavera ile Karşılaştırılması için Yapılan Testler

TBMA’nın B&K2 sezgiseli, yalın GA ile kıyaslanmasını takiben, TBMA’nın performansı, inşaat projelerinde yaygın olarak kullanılan MS Project 2010 ve Primavera 6.7 yazılımlarının

kullandığı toplam 9 sezgiselin performansı ile kıyaslanmıştır. Bu testlerde 7.1 Bölümde değinilen 15 adet 30 faaliyetli (J30) probleme ilaveten, Kolisch, Sprecher, ve Drexl'in oluşturduğu dört kaynaklı problemlerden (Kolisch v.d. 1995) rastgele seçilen 15 adet 60 faaliyetli (J60), ve 15 adet 120 faaliyetli (J120) problemden oluşan toplam 45 problemlik set kullanılmıştır. Test seti, ilk olarak, MS Project 2010'la "Standart Priority" (STD) koşuluna göre, Primavera 6.7'yle aktivite sırası (ID), toplam bolluk (TF), erken başlama (ES), geç tamamlama (LF) kriterlerinin artan (ascending-A) ve azalan (descending-D) yöntemleri kullanılarak çözdürülmüştür. Testlerde TBMA'nın 500,000 iş programı sonucunda elde ettiği sonuç verilmiş ve ölçüt olarak ortalama kaynak miktarından mutlak sapma miktarı kullanılmıştır. 30 faaliyetli problemler proje kapsamında geliştirilen doğrusal tamsayıli eniyileme modeli ve MS Excel-GAMS/CPLEX uygulaması kullanılarak çözdürülmüş ve bu problemler için kıyaslamalarda optimal çözümden sapma (%) miktarları kullanılmıştır. 60 ve 120 faaliyetlik problemler için ise elde edilen en iyi değerden (üst sınırdan) sapma miktarları (%) kullanılmıştır.

Testlere ait sonuçların özeti Tablo 12' de verilmiştir. Tablo'dan da görüleceği üzere TBMA'nın performansı testlerde kullanılan 9 sezgiselle göre çok daha iyidir. 30 faaliyetlik problemler için TBMA'nın optimal çözümden sapma miktarı sadece % 0.5' dir. 30 faaliyetlik problem için sezgiseller arasında en iyi olan erken başlama-azalan (ESD) sezgiselinin ise optimal çözümden sapma miktarı % 41.8' dir. 60 ve 120 faaliyetlik toplam 30 problemden 29'unda TBMA en iyi sonucu elde etmiştir. Diğer 9 sezgisel 30 faaliyetli problemlerde olduğu gibi oldukça kötü sonuçlar elde etmişlerdir. TBMA'nın 60 ve 120 faaliyetli problem setleri için ortalama %0.0 ve %0.1 ÜSS değerlerine rağmen, sezgiseller 60 faaliyetli problemler için en iyi %46.3, 120 faaliyetli problemler içinse en iyi %41.7 ortalama ÜSS değerlerini elde edebilmişlerdir. 30, 60, ve 120 faaliyetli problemler için MS Project 2010 ve Primavera 6.7 yazılımlarının elde ettiği kötü sonuçlar KDP konusundaki TBMA gibi yöntemlerin önemini destekler niteliktedir.

Tablo 12. TBMA'nın MS Project ve Primavera ile karşılaştırılması

Problem Seti	MSP			Primavera 6.7						111M140	
	STD	IDA	IDD	TFA	TFD	ESA	ESD	LFA	LFD	TBMA	Süre (S.)
J30 (15)	75.4	42.9	48.9	48.0	43.6	45.0	41.8	45.9	44.4	0.5	4.7
J60 (15)	89.3	59.1	62.0	49.4	54.0	61.3	52.4	46.3	57.5	0.0	8.3
J120 (15)	100.3	58.4	53.7	41.7	61.1	69.5	56.3	48.9	64.1	0.1	15.8
Ortalama	88.3	53.5	54.9	46.4	52.9	58.6	50.1	47.0	55.3	0.2	9.6

#### 7.4. TBMA' nın AGA ile Karşılaştırılması için Yapılan Testler

TBMA' nın performansının belirlenmesi amacıyla yapılan son testte proje önerisini aşamasında yer almayan ve proje kapsamı dahilinde gerçekleştirilen üst sezgisel algoritmaların geliştirilmesi aşamasını takiben yayınlanan ve KDP konusundaki en son ve en kapsamlı çalışmalardan olan Ponz-Tienda vd. (2013) tarafından geliştirilen adaptif genetik algoritmasıyla (AGA)'nın performansı TBMA' nın performansı ile kıyaslanmıştır. AGA 'nın performansını belirlemek için Ponz-Tienda v.d. (2013) 7.2. Bölümde de değinilen ve Kolisch, Sprecher, ve Drexl'in oluşturduğu (Kolisch v.d. 1995) 480 adet 30 faaliyetli (J30), 480 adet 60 faaliyetli (J60), ve 480 adet 120 faaliyetli (J120)'den meydana gelen toplam 1440 problemlilik bir problem seti ve karelerin toplamı ölçütü kullanılmıştır. Bu bölümde, proje kapsamında geliştirilen TBMA'nın performansı, 1440 problemlilik test seti kullanılarak, Ponz-Tienda v.d. (2013) tarafından geliştirilen adaptif genetik algoritmasının (AGA) algoritmasının performanslarıyla karşılaştırılmıştır. Testlerde TBMA'nın 500,000 iş programı sonucunda elde ettiği sonuç verilmiş ve ölçüt olarak karelerin toplamı (KT) ölçütü kullanılmıştır. 30 faaliyetli 480 problemlilerden 475 tanesi proje kapsamında geliştirilen doğrusal tamsayıli eniyileme modelin ve MS Excel-GAMS/CPLEX uygulaması kullanılarak çözdürülmüştür ve bu optimal çözümler algoritmaların performansının değerlendirilmesinde kullanılmıştır. Aynı zamanda KDP ile ilgili literatürde yaygın olarak kullanılan kritik yol yöntemi ve karelerin toplamı ölçütü için optimal çözümlü büyük bir problem seti oluşturulmuştur. 5 problem için ise 5 saatlik süre içerisinde çözüm elde edilememiştir. Kıyaslamalarda problemler için elde edilen en iyi değerden (üst sınırdan) sapma miktarları (%) kullanılmıştır. AGA ve TBMA'nın her problem için işlem süreleri, ve üst sınırdan sapma (ÜSS) değerleri 30 faaliyetli 475 problem için proje kapsamında geliştirilen MS Excel-GAMS/CPLEX uygulaması ile elde edilen optimal çözümler <http://www2.ce.metu.edu.tr/~sonmez/research/111M140> adresli web sitesinde araştırmacıların erişimine sunulmuştur. Tablo 13 de ise testlerin sonuçları özetlenmiştir.

Tablo 13' de özeti verilen test sonuçlarında, J30 problem seti için TBMA %0,2 ÜSS değerine sahip olarak problemlerin çoğuna optimal ve ya da optimal'a çok yakın olan sonuçlar üretmiştir. TBMA toplam 475 optimal sonucu olan problemlerden 232 problem için optimal sonucu bulmayı başarmıştır. AGA ise %0,7 ÜSS değeri ile ve 475'den 76 problemlin optimal sonucuna bularak J30 problemlerinin çözümünde ikinci sırada yer almaktadır. AGA'nın denemeleri, 3.6 GHz işlemci hızına sahip Intel Core i7 CPU çekirdekli bir bilgisayarda gerçekleştirilmiş ve J30 problemleri için ortalama çözüm süresi 15 saniye olarak verilmiştir

(Ponz-Tienda vd. 2013). Tablo 13’de görüldüğü gibi TBMA, AGA karşısında, daha düşük hızlı işlemcili (3.00 GHz) bir bilgisayar’da çalıştırılmasına rağmen, daha kısa bir sürede (J30 seti için ortalama 3,2 saniye), daha iyi sonuçlar elde etmeyi başarmıştır. 7.2. bölümde değinildiği gibi, TBMA’nın 500,000 iş programı denemesi için kullanılan işlem süresinin 4. Dönem Raporu’nda verilen sürelerle göre daha kısa olmasının sebebi, son dönemde TBMA’nın hızlanması için yapılan iyileştirmelerdir.

Diğer setler için J60 problemlerinde %0,0 ve J120 problemlerinde %0,1 ÜSS değeri elde eden TBMA bir çok problem için en iyi değeri bulmayı başarmıştır. AGA J60 problem setinde %2,3 ÜSS değeri, J120 problem setinde ise %3,7 ÜSS değeri elde edilmiştir. KDP konusundaki en güncel ve kapsamlı çalışmalardan olan Ponz-Tienda v. d. (2013) tarafından geliştirilen adaptif genetik algoritmasının (AGA) ile yapılan karşılaştırmalar, proje kapsamında geliştirilen TBMA’ nın KDP’nin çözümünde etkin ve başarılı bir algoritma olduğunu teyit etmektedir.

Tablo 13. TBMA’nın AGA(Ponz-Tienda ve vd. 2013). ile karşılaştırılması

Set	AGA (Tienta ve d. 2013)			TBMA-111M140		
	ÜSS (%)	Opt. Sonuç Sayısı	Süre (S.)	ÜSS (%)	Opt. Sonuç Sayısı	Süre (S.)
J30 (480)	0.7	76	15	0.2	232	3.2
J60 (480)	2.3	-	-	0.0	-	6.1
J120 (480)	3.7	-	-	0.1	-	11.0
Ortalama:	2.2			0.1		19.5

## 8. GELİŞTİRİLEN ÜST SEZGİSELİN MS PROJECT’e ENTEGRASYONU

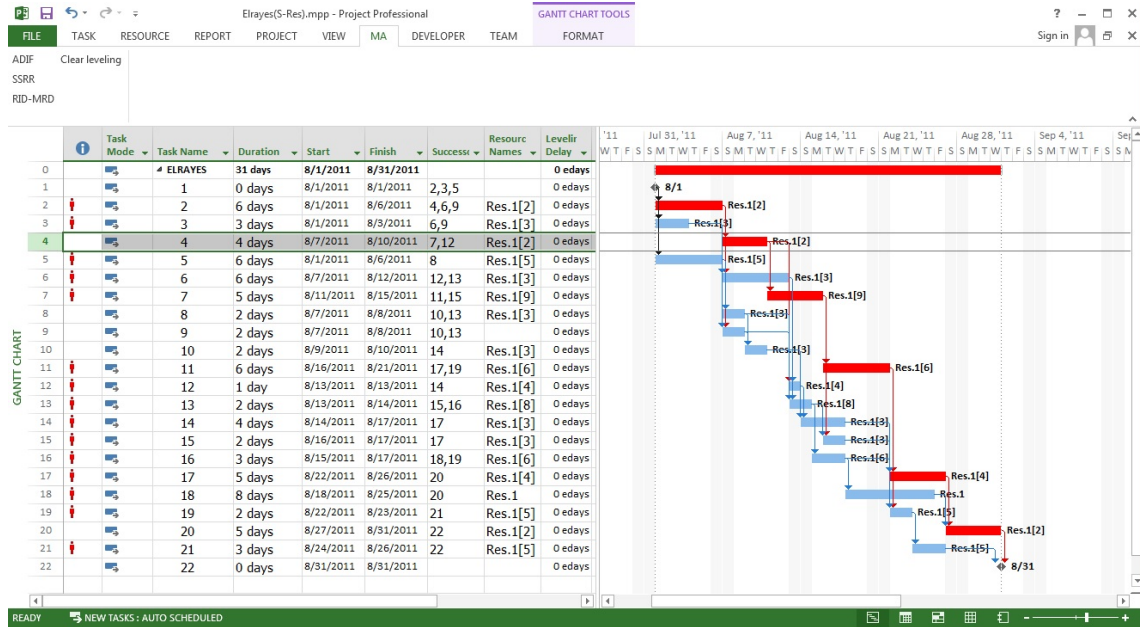
Proje önerisinde geliştirilecek üst-sezgisel algoritmanın yaygın etkisinin artırılması için Microsoft Project 2010 yazılımı ile entegrasyonun planlanmıştır. Ancak proje başladıktan sonra Microsoft Project bu yazılımın yeni sürümünü sunmuştur. Bu sebeple, proje kapsamında geliştirilen tavlama benzetimli memetik algoritma (TBMA) Microsoft Project Professional 2013’e yazılımla ilgili software development kit (SDK) kullanılarak bir eklenti (add-in) olarak eklenebilecek şekilde entegre edilmiştir. Bu doğrultuda Microsoft Project’ in proje kapsamında geliştirilen TBMA, faaliyetlere ait numaraları, öncel faaliyetleri, faaliyet çalışma sürelerini ve kaynak ihtiyaçları Microsoft Project’in veri tabanından girdi olarak alıp, kaynak dengelemesini yaptıktan sonra, erken başlangıç (EB) zamanına göre erteleme



miktarını, Microsoft Project'in veritabanına çıktı olarak aktarmak yoluyla Microsoft Project ile arasındaki veri alışverişini gerçekleştirilmektedir. Proje kapsamında gerçekleştirilen TBMA'a ait eklenti karelerin toplamı (KT), ortalama kaynak miktarından mutlak sapma miktarı (OKMSM), ve azami kaynak kullanımı ve atıl kaynak günü (MRD+RID) ölçütleri için kaynak dengelemesi imkanı sunmaktadır. Eklentinin ayrıntılarına ilerleyen bölümlerde değinilmiştir.

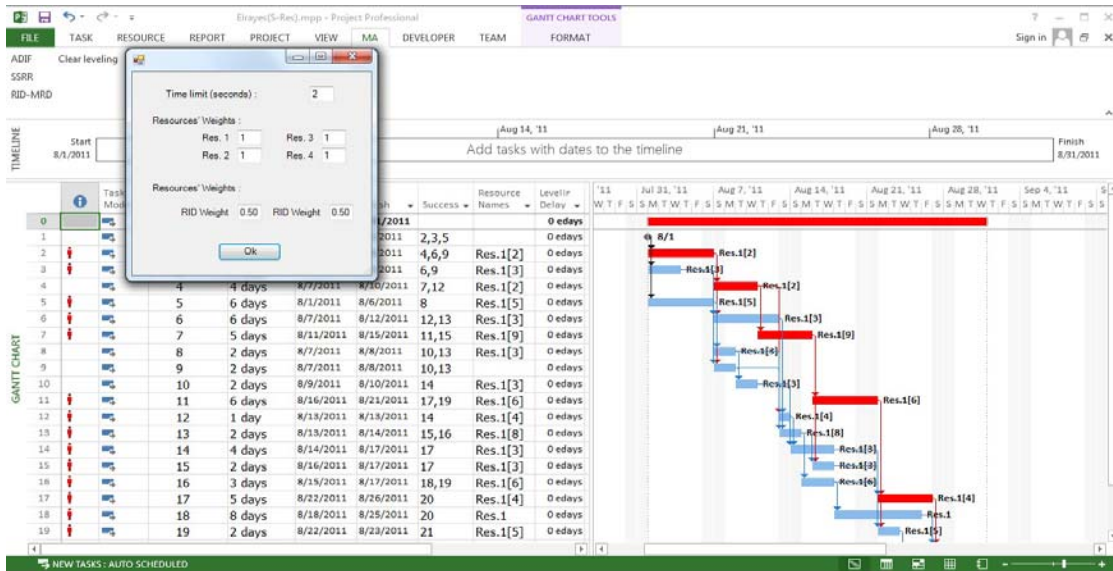
## 8.1. TBMA Microsoft Project Professional 2013 Eklentisi

Proje kapsamında gerçekleştirilen TBMA'a ait eklenti MS Project 2013'e diğer eklentiler gibi yüklendikten sonra MS Project'e ait menünün sol tarafında kaynak dengelemesi için üç farklı seçenek sunmaktadır. Şekil 27'de eklenti El-Rayes ve Jun (2009)'a ait problem kullanılarak gösterilmiştir. Bu seçeneklerde ADIF ortalama kaynak miktarından mutlak sapma miktarına göre, SSRR karelerin toplamına göre, RID-MRD ise azami kaynak kullanımı ve atıl kaynak günü ölçütüne göre kaynak dengelemesi yapmaktadır. Clear leveling seçeneği ise kaynak dengelemesi yapıldıktan sonra istenirse kritik yol yöntemine ait iş programını elde etmek içindir. TBMA'a ait eklenti Microsoft Project yazılımında kritik yol yöntemini kullanarak iş programı oluşturulduktan sonra üç ölçütten birine göre kaynak dengeleme imkanı vermektedir.

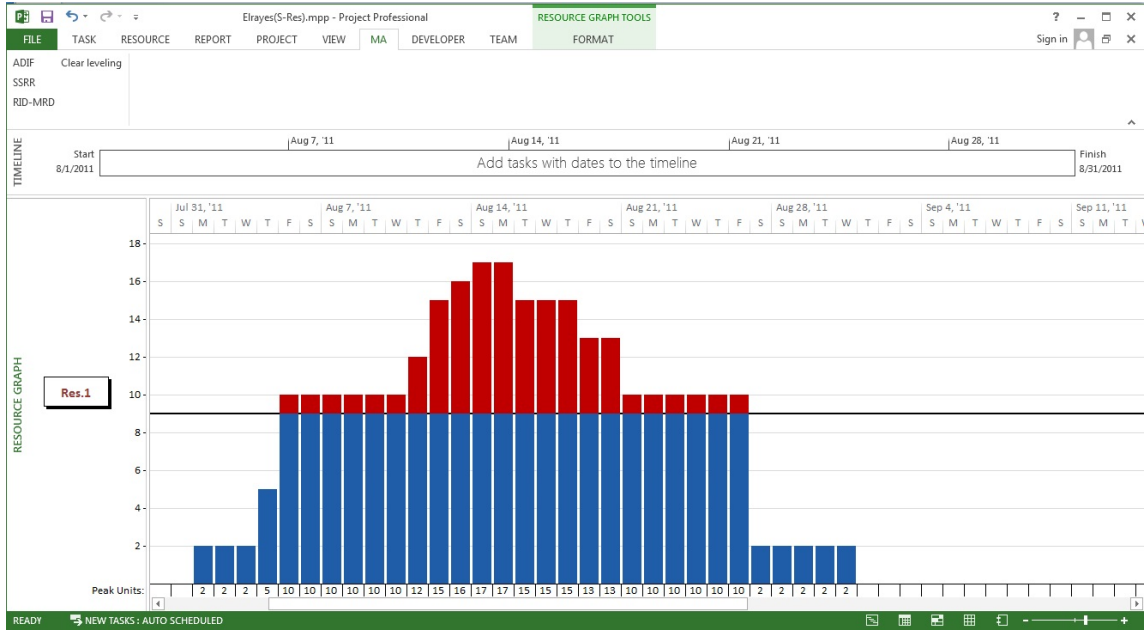


Şekil 27. TBMA'a ait eklentinin kaynak dengeleme seçenekleri

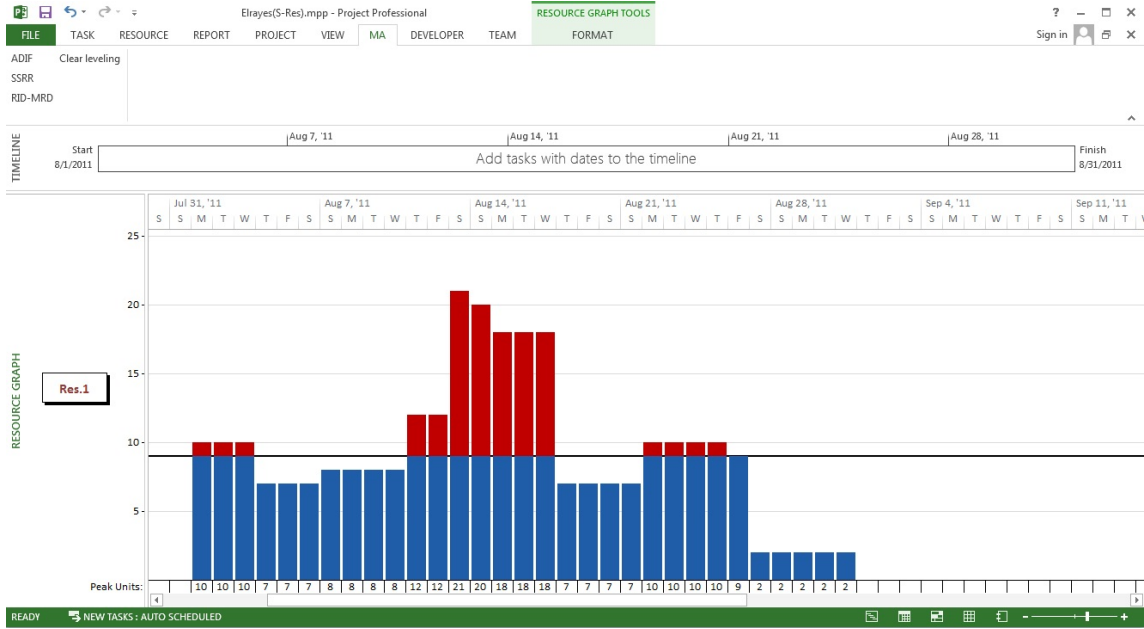
Örnek olarak El-Rayes ve Jun (2009)' a ait problemin dengelenmesi için azami kaynak kullanımı ve atıl kaynak günü (MRD+RID) ölçütü seçilmiştir. Ekranda sol üst köşede beliren TBMA'a eklentisine ait RID-MRD seçeneği tıklandıktan sonra ekranda bu ölçüte ait veri penceresi çıkmaktadır (Şekil 28). Bu veri penceresinde kaynakların ağırlıkları, benzer şekilde RID-MRD ölçütünde RID'in ve MRD'in ağırlığı ile kaynak dengelemesi için işlem süresi sınırı (saniye cinsinden) sorulmaktadır. Gerekli veriler girilip "OK" tuşuna basıldıktan sonra TBMA eklentisi kaynak dengelemesini belirtilen seçeneklere göre yapmakta ve elde ettiği çözüme ait iş programını ve kaynak profilini MS Project programı dahilinde vermektedir. Dengeleme sonucunda TBMA'nın elde ettiği kaynak profili Şekil 29'da verilmiştir. Aynı probleme ait kritik yol yöntemi ile elde edilmiş erken başlangıç kaynak profili Şekil 30'da verilmiştir. Şekil 29 ve Şekil 30 incelendiğinde TBMA eklentisinin bir numaralı kaynak için sadece azami kaynak miktarını 21'den 17'ye düşürmediği aynı zamanda, atıl kaynak günlerini önemli ölçüde azaltıp özellikle inşaat yapım projeleri için çok daha verimli kaynak planlamasına imkan tanıdığı gözlemlenmektedir. El-Rayes ve Jun (2009)'a ait problemde ortalama kaynak miktarından mutlak sapma (OKMSM) ve karelerin toplamı ölçütleri (KT) için TBMA eklentisinin elde ettiği dengelenmiş kaynak profilleri Şekil 31 ve Şekil 32'de verilmiştir. Her iki ölçütte azami kaynak miktarını 17'ye düşürmekte ancak atıl kaynak günlerini azaltmamaktadır. Bu örnekte de görüldüğü üzere El-Rayes ve Jun (2009)' un değindiği gibi RID-MRD ölçütü inşaat projelerinin verimli planlanması açısından mevcut ölçütlere göre çok daha başarılıdır. Bu örnek RID-MRD ölçütü ile kapsamlı kesin ve üst-sezgisel çalışmaların yapıldığı ve MS Project uygulamasının geliştirildiği ilk çalışma olan bu projenin önemini irdelemektedir.



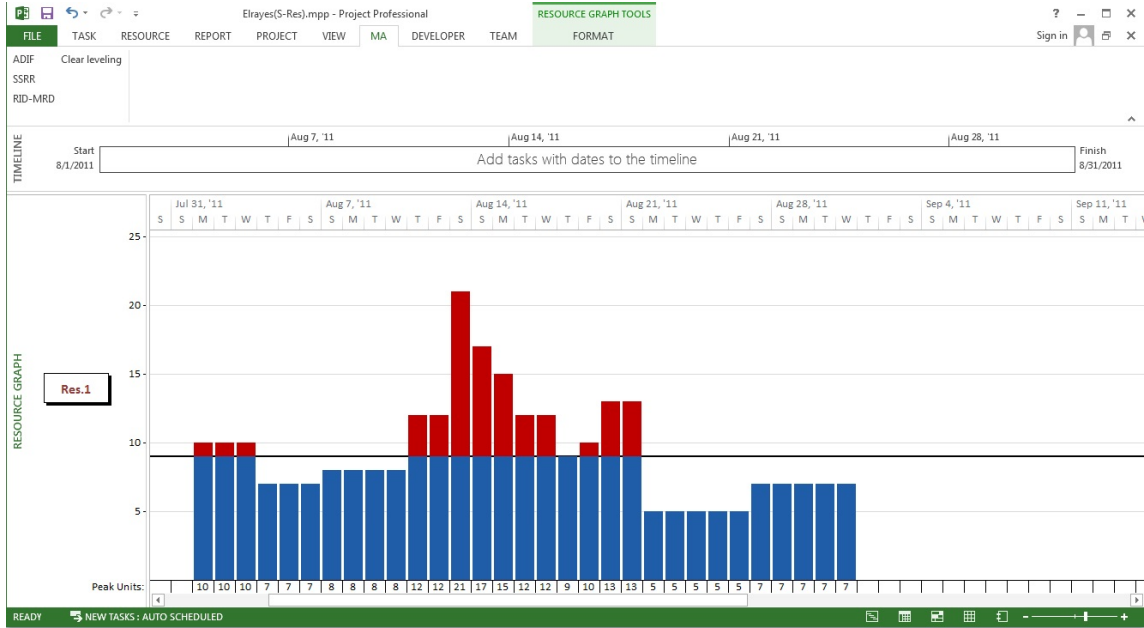
Şekil 28. MRD+RID ölçütüne ait veri penceresi



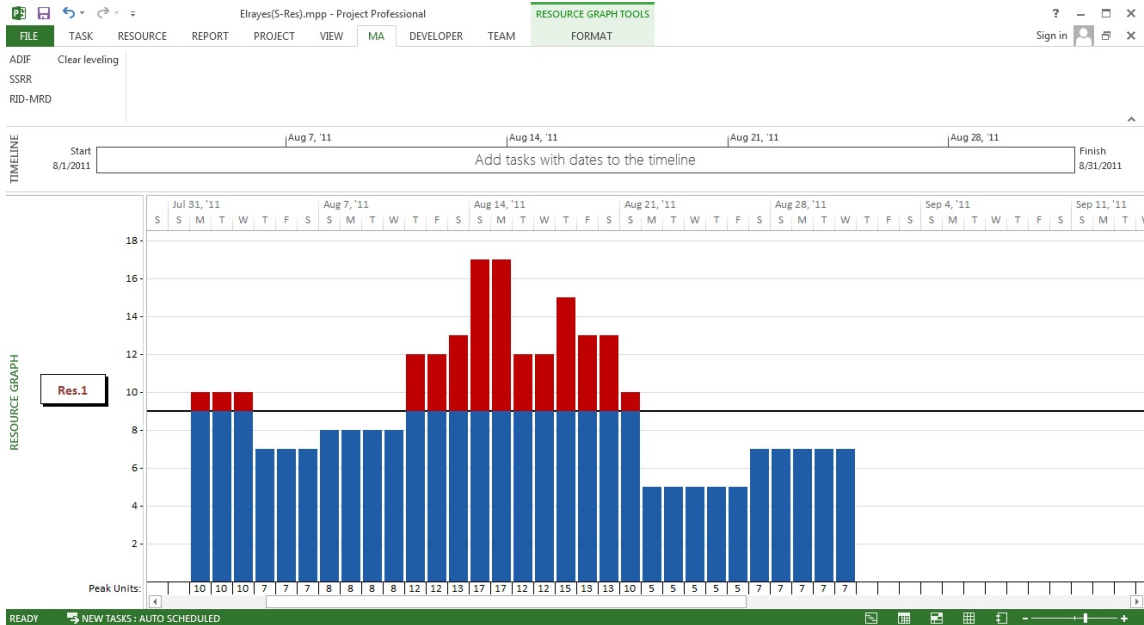
Şekil 29. El-Rayes ve Jun (2009)' a ait problemin MRD+RID ölçütüne göre TBMA eklentisi ile dengelenmiş kaynak profili



Şekil 30. El-Rayes ve Jun (2009)' a ait problemin erken başlangıç kaynak profili



Şekil 31. El-Rayes ve Jun (2009)'a ait problemin OKMSM ölçütüne göre TBMA eklentisi ile dengelenmiş kaynak profili



Şekil 32. El-Rayes ve Jun (2009)'a ait problemin KT ölçütüne göre TBMA eklentisi ile dengelenmiş kaynak profili



## 8.2. TEPE İnşaat Planlama Bölümü'nün görüşleri

Proje kapsamında gerçekleştirilen TBMA'a ait eklentinin sektör ihtiyaçlarını karşılayabilmesi açısından TEPE İnşaat Bütçe ve Planlama Bölümü Müdür Yardımcısı Tuğrul Tümay'ın görüşleri alınmıştır. Bu doğrultuda yapılan görüşmelerde Tuğrul Bey uygulama ile genelde olumlu görüş bildirmiştir. Tuğrul Bey kaynak dengelemesinin çok faydalı bir uygulama olduğunu ancak Türk müteahhitleri arasında yaygın olarak kullanılmadığını, bu tür uygulamaların Türk inşaat sektöründe kaynak dengelemesinin yaygınlaşmasına destek olabileceğini belirtmiştir. Tuğrul Bey ayrıca kaynak dengelemesi yapıldıktan sonra istenirse kritik yol yöntemine ait iş programını elde edebilmek için Clear Leveling seçeneğinin de eklentiye eklenmesini önermiştir. Bu görüş dahilinde ilk sürümde olmayan bu seçenek TBMA'a ait eklentinin son sürümünde dahil edilmiştir.

## 8.3. TBMA Microsoft Project Eklentisinin Gerçek bir İnşaat Projesi ile Denenmesi

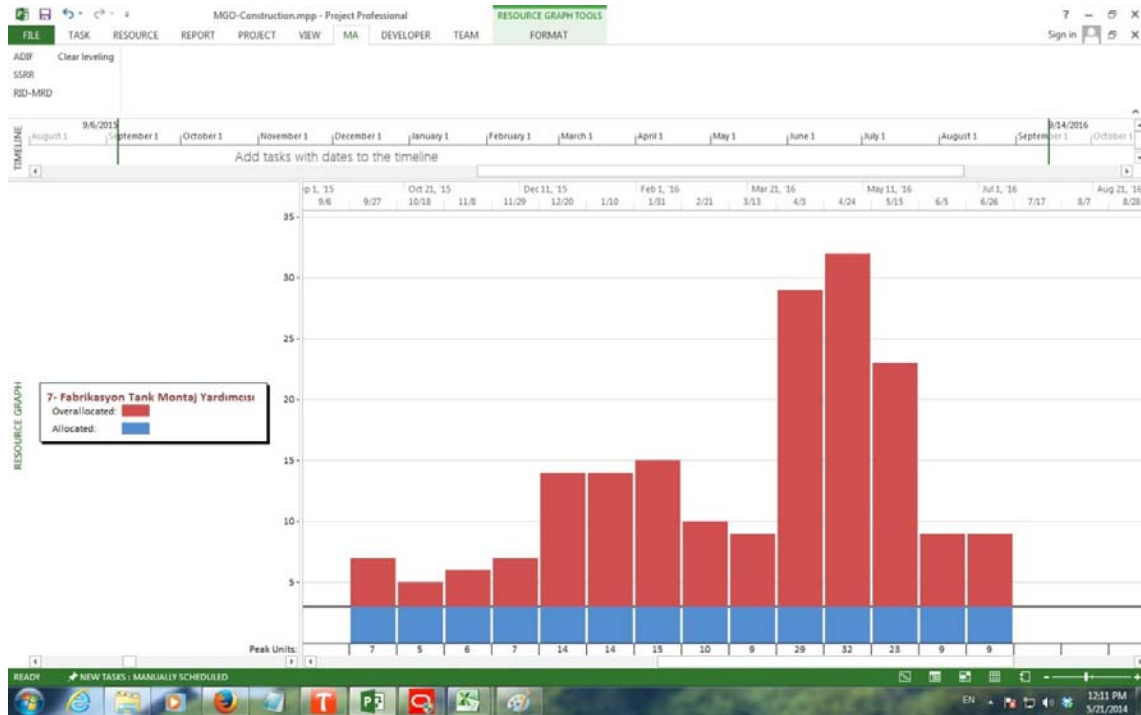
Son olarak proje kapsamında gerçekleştirilen TBMA eklentisi gerçek bir endüstriyel tesis projesi ile test edilmiştir. Proje toplam 522 yapım faaliyetini ve 18 işgücü kaynağını içermektedir. Projeye ait işgücü kaynakları Tablo 14'de verilmiştir. Proje farklı alanlarda aynı

Tablo 14. İnşaat Projesine ait Kaynaklar

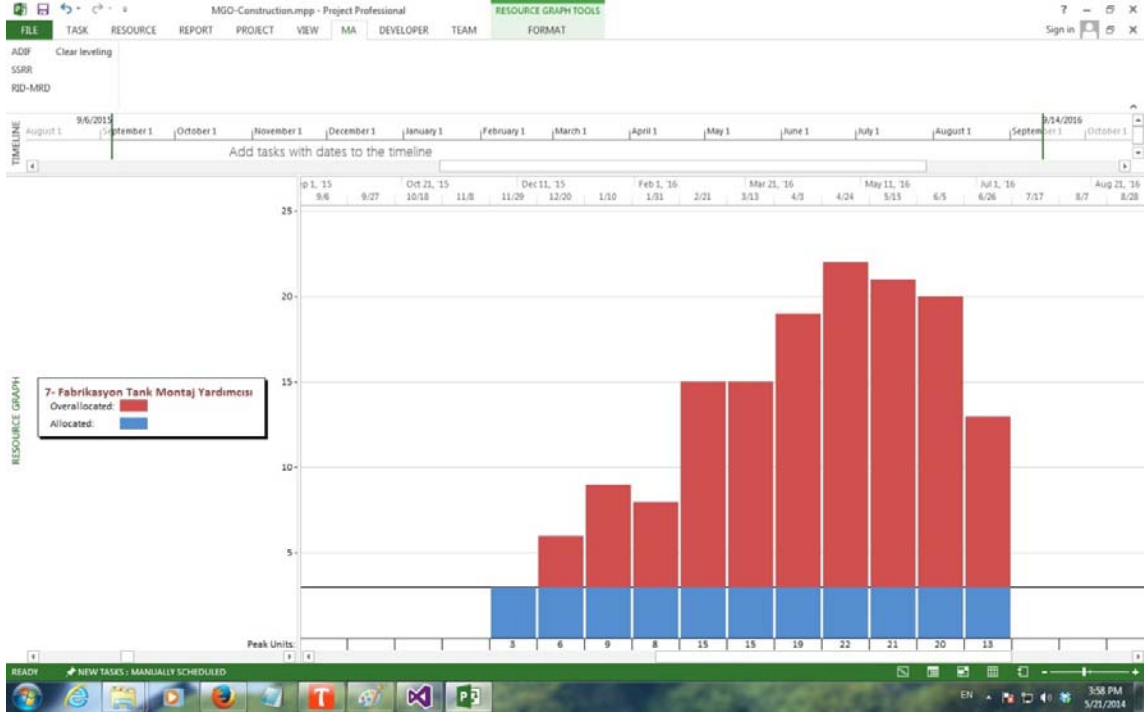
No	Kaynak
1	Marangoz Ustası
2	Marangoz Yardımcısı
3	Donatı Ustası
4	Betoncu
5	Elektrikçi
6	Elektrikçi Yardımcısı
7	Fabrikasyon Tank Montaj Ustası
8	Fabrikasyon Tank Montaj Yardımcısı
9	Fabrikasyon Tank Kaynak Ustası
10	Fabrikasyon Tank Kaynak Yardımcısı
11	Ekipman Montaj Ustası
12	Ekipman Montaj Yardımcısı
13	Endüstriyel Boru Montajcısı
14	Endüstriyel Boru Montajcısı Yardımcısı
15	Endüstriyel Boru Kaynakçısı
16	Endüstriyel Boru Kaynakçı Yardımcısı
17	Yapısal Çelik Montajcısı
18	Yapısal Çelik Kaynakçısı

anda çalışma imkanı tanıdığından genel olarak kritik yol dışındaki faaliyetlerin bollukları büyüktür. Gerek faaliyet sayısı, gerekse de bolluk miktarları değerlendirildiğinde proje oldukça zor bir kaynak dengelemesi problemini içermektedir.

Proje kapsamında geliştirilen TBMA eklentisi endüstriyel tesis projesine ait yapım faaliyetlerinin kaynak dengelenmesinde test edilmiştir. Projeye ait iş programının kaynak dengelenmesi için MRD+RID ölçütü seçilmiş, işlem süresi olarak ise 2 dakika kullanılmıştır. Dengeleme esnasında her kaynağa eşit ağırlık verilmiştir. TBMA eklentisi sorunsuz çalışmış ve projeye ait bütün kaynakların dengelemesini yapmıştır. Örnek olması açısından fabrikasyon tank montaj yardımcısı, fabrikasyon tank kaynak ustası, ve donatı ustasına ait kritik yol yönetimi ile elde edilmiş erken başlangıç ve TBMA eklentisi ile elde edilmiş dengelenmiş kaynak profilleri Şekil 33 – Şekil 38’de verilmiştir. TBMA eklentisi kaynaklar için hem azami kaynak miktarını hem de atıl kaynak günlerini önemli ölçüde azaltmıştır. Endüstriyel tesis projesi ile yapılan test, proje kapsamında geliştirilen üst sezgisel algoritmanın inşaat projelerine sağlayabileceği katkıyı teyit etmektedir.



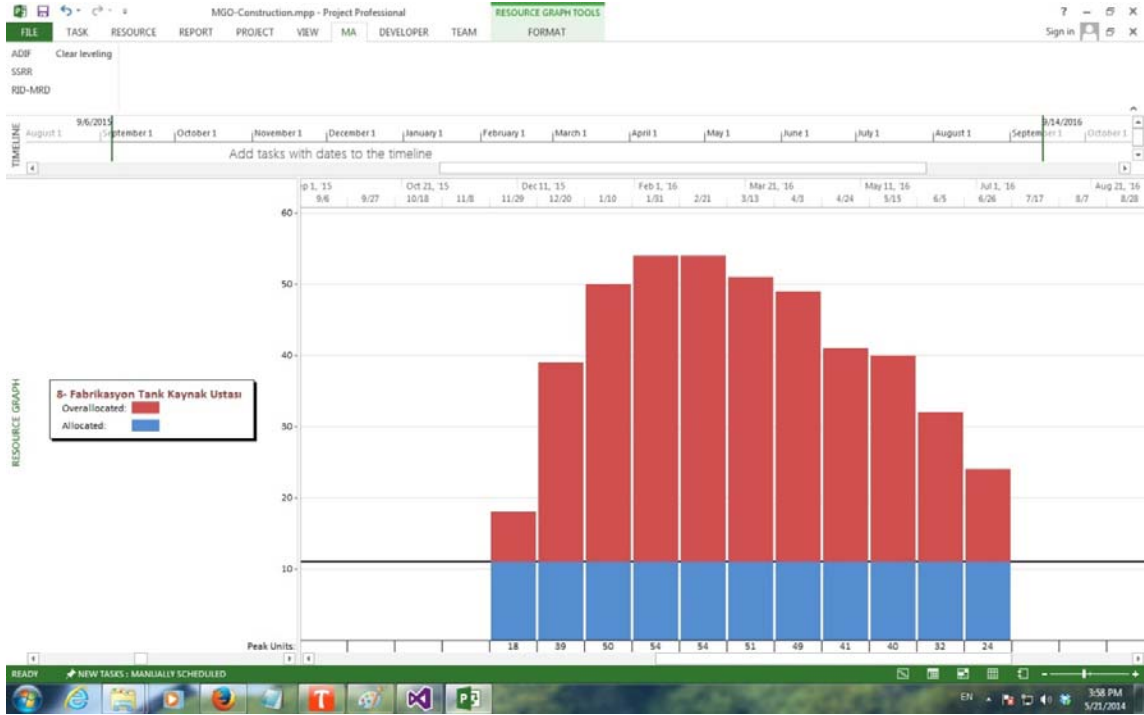
Şekil 33. Fabrikasyon tank montaj yardımcısı için erken başlangıç kaynak profili



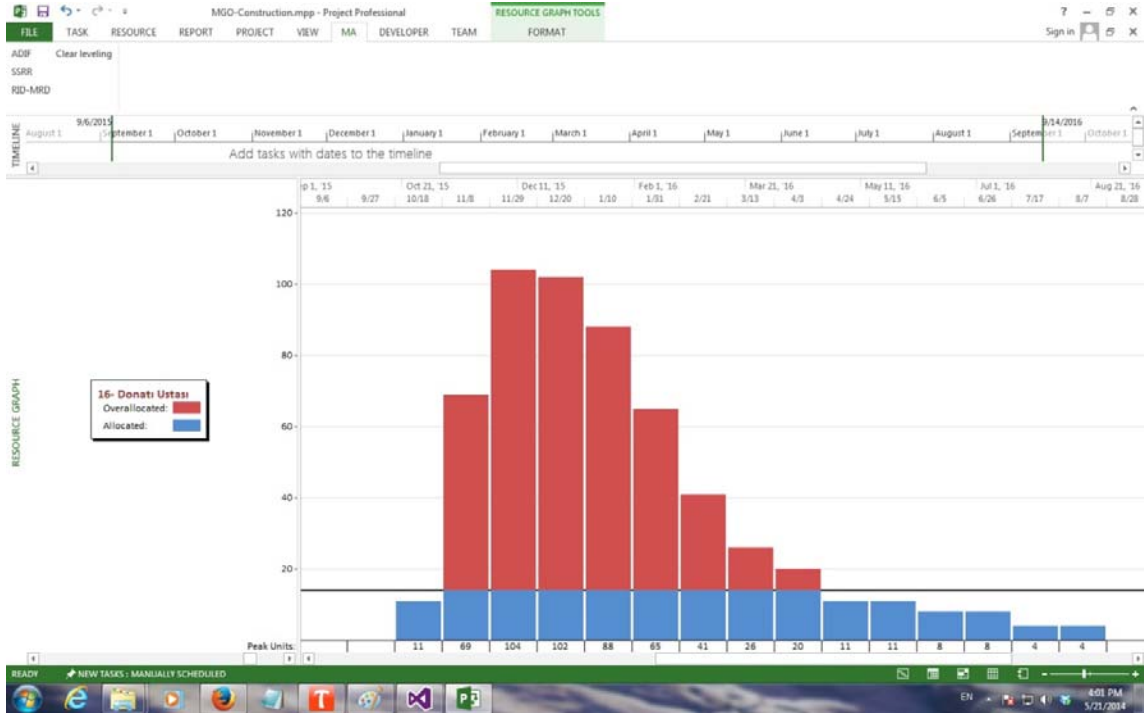
Şekil 34. Fabrikasyon tank montaj yardımcısı için TBMA eklentisi ile dengelenmiş kaynak profili



Şekil 35. Fabrikasyon tank kaynak ustası için erken başlangıç kaynak profili



Şekil 36. Fabrikasyon tank kaynak ustası için TBMA eklentisi ile dengelenmiş kaynak profili



Şekil 37. Donatı ustası için erken başlangıç kaynak profili





Şekil 38. Donatı ustası için TBMA eklentisi ile dengelenmiş kaynak profili

## 9. SONUÇLAR VE ÖNERİLER

Proje kapsamında yüksek performanslı kesin ve üst-sezgisel yöntemler geliştirilerek küçük, orta, ve büyük ölçekli şebekeler için kaynak dengeleme probleminin çözümünde ilerleme sağlanması hedeflenmiştir. Bu doğrultuda proje önerisinde ilk hedef olarak KDP'nin kesin çözümü için bir dal ve sınır algoritması geliştirilmesi ve geliştirilecek kesin algoritma ile Neumann ve Zimmermann (2000) 'ın önerdiği dal-sınır algoritmasının performansının iyileştirilerek, en çok 20 faaliyetli ve 4 kaynaklı küçük ölçekli KDP' in çözülmesi belirlenmiştir. Proje önerisinde ikinci hedef olarak ise KDP için yeni bir melez üst sezgisel algoritma geliştirilerek, kaynak dengelemesi için kullanılan aralarında Burgess sezgisel metodu ve yalın gen algoritmalarının bulunduğu güncel yöntemlerin iyileştirilmesi olarak belirlenmiştir. Bu doğrultuda sezgisel ve üst sezgisel algoritmaların performansının değerlendirilmesi için gerekli olan çözümlü örnek problemlerin oluşturulması da proje önerisinde belirtilen hedefler arasındadır. Proje önerisinde belirtilen bir diğer hedef ise proje kapsamında geliştirilecek kesin ve üst sezgisel yöntemlerinin karelerin toplamı ve ortalama kaynak miktarından mutlak sapma ölçüleri ile birlikte inşaat projeleri için son zamanlarda önerilen azami kaynak kullanımı ve atıl kaynak günü ölçütü için de başarılı sonuçlar elde etmesidir. Son olarak



proje önerisinde geliştirilecek üst-sezgisel kaynak dengeleme algoritmasının Microsoft Project paket programı ile entegre edilerek projenin yaygın etkisinin artırılması hedeflenmiştir.

Kesin yöntemlerle ilgili olarak yapılan kapsamlı testler, proje kapsamında geliştirilen dal ve sınır algoritmasının literatürdeki mevcut Neumann ve Zimmermann (2000) 'ın ve Gather vd. (2010)' in geliştirdiği dal ve sınır algoritmalarının performansının önemli ölçüde önüne geçtiğini teyit etmektedir. Bu doğrultuda proje hedeflerinden ilkinde ulaşılmıştır. Dal ve sınır algoritması geliştirilirken Neumann ve Zimmermann (2000) 'ın önerdiği alt sınır hesabına ilaveten KDP için yeni bir alt sınır hesap yöntemi bulunmuş, bu sayede KDP'nin dal ve sınır algoritması ile çözümünde önemli bir gelişme kaydedilmiştir. Ayrıca proje kapsamında geliştirilen karma-tamsayılı doğrusal modelleri ve dal ve sınır algoritması kullanılarak karelerin toplamı, ortalama kaynak miktarından mutlak sapma, ve azami kaynak kullanımı ve atıl kaynak günü ölçütleri için 30 faaliyet ve dört kaynağa kadar olan problemler için optimal sonuçlar elde edilerek çözümlü örnek problemlerin oluşturulmuştur. Bu doğrultuda literatürde ilk kez azami kaynak kullanımı ve atıl kaynak günü ölçütü için problemler optimal olarak çözülmüştür.

Proje kapsamında KDP'nin çözümünde güncel üst-sezgisel yöntemlerin iyileştirilmesi için tavlama benzetimli bir memetik algoritma önerilmiştir. Önerilen üst sezgisel yöntem kaynak kullanımının verimsiz olduğu kısımların azaltılması için memetik algoritmaların yerel taramalar için bireysel öğrenme özellikleri kullanmakta, çözüm uzayı tarama işlemini kontrol edilmesi ve özellikle tarama işlemin sonlarına doğru iyi çözümlerin muhafaza edilip, çözümlerin optimale yakınsaması amacıyla tavlama benzetimi yöntemini kullanmaktadır. Proje kapsamında geliştirilen tavlama benzetimli memetik algoritmanın (TBMA) performansı aralarında iyileştirilmiş Burgess sezgisel metodu, yalın gen algoritmaları, MS Project 2010 ve Primavera 6.7 yazılımlarının kullandığı toplam dokuz sezgisel ve Ponz-Tienda v. d. (2013) tarafından geliştirilen adaptif genetik algoritmasının da olduğu 12 farklı metodun performansı ile 300 faaliyet ve dört kaynağa kadar olan 1920 problem kullanılarak yapılan kapsamlı testlerle kıyaslanmıştır. Performans test sonuçları TBMA'nın KDP'nin çözümünde performansının mevcut sezgisel ve üst-sezgisel yöntemlere çok daha iyi olduğu teyit etmektedir. Proje kapsamında geliştirilen kesin yöntemlerle optimal çözümleri elde edilen 30 faaliyet ve dört kaynağa kadar olan problemlerle yapılan testler, TBMA'nın 30 faaliyete kadar küçük ölçekli problemler için optimal veya optimale yakın çözümleri birkaç saniye içerisinde elde ettiğini göstermektedir. Bu doğrultuda proje kapsamında KDP'nin çözümünde güncel yöntemlerin iyileştirilmesi yolunda önemli bir adım atılmıştır. Proje kapsamında geliştirilen



kesin yöntemlerle çözülen problemlerin optimal sonuçları ve TBMA ile yapılan testlerin sonuçları proje önerisinde hedeflendiği gibi proje için oluşturulan bir web sitesinde araştırmacıların erişimine sunulmuştur.

Proje kapsamında geliştirilen tavlama benzetimli memetik üst-sezgisel algoritmanın yaygın etkisinin artırılması için Microsoft Project 2013 yazılımı ile entegrasyonu bir eklenti geliştirilerek yapılmıştır. Bu doğrultuda gerçekleştirilen TBMA'a ait eklenti karelerin toplamı, ortalama kaynak miktarından mutlak sapma miktarı, ve azami kaynak kullanımı ve atıl kaynak günü ölçütleri için kaynak dengelemesi imkanı sunmaktadır. 522 yapım faaliyetini ve 18 işgücü kaynağını içeren gerçek bir endüstriyel tesis projesi ile yapılan teste TBMA'a ait eklentinin Microsoft Project 2013 yazılımı ile birlikte sorunsuz çalışmıştır. TBMA eklentisi iki dakikalık süre içerisinde yaptığı kaynak dengelemesi ile endüstriyel tesis projesi için kritik yol yöntemi ile elde edilmiş kaynak profilindeki kaynak miktarını ve de atıl kaynak günlerini önemli ölçüde azalmıştır. TBMA eklentisinin geliştirilmesiyle proje önerisinde belirtilen son hedefe de ulaşılmıştır.

Projede dal ve sınır yöntemli algoritmaları üzerine çalışan yüksek lisans öğrencisi Hüseyin Yeniocak bu konudaki yüksek lisans tezini tamamlayıp yayımlamıştır. Yine projede üst sezgisel yöntem konusunda çalışan doktora öğrencisi Mahdi Abbasi Iranagh'ın Aralık 2014 itibari ile, yüksek lisans öğrencisi Emad Rezvan Khah'ın ise Eylül 2014 itibari ile tez çalışmalarını tamamlamaları hedeflenmektedir. Proje çalışmalarından bir adet yurt içi iki adet de yurt dışı bildiri yayımlanmıştır. Yayımlanan bildirimler Ek-1, Ek-2 ve Ek-3 olarak rapor sonuna eklenmiştir. Ayrıca proje kapsamında geliştirilen TBMA algoritması ve doğrusal tamsayıli eniyileme model çalışmalarından bir makale hazırlanmış ve makale Science Citation Index kapsamında taranan ve yapım yönetimi dalının saygın dergileri arasında olan Automation in Construction dergisine sunulmuştur. Gelen hakem raporları dahilinde makalenin 1. revizyonu yapılmıştır ve bu revizyon sonucu bir hakem makaleyi kabul etmiş, bir hakem ufak revizyon önermiş, bir hakem ise orta kapsamlı revizyon önermiştir. Yapılacak ikinci revizyon sonucunda makalenin kabul edilmesi beklenmektedir. Makale ve makale ile ilgili hakem raporları Ek-4'de verilmiştir. Yine proje kapsamında yapılan dal ve sınır çalışmaları ile ilgili olarak bir makale hazırlanmaktadır. Bu makalenin yöneylem araştırması konusunda Science Citation Index kapsamında taranan bir dergiye gönderilmesi planlanmaktadır. Ayrıca üst-sezgisel algoritmalar konusunda yapılan çalışmaların son durumunun Kasım 2014'de yapılacak 3. Proje ve Yapım Yönetimi Kongresi'nde sunulması ve bu konuda ikinci bir yurtiçi bildiri yayınlanması planlanmaktadır. Bu konuda sunulan bildiri



özeti kabul edilmiştir. Sonuç itibari ile proje kapsamında yapılan çalışmalar bildiriler, makaleler ve proje için oluşturulan bir web sitesi yoluyla akademik dünyaya sunulmuştur.

Proje kapsamında geliştirilen kesin ve üst sezgisel yöntemler proje hedeflerine ulaşmış olmakla birlikte performanslarının gelecekte çeşitli yöntemler yoluyla daha da geliştirilme potansiyeli mevcuttur. Bu doğrultuda özellikle son yıllarda bilgisayarlarda 6 çekirdeğe kadar olan çok çekirdekli işlemciler yaygın olarak kullanılmaktadır. Gelecekteki bilgisayarlarda çekirdek sayısının daha da artacağı öngörülmektedir. Çok çekirdekli bilgisayarlar paralel işlem özellikleri ile KDP için geliştirilen gerek kesin gerekse üst sezgisel yöntemlerin performanslarının iyileştirilmesinde önemli potansiyel sunmaktadır. Yine proje kapsamında geliştirilen üst sezgisel algoritmanın bütün potansiyel kaynak profillerini taraması yerine, çizge kuramı ile belirlenen ve içlerinde en az bir optimal profil barındıran kaynak profil setinin bir alt seti olan özel bir kaynak profili setinin içeren profilleri taramasının üst sezgisel algoritmaların performansını geliştireceği düşünülmektedir. Bu doğrultuda proje kapsamında geliştirilen TBMA algoritmasının Mahdi Abbasi Iranagh'ın doktora tezi dahilinde iyileştirilmesinin imkanları araştırılmaktadır.

## **Kaynaklar**

Ballestin, F., Schwindt, C., Zimmermann, J., 2007. "Resource Leveling in Make-to-Order Production: Modelling and Heuristic Solution Method", International Journal of Operational Research, 4(1), 50-62.

Bandelloni, M., Tucci, M., Rinaldi, R., 1994. "Optimal resource leveling using non-serial dynamic programming", European Journal of Operational Research 78, 162-177.

Bettemir Ö. H. 2009. "Optimization of time-cost-resource trade-off problems in project scheduling using meta-heuristic algorithms." PhD Thesis, Middle East Technical University, Ankara, Turkey.

Burgess, A. R., Killebrew, J. B., 1962. "Variation in activity level on a cyclic arrow diagram." J. Industrial Engrg., 13(2), 76-83.

Blum C, Roli A., 2008. "Hybrid Metaheuristics: An Introduction, in Hybrid Metaheuristics An Emerging Approach to Optimization Studies in Computational Intelligence" vol. 114, Blum, C.; Blesa Aguilera, M.J.; Roli, A.; Sampels, M. (Eds.) Berlin Heidelberg: Springer-Verlag

Chan, W., Chua, K. H., Kannan, G., 1996. "Construction resource Scheduling with Genetic Algorithms", *Journal of Construction Engineering and Management*, 122( 2), 125-132.

Chen D., Lee C. Y., Park C. H. 2005. "Hybrid genetic algorithm and simulated annealing (HGASA) in Global Function Optimization". *Proceedings of the 17th IEEE International Conference in Tools with Artificial Intelligence (ICTAI'05)*, 1082- 3409/05.

Chica, M., Cordon, O., Damas, S., & Bautista, J., 2012. "Multiobjective Memetic Algorithms for Time and Space Assembly Line Balancing. *Engineering Applications of Artificial Intelligence*" 25, 254–273.

Christodoulou, S. E., Ellinas, G., Michaelidou-Kamenou, A., 2010. "Minimum Moment Method for Resource Leveling Using Entropy Maximization", *Journal of Construction Engineering and Management*, 136(5), 518-527.

Debels D., ve Vanhoucke M., 2007, "A decomposition-based genetic algorithm for the resource-constrained project-scheduling problem", *Operations research* 55(3) : 457 – 469.

Doulabi, S. H. H., Seifi, A., Shariat, S. Y., 2011. "Efficient Hybrid Genetic Algorithm for Resource Leveling via Activity Splitting", *Journal of Construction Engineering and Management*, 137(2), 137–146.

Easa, S.M., 1989. "Resource leveling in construction by optimization", *Journal of Construction Engineering and Management*, 115, 302-316.

El-Rayes K., Jun, D. H., 2009. "Optimizing resource leveling in construction projects", *Journal of Construction Engineering and Management*, 135(11), 1172–1180.

Franck, B., Neumann, K., & Schwindt, C. 2001. "Truncated branchand- bound, schedule construction, and schedule improvement procedures for resource-constrained project scheduling". *OR Spektrum*, 23, 297–324.

Gather T., Zimmermann J., and Bartels J. H., 2010, "Exact methods for the resource leveling problem", Springer Science+Bussiness Media, LLC.



Geng J. Q., Weng L. P., Liu S. H., 2010. "An improved ant colony optimization algorithm for nonlinear resource leveling problems", *Computers and Mathematics with Applications*, doi:10.1016/j.camwa.2010.09.058,1-6.

Guo, Y., Li, N., Ye, T., 2009. "Multiple Resource Leveling in Multiple Projects Scheduling Problem Using Particle Swarm Optimization", *Fifth International Conference on Natural Computation*, 260-264.

Han M., Li P. and Sun J., 2006. "The algorithm for berth scheduling problem by the hybrid optimization strategy GASA", *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference*, 1-4.

Harris, R. B., 1990. "Packing Method for Resource Leveling (PACK)", *Journal of Construction Engineering and Management*, 125(3), 167-175.

Hegazy, T., 1999. "Optimization of resource allocation and levelling using genetic algorithms." *Journal of Construction Engineering and Management*, 125(3), 167–175.

Hegazy, T., Erşahin, T., 2001. "Simplified Spreadsheet Solutions: II: Overall Schedule Optimization". *Journal of Construction Engineering and Management*, Vol. 127 (6), 112-117 .

Hiyassat, M. A. S., 2001. "Applying Modified Minimum Moment Method to Multiple Resource Leveling", *Journal of Construction Engineering and Management*, 127(3), 192-198.

Hwang S. F., He R.S. 2006. "Improving Real-Parameter Genetic Algorithm with Simulated Annealing for Engineering Problems". *Advances in Engineering Software* 37, 406 – 418.

Jun, D. H., El-Rayes, K., 2011. "Multiobjective Optimization of Resource Leveling and Allocation during Construction Scheduling", *Journal of Construction Engineering and Management*, 137(12), 92-99 .

Karshenas S., Haber D., 1990. "Economic Optimization of Construction Project Scheduling", *Construction Management and Economics*", Vol. 8(2), 135-146.

Kartam, N., Tongthong T., 1998. "An artificial neural network for resource leveling problems", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 12, 273–287.

Kastor A., Sirakoulis K., 2009. "The effectiveness of resource levelling tools for resource constraint project scheduling problem", *International Journal of Project Management*, 27, 493–500.

Kolisch R., 1999. "Resource allocation capabilities of commercial project management software packages", *Interfaces* 29(4), 19 – 31.

Kolisch, R., A. Sprecher, A. Drexler., 1995. "Characterization and generation of a general class of resource constrained project scheduling problems." *Management Science*, 41, 1693–1703.

Kolish R. and Sprecher A., 1997. "PSPLIB – A project scheduling problem library", *European Journal of Operational Research*, 96, pp. 205 – 216.

Leu, S.S., Yang, C.H., 1999. "GA-Based Multicriteria Optimal Model for Construction Scheduling", *Journal of Construction Engineering and Management*, 125(6), 420-427.

Leu, S.S., Yang, C.H., Huang J. C., 2000. "Resource leveling in construction by genetic algorithm-based optimization and its decision support system application", *Automation in Construction*, 10, 27–41.

Liberatore M. J., Johnson B. P. and Smith C. A., 2001. "Project management in construction: Software use and research directions", *Journal of Construction Engineering and Management* 127(2), 101 – 107.

Martinez, J., and Ioannou, P., 1993. "Resource leveling based on the modified minimum moment heuristic." *Proc., 5th Int. Conf., Computing in Civ. and Build. Engrg., ASCE, Reston, Va.*, pp. 287–294.

Mattila, K. G., Abraham, D. M., 1998. "Resource leveling of linear schedules using integer linear programming." *Journal of Construction Engineering and Management*, 124(3), 232-244.

Mutlu, C., 2010. A Branch and Bound Algorithm for Resource Leveling Problem, Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi, Ankara, Türkiye.



Mubarak, S. A., 2004. *Construction Project Scheduling and Control*, Pearson Prentice Hall, Upper Saddle River, New Jersey.

Neri, F., & Cotta, C. (2012). "Memetic Algorithms and Memetic Computing Optimization." *Swarm and Evolutionary Computation*, 2, 1–14, (2012).

Neumann K., Zimmermann J., 1999. "Resource Leveling for Projects with Schedule-Dependent Time Windows", *European Journal of Operational Research*, 117, 591-605.

Neumann K., Zimmermann J., 2000. "Procedures for resource leveling and net present value problems in project scheduling with general temporal and resource constraints", *European Journal of Operational Research*, 127, 425-443.

Neumann K., Schwindt C., Zimmermann J., 2003. *Project Scheduling with Time Windows and Scarce Resources*, Springer, Berlin.

Nguyen, Q. H., Ong, Y. S., & Lim M. H. 2009. "A probabilistic memetic framework." *IEEE Transactions on Evolutionary Computation*, 13(3), 604–623.

Newitt, J. S., 2005. *Construction Scheduling: Principles and Practices*, Pearson Prentice Hall, Upper Saddle River, New Jersey.

Pishvaei, M. S., R. Z. Farahani, et al. 2010. "A memetic algorithm for bi-objective integrated forward/reverse logistics network design." *Computers & Operations Research*. 37(6): 1100-1112.

Oral M., Oral E. L., Bozkurt S., Erdiř E., 2003. "Yapım projelerinde genetik algoritma kullanarak kaynak seviyeleme", *Çukurova Üniversitesi Mimarlık Fakültesi Dergisi*, 18(2), 185-194.

Pang, N., Shi, Y., You, Y., 2008. "Resource Leveling Optimization of Network Schedule Based on Particle Swarm Optimization with Constriction Factor", *International Conference on Advanced Computer Theory and Engineering*, 652-656.





Qi J. X., Wang Q., Guo X. Z., 2007. "Improved particle swarm optimization for resource leveling problem" Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 896-901.

Rieck J., Zimmermann J., Gather T., 2012. "Mixed-integer linear programming for resource leveling problems." European Journal of Operational Research 221, 27-37.

Roca, J., Pugnaghi, E., Libert, G., 2008. "Solving an Extended Resource Leveling Problem with Multiobjective Evolutionary Algorithms", Proceedings of World Academy of Science, Engineering and Technology, 36.

Savin D., Alkass S., Fazio P., 1997. "A procedure for calculating the weight-matrix of a neural network for resource leveling" Advances in Engineering Software 28 28, 271-283.

Senouci, A. B., Eldin, N. N., 2004. "Use of Genetic Algorithms in Resource Scheduling of Construction Projects", Journal of Construction Engineering and Management, 130(6), 869-877.

Son, J., Skibniewski, M. J., 1999. "Multiheuristic approach for resource leveling problem in construction engineering: Hybrid approach." Journal of Construction Engineering and Management, 125, 23–31.

Son, J., Mattila, K. G., 2004. "Binary resource leveling model: Activity splitting allowed." Journal of Construction Engineering and Management, 130(6), pp. 887-894.

Stevens J. D., 1990, Techniques for Construction Network Scheduling, Mc Graw-Hill, New York.

Ponz-Tienda J.L, Yepes V., Pellincer E., Moreno-Flores J., 2013, "The Resource Leveling Problem with multiple resource using adaptive genetic algorithm," Automation in Construction, 29, 161–172.

Wang L., Zheng D., 2001. "An effective hybrid optimization strategy for job-shop scheduling problems". Computers & Operations Research 28, 585 – 596.



Woodworth, M. W., Willie, C. J., 1975. "A Heuristic Algorithm for Resource Leveling in Multi-Project, Multi Resource Scheduling", Decision Sciences, 6(3), 525-540.

Xiong, Y., Kuang, Y. P., 2006. "Ant Colony Optimization Algorithm for Resource Leveling Problem of Construction Project", The CRIOCM 2006 International Symposium on "Advancement of Construction Management and Real Estate".

Younis, M.A., Saad, B., 1996, "Optimal resource leveling of multi-resource projects", Computers and Industrial Engineering 31, 1-4.

Zamani M. R., 2001. "A high-performance exact method for the resource-constrained project scheduling problem". Computers & Operations Research 28, 1387 – 1401.

Zheng, D. X. M., NG, s. T., Kumaraswamy, M. M., 2003. "Applying a genetic algorithm-based multiobjective approach for time-cost optimization", ASCE Construction Research Congress, pp. 1-8.

#### **Akademik Çıktılar ve Yayınlar**

##### **SCI-Expanded kapsamında taranan dergilerde değerlendirme aşamasındaki makale**

Sönmez R., Atan T., Iranagh M.A. "A Memetic Algorithm for the Resource Leveling Problem", Automation in Construction dergisine ilk revizyona gelen yorumlar dahilinde ikinci revizyonu gönderilecek (EK4).

##### **Uluslararası bilimsel toplantılarda sunulan ve bildiri kitabında basılan bildiriler**

Iranagh M.A., Atan T., Sonmez R, 2013. "A mixed-integer linear model for optimization of resource idle days in project scheduling", Proceedings Creative Construction Conference, Budapest, Hungary, 368–381 (EK1).

Iranagh M.A., Sonmez R., 2012. "A genetic algorithm for resource leveling of construction projects", Proceedings 28th Annual ARCOM Conference, Edinburgh, UK, 1047–1055 (EK2).



### **Ulusal bilimsel toplantılarda sunulan ve bildiri kitaplarında basılan bildiri**

Iranagh M.A., Atan T., Sonmez R, 2012. “MS Project Paket Programlarının Kaynak Dengeleme Problemi Çözümündeki Performansı”, 2. Proje ve Yapım Yönetimi Kongresi, İzmir Yüksek Teknoloji Enstitüsü, Urla-İzmir (EK3).

### **Yüksek Lisans Tezleri**

Yeniocak H., 2013. “Kaynak Dengeleme Probleminin Kesin Çözümü için bir Dal ve Sınır Algoritması Geliştirilmesi”, Yüksek Lisans Tezi, Orta Doğu Teknik Üniversitesi İnşaat Mühendisliği Bölümü, Ankara.

### **Web Sitesi**

Proje kapsamında yapılan çalışmaların sonuçlarının araştırmacıların erişimine sunulduğu <http://www2.ce.metu.edu.tr/~sonmez/research/111M140> adresli web sitesi.

### **Hazırlık Aşamasındaki Akademik Çıktılar ve Yayınlar**

Dal ve sınır algoritması çalışmaları ile ilgili SCI-Expanded kapsamında taranan ve yöneylem araştırması konusundaki bir dergiye sunulacak bir makale.

Üst-sezgisel algoritmalar konusunda yapılan çalışmaların son durumu içeren Kasım 2014’de yapılacak 3. Proje ve Yapım Yönetimi Kongresi’ne sunulacak bildiri (Özeti kabul edildi).

Mahdi Abbasi Iranagh’ın doktora tezi (Aralık 2014 itibari).

Emad Rezvan Khah’ın yüksek lisans tezi (Eylül 2014 itibari).

**TÜBİTAK**  
**PROJE ÖZET BİLGİ FORMU**

Proje Yürütücüsü:	Doç. Dr. RIFAT SÖNMEZ
Proje No:	111M140
Proje Başlığı:	İnşaat Yapım Projelerinde Kaynak Dengeleme Problemi İçin Yüksek Performanslı Kesin Ve Üst-Sezgisel Yöntemler Geliştirilmesi
Proje Türü:	Araştırma
Proje Süresi:	30
Araştırmacılar:	İSMAİL ALPHAN NURTUĞ, SABRİ TANKUT ATAN
Danışmanlar:	
Projenin Yürütüldüğü Kuruluş ve Adresi:	ORTA DOĞU TEKNİK Ü. MÜHENDİSLİK F. İNŞAAT MÜHENDİSLİĞİ B.
Projenin Başlangıç ve Bitiş Tarihleri:	01/10/2011 - 01/04/2014
Onaylanan Bütçe:	150975.0
Harcanan Bütçe:	132341.95
Öz:	<p>Bu projede kaynak dengeleme probleminin (KDP) çözümünde ilerleme sağlanabilmesi için yüksek performanslı kesin ve üst-sezgisel yöntemler geliştirilmiştir. Yapılan kapsamlı testler proje kapsamında geliştirilen dal ve sınır algoritmasının literatürdeki mevcut dal ve sınır algoritmalarının performansının önemli ölçüde önüne geçtiğini teyit etmektedir. Dal ve sınır algoritması geliştirilirken KDP için yeni bir alt sınır hesap yöntemi bulunmuş, bu sayede KDP'nin dal ve sınır algoritması ile çözümünde önemli bir gelişme kaydedilmiştir. Geliştirilen dal ve sınır algoritması ile inşaat projeleri için pratik açıdan önemli olan azami kaynak kullanımı ve atıl kaynak günü ölçütü için 30 faaliyet ve dört kaynağa kadar olan problemler literatürde ilk kez optimal olarak çözülmüştür.</p> <p>Proje kapsamında KDP'nin çözümünde güncel üst-sezgisel yöntemlerin iyileştirilmesi için tavlama benzetimli bir memetik algoritma (TBMA) geliştirilmiştir. TBMA kaynak kullanımının verimsiz olduğu kısımların azaltılması için memetik algoritmaların yerel taramalar için bireysel öğrenme özellikleri kullanmakta, çözüm uzayı tarama işlemini kontrol edilmesi ve özellikle tarama işleminin sonlarına doğru iyi çözümlerin muhafaza edilip, çözümlerin optimala yakınsaması amacıyla tavlama benzetimi yöntemini kullanmaktadır. TBMA'nın performansı mevcut 12 farklı sezgisel ve üst-sezgisel metodun performansı ile 300 faaliyet ve dört kaynağa kadar olan 1920 problem kullanılarak üç farklı ölçüt için kıyaslanmıştır. Performans test sonuçları TBMA'nın KDP'nin çözümünde performansının mevcut sezgisel ve üst-sezgisel yöntemlere çok daha iyi olduğu teyit etmektedir. Proje kapsamında geliştirilen tavlama benzetimli memetik üst-sezgisel algoritmanın yaygın etkisinin artırılması için Microsoft Project 2013 yazılımına bir eklenti olarak ile entegre edilmiş ve geliştirilen eklenti 522 yapım faaliyeti ve 18 işgücü kaynağını içeren gerçek bir endüstriyel tesis inşaat projesi ile test edilmiştir. Proje kapsamında yapılan çalışmaların sonuçları bildiriler, makaleler ve proje için oluşturulan bir web sitesi yoluyla akademik dünyaya sunulmuştur.</p>
Anahtar Kelimeler:	Proje yönetimi, kaynak dengelemesi, kaynak yönetimi, optimizasyon, üst sezgisel algoritmalar.
Fikri Ürün Bildirim Formu Sunuldu Mu?:	Hayır