

**T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**SINIRLI KAYNAKLI İŞ PROGRAMI PROBLEMLERİNİN OPTİMUM VEYA
YAKIN OPTİMUM ÇÖZÜLMESİ**

YÜKSEK LİSANS TEZİ

Derya ÇAKMAK

İnşaat Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Önder Halis BETTEMİR

Ağustos 2021

T.C
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

**SINIRLI KAYNAKLI İŞ PROGRAMI PROBLEMLERİNİN OPTİMUM VEYA
YAKIN OPTİMUM ÇÖZÜLMESİ**

YÜKSEK LİSANS TEZİ

Derya ÇAKMAK
36173621019

İnşaat Mühendisliği Anabilim Dalı

Tez Danışmanı: Doç. Dr. Önder Halis BETTEMİR

Ağustos 2021

TEŞEKKÜR VE ÖNSÖZ

Bu tez çalışmasının her aşamasında yardım, öneri, bilgi, tecrübe ve desteklerini esirgemedi beni her konuda yönlendiren danışman hocam Sayın Doç. Dr. Önder Halis BETTEMİR'e,

Çalışmalarında ayrıca tüm hayatım boyunca olduğu gibi bu çalışmalarım süresince benden her türlü desteklerini esirgemeyen aileme, özellikle annem Türkan Çakmak'a,

teşekkür ederim.



ONUR SÖZÜ

Doktora veya yüksek lisans tezi olarak sunduğum “Sınırlı Kaynaklı İş Programı Problemlerinin Optimum veya Yakın Optimum Çözülmesi” başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığına ve yararlandığım bütün kaynakların hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Derya ÇAKMAK



İÇİNDEKİLER

TEŞEKKÜR VE ÖNSÖZ	i
ONUR SÖZÜ	ii
İÇİNDEKİLER.....	iii
ÇİZELGELER DİZİNİ.....	iv
ŞEKİLLER DİZİNİ.....	vi
SEMBOLLER VE KISALTMALAR	vii
ÖZET	viii
ABSTRACT	ix
1. GİRİŞ.....	1
1.1 Tezin Amaç ve Kapsamı.....	1
2. LİTERATÜR TARAMASI.....	3
2.1 Problemin Formül ve sınıflandırma şeması.....	12
2.2 Tek- modlu durum.....	16
2.3 Dal-sınır yöntemi.....	17
2.4 Alt sınırlar.....	21
3. VAKA ÇALIŞMALARI.....	40
3.1 Örnek Problem 1.....	40
3.2 Örnek Problem 2.....	52
3.3 Örnek Problem 3.....	53
3.4 Örnek problem 4.....	54
3.5 Örnek problem 5 ve 6	56
3.6 Örnek problem 7.....	58
3.7 Örnek problem 8.....	61
3.8 Örnek problem 9.....	63
3.9 Örnek problem 10.....	66
3.10 Örnek problem 11	69
3.11 Örnek problem 12.....	73
4. TARTIŞMA VE SONUÇ	80
5. KAYNAKLAR.....	81
ÖZGEÇMİŞ	88

ÇİZELGELER DİZİNİ

Çizelge 2.1: Wang (2017) tarafından incelenen önceliklerin listesi	11
Çizelge 2.2 : Önerilen algoritma ile elde edilen sonuçlar	31
Çizelge 2.3 : Önerilen algoritmanın performansının diğer algoritmalarla karşılaştırılması	31
Çizelge 3.1: Problem 1'in öncelik sıralama alternatifleri	51
Çizelge 3.2: Problem 2'nin kaynak kısıtı olmadan çözümü	52
Çizelge 3.3: Aktivite önceliklerinin oluşturulması	53
Çizelge 3.4: Problem 3'ün kaynak kısıtı olmadan çözümü.....	54
Çizelge 3.5: Aktivite önceliklerinin oluşturulması	54
Çizelge 3.6: Problem 4'ün kaynak kısıtı olmadan çözümü.....	55
Çizelge 3.7: Aktivite önceliklerinin oluşturulması	55
Çizelge 3.8: Problem 5'in kaynak kısıtı olmadan çözümü	56
Çizelge 3.9: Problem 5'in aktivite öncelikleri ve süreleri.....	57
Çizelge 3.10: Problem 6'nın kaynak kısıtı olmadan çözümü	57
Çizelge 3.11: Problem 6'nın aktivite öncelikleri ve süreleri.....	58
Çizelge 3.12: Problem 7'nin aktivite süreleri ve kaynak ihtiyaçları ile kaynak kısıtları kaldırıldığında elde edilen olay zamanları	59
Çizelge 3.13: Problem 8'in aktivite süreleri ve kaynak ihtiyaçları ile kaynak kısıtları kaldırıldığında elde edilen olay zamanları	61
Çizelge 3.14: A-B-D-G öncelik sıralaması ile B-E-H hattı kombinasyonun öncelik sıralaması	62
Çizelge 3.15: B-A-D-G öncelik sıralaması ile B-E-H hattı kombinasyonunun öncelik sıralaması	62
Çizelge 3.16: Problem 9'un aktivite süreleri ve kaynak ihtiyaçları	64
Çizelge 3.17: A-B-E-H-J öncelik sıralaması ile D-G hattı kombinasyonunun tabi öncelik sıralaması	64
Çizelge 3.18: B-A-E-H-J öncelik sıralaması ile D-G hattı kombinasyonunun öncelik sıralaması	65
Çizelge 3.19: B-E-A-H-J öncelik sıralaması ile D-G hattı kombinasyonunun öncelik sıralaması	65
Çizelge 3.20: Problem 10'un aktivite süreleri ve kaynak ihtiyaçları ile kaynak kısıtları kaldırıldığında elde edilen olay zamanları	67
Çizelge 3.21: B-C-F-I öncelik sıralaması ile E-H hattı kombinasyonunun öncelik sıralaması	67
Çizelge 3.22: C-B-F-I öncelik sıralaması ile E-H hattı kombinasyonunun öncelik sıralaması	68
Çizelge 3.23: A-D-G ve J-K hattı kombinasyonu	68
Çizelge 3.24: Problem 11'in aktivite süreleri ve kaynak ihtiyaçları	70
Çizelge 3.25: B-F-C-I-J-M ile E-H-L hatlarının kombinasyonundan elde edilen öncelik sıralamaları	71
Çizelge 3.26: B-C-F-I-J-M ve E-H-L hatları kombinasyonu öncelik sıralaması.....	72
Çizelge 3.27: Problem 12'nin aktivite süreleri ile kaynak ihtiyaçları.....	74

Çizelge 3.28: Örnek olarak A-F-D-G-M ile B-E-H-J hattının öncelik sıralaması.....	75
Çizelge 3.29: A-D-F-G-M ile B-E-H-J hattı kombinasyonunun öncelik sıralaması	76
Çizelge 3.30: A-D-G-F-M ile B-E-H-J hattı kombinasyonunun öncelik sıralaması	77
Çizelge 3.31: A-D-G-M-F ile B-E-H-J hattı kombinasyonunun öncelik sıralaması	77
Çizelge 3.32: 1276043 alternatif için öncelik sırasına göre kaynak çakışmasının olduğu aktivitelerin öncelik sıralaması	78



ŞEKİLLER DİZİNİ

Şekil 3.1: Örnek proje-1	40
Şekil 3.2: Veri giriş ekranı.	41
Şekil 3.3: Erken Olay Zamanı hesaplamasının hesap tablosunda gerçekleştirilmesi.	41
Şekil 3.4: Geç Olay Zamanı hesaplamasının hesap tablosunda gerçekleştirilmesi.	41
Şekil 3.5: Aktivitelerin Erken ve Geç Başlangıç Zamanlarının hesaplanması	42
Şekil 3.6: Toplam bolluk sürelerinin hesaplanması	44
Şekil 3.7: Başlangıç sürelerinin hesaplanması	44
Şekil 3.8: Bitiş sürelerinin hesaplanması	44
Şekil 3.9: Aktivite/Süre hesaplanması	44
Şekil 3.10: Çakışan Aktivitelerin belirlenmesi	45
Şekil 3.11: Aktivite önceliğinin tespit edilmesi	46
Şekil 3.12: En düşük önceliğin belirlenmesi	46
Şekil 3.13: C aktivitesinin 4 gün ertelenmesi sonucunda oluşan iş programı	47
Şekil 3.14: A-B-C-D alternatifinin ikinci deneme sonucunda oluşan iş programı	48
Şekil 3.15: A-B-C-D alternatifinin birinci deneme sonucunda oluşan iş programı	49
Şekil 3.16: A-C-B-D alternatifinin ikinci deneme sonucunda oluşan iş programı	49
Şekil 3.17: A-C-D-B alternatifinin birinci deneme sonucunda oluşan iş programı	49
Şekil 3.18: A-C-D-B alternatifinin ikinci deneme sonucunda oluşan iş programı	50
Şekil 3.19: C-D-A-B alternatifinin birinci deneme sonucunda oluşan iş programı	50
Şekil 3.20: C-D-A-B alternatifinin ikinci deneme sonucunda oluşan iş programı	50
Şekil 3.21: C-A-B-D alternatifinin birinci deneme sonucunda oluşan iş programı	50
Şekil 3.22: C-A-B-D alternatifinin ikinci deneme sonucunda oluşan iş programı	50
Şekil 3.23: C-A-D-B alternatifinin birinci deneme sonucunda oluşan iş programı	51
Şekil 3.24: C-A-D-B alternatifinin ikinci deneme sonucunda oluşan iş programı	51
Şekil 3.25: Problem 2'nin şebeke diyagramı	52
Şekil 3.26: Problem 3'ün şebeke diyagramı	53
Şekil 3.27: Problem 4'ün şebeke diyagramı	55
Şekil 3.28: Problem 5'in şebeke diyagramı	56
Şekil 3.29: Problem 6'nın şebeke diyagramı	57
Şekil 3.30: Problem 7'nin şebeke diyagramı	59
Şekil 3.31: Problem 8'in şebeke diyagramı	61
Şekil 3.32: Problem 9'un şebeke diyagramı	63
Şekil 3.33: Problem 10'un şebeke diyagramı	66
Şekil 3.34: Problem 11'in şebeke diyagramı	70
Şekil 3.35: Problem 12'nin şebeke diyagramı	73

SEMBOLLER VE KISALTMALAR

KKPÇP	: Kaynak Kısıtlı Proje Çizelgeleme Problemi
ÇMKKPPP	: Çok Modlu Kaynak Kısıtlı Proje Programlama Problemi
PSO	: Parçacık Sürüsü Optimizasyonu
SPŞ	: Seri Programlama Şeması
PPŞ	: Paralel Programlama Şeması
GA	: Genetik Algoritma
DN	: Düğüm Noktası
KKPPP	: Kaynak Kısıtlı Proje Programlama Problemi
PTAA	: Paralel Tabu Arama Algoritması
TA	: Tabu Arama
TAA	: Tabu Arama Algoritması
GİB	: Grafik İşleme Birimi
MİB	: Merkezi İşlem Birimi
PPD	: Performans Programlama Dizini
ÇPO	: Çoklu Proje Ortamı
ÇZK	: Çoklu Zaman Kısıtlaması
KPP	: Kaynak Portföy Problemi
KT	: Kaynak Tahsisi
AK	: Ağ Karmaşıklığı
KF	: Kaynak Faktörü
KG	: Kaynak Gücü
MKF	: Maksimum Kullanım Faktörü
TB	: Tavlama Benzetimi
YYA	: Yinelemeli Yerel Arama
DİY	: Daha İyi Yürüyüş
ÇFS	: Çoklu Faktör Sistemi
BVÇ	: Basit Vaka Çalışması
KDS	: Karmaşık Dinamik Senaryo
EEBT	: En Erken Bitiş Tarihi
MTA	: Maksimum Toplam Ardıl
KTDP	: Karışık Tamsayı Doğrusal Programlama
TMKKPPP	: Tek Modlu Kaynak Kısıtlı Proje Programlama Problemi
VZA	: Veri Zarflama Analizi
DDKASA	: Değiştirilmiş Değişken Komşu Arama Sezgisel Algoritması

ÖZET

Yüksek Lisans Tezi

SINIRLI KAYNAKLI İŞ PROGRAMI PROBLEMLERİNİN OPTİMUM VEYA YAKIN OPTİMUM ÇÖZÜLMESİ

Derya ÇAKMAK

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
İnşaat Mühendisliği Anabilim Dalı

88+IX sayfa

2021

Danışman: Doç. Dr. Önder Halis BETTEMİR

Kaynak kısıtlı proje çizelgeleme problemi özellikle imalat ve inşaat sektöründe yaygın olarak karşılaşılan bir problemdir. Proje tabanlı faaliyet gösteren inşaat sektöründe kaynak tahsisi her proje için baştan yapıldığı için planlamacılar kaynak kısıtlarının ihlalini düzenlerken en iyi çözümü bulmak için yeterli vakte sahip olmayabilirler. Ayrıca küçük ölçekli inşaat işlerini gerçekleştiren firmalarda optimizasyon alanında uzman inşaat mühendislerinin istihdam edilmesi düşük olasılıktır. Bunun sonucunda kaynak kısıtlarını ihlal etmeyen fakat optimum çözümden uzak bir çözüm elde edebilirler. Bu tez çalışmasında karmaşık optimizasyon yöntemlerini kullanmadan tüm proje çizelgeleme olasılıklarını deneyerek kaynak kısıtlı proje çizelgeleme problemini çözen bir algoritma geliştirilmiştir. Algoritma hesap tablosu üzerinde programlanmış ve küçük ölçekli projelerde denenerek çalıştığı gösterilmiştir. Kaynak kısıtlı proje çizelgeleme problemleri NP-Zor türü problemler olduğu için arama uzayı aktivite sayısının artması ile üstel biçimde arttığı için büyük problemlerde hesaplama süresi çok artmaktadır. Bu nedenle bu çalışmada küçük ölçekli problemler çözülmüştür. Fakat ileri çalışma olarak yöntemin hızlandırılması, uygun olmayan çözümlerin elenmesi ve paralel hesaplama ile daha hızlı çözümün elde edilmesi amaçlanmaktadır. Böylece daha büyük problemlerin çözümü mümkün olacaktır. Gerçekleştirilen vaka analizi çalışmalarından 6'sının tam sonuçlarının 1 dakikadan daha kısa sürede elde edilmesi, diğerlerinin ise makul hesap süresinde elde edilmesi ve yöntemin uygulanabilmesi için karmaşık optimizasyon yöntemlerinin bilinmesine gerek duyulmaması inşaat sektöründe önerilen algoritmanın uygulanabileceği gösterilmiştir.

Anahtar Kelimeler: Optimizasyon, Kaynak kısıtlı proje çizelgeleme problemi, proje yönetimi.

ABSTRACT

Master Thesis

LIMITED RESOURCE WORK PROGRAM OPTIMUM OR NEAR OPTIMUM SOLVING OF PROBLEMS

Derya ÇAKMAK

Inonu University
Graduate School of Nature and Applied Sciences
Department of Civil Engineering

88+IX pages

2021

Supervisor: Doç. Dr. Önder Halis BETTEMİR

Resource constrained project scheduling problem is a prevalent problem for manufacturing and construction sectors. Resource allocation is redone from the beginning in the construction sector where project management is implemented, therefore project planners may not have adequate time to obtain optimum solution. Moreover, employment of civil engineering who are talented in the optimization task is less probable in small-scale construction firms. As a result of this, resource overrun problems are solved without converging optimum solution. In this study, an exhaustive enumeration based algorithm which can solve resource constrained project scheduling problems without implementing complex optimization methods is developed. The algorithm is programmed on spreadsheet and small-scale problems are solved in order to represent the proposed algorithm can obtain the optimum solution. Search domain expands exponentially and solution time excessively prolongs when the number of activities in the project increases because the resource constrained project scheduling problem is NP-Hard. Therefore in this study small-scale problems are solved. However, as a future study speeding up the method, elimination of infeasible solutions, and parallel computing are intended. Thence solution of larger scale problems would be possible. It is considered that the construction sector may implement the proposed algorithm since among the twelve case study problems, six case studies are solved within one minute, the remaining problems are solved within reasonable time and the implementation of the proposed method does not require complex optimization algorithms.

Keywords: Optimization, resource-constrained project scheduling problem, project management.

1. GİRİŞ

Kaynak kısıtlı proje çizelgeleme problemleri (KKPÇP), kaynaklar üzerindeki kısıtları ihlal etmeden bir projeyi oluşturan tüm faaliyetlerin, öncel ilişkilerini dikkate alarak proje süresi en kısa olacak şekilde çizelgelenmesini amaçlar (Ulusoy, 2006). Bu sayede temin edilebilen kaynak miktarının sınırlı olduğu durumlarda en kısa sürede projenin tamamlanması sağlanır. Kalabalık şantiye kurulmadan uygun sürede projenin tamamlanmasını ve çok pahalı olan iş makinesi yatırımlarından önemli tasarrufların elde edilmesini sağladığı için KKPÇP inşaat sektörü açısından oldukça önemlidir. Kaynak kısıtlı iş programının oluşturulması bir üretim sürecinin, bir yapının ya da küçük üretim gruplarının programlanması ile ilgilidir ve sınırlı kaynaklar aktivitelere dayalı olarak zamanla tahsis edilir. Bu uygulamalar; inşaat mühendisliği, yazılım geliştirme gibi çeşitli endüstrilerde bulunabilir. Ayrıca yalın işletme kavramını uygulayabilmek için kapasite azalma yoluna giden sipariş usulü çalışan şirketler açısından proje programlama artan öneme sahiptir. Aynı şekilde, proje programlama araştırmacılar için çok cazip edicidir, çünkü; bu alandaki modeller zengindir ve bu yüzden çözüm bulmak zordur.

1.1 Tezin Amaç ve Kapsamı

Son yıllarda hem bilim hem de uygulamadaki proje programlamaya artan ilgi dikkat çekmektedir. Kaynak kısıtlı iş programı probleminde sınırlı kaynaklar iş programı ile birbirine bağımlı aktivitelere zamanla tahsis edilir. Proje programlama, sipariş usulü çalışan kapasitelerin yalın yönetim kavramlarını sağlamasından dolayı şirketler için önemlidir. Ayrıca proje programlama araştırmacılar için de çok cazip edicidir çünkü; bu alandaki problemler çok çeşitli olması bakımından zengindir ve çoğu yaygın olarak bilinen optimizasyon problemleri, genel proje programlama modellerinin özel bir durumudur. Örneğin kaynak-kısıtlı proje programlama problemi imalathane iş programı problemini özel bir durum olarak içerir. Proje programlama problemleri genelde hesap yükü açısından zordur.

Proje programlama hem uygulamada hem de bilimde son yıllarda gelişmiştir, farklı problem sınıflarını ayırmak için sayısız kısaltmalar üretilmiştir. Ayrıca aynı konuyu belirtmek için proje programlama araştırmacıları tarafından farklı semboller kullanılmaktadır. Bu alandaki modellerin standart formda olmaması nedeniyle konunun tamamı hakkında net bir görüşe sahip olmak bazen zordur. Son zamanlarda ilk kez

Herroelen vd. (1997) proje programlama için bir sınıflandırma şeması oluşturmuştur. Oluşturulan bu şema yaygın olarak kabul edilen makine programlama ile uyumlu değildir. Dolayısıyla bir taraftan makine programlama ile proje programlama arasında hala bir boşluk olup diğer taraftan her ikisine göre ortak bir formül ve bir sınıflandırma şeması vardır. Kaynak ortamının tanımını, aktivitelerin özelliklerini ve işlevini sırasıyla makine programlama ile uyumlu olacak ve şimdiye kadar ele alınan en önemli modelleri sınıflandırmaya izin verecek şekilde bir sınıflandırma şeması sağlanmaktadır.

Bu tezin diğer bir amacı problemin çözümüne önemli katkı sunan çalışmaları incelemektir. Elmaghraby (1995), Herroelen vd. (1997) ile Ozdamar ve Ulusoy (1995) tarafından yapılan çalışmalar problemin çözümüne önemli katkıda bulunulmuştur. Fakat kısıt yayılma tekniği son zamanlarda gelişmekte olup bu tekniklerin amacı çeşitli kısıtları yerine getirerek (ihlal etmeden) kısıt içeren problemleri çözmektir.

2. LİTERATÜR TARAMASI

Kaynak kısıtlı iş programı inşaat ve endüstri mühendisliği ile ilgili önemli bir problem olduğu için bu problem üzerinde çok sayıda çalışma bulunmaktadır. Zhang vd. (2004) parçacık sürüsü optimizasyonu tabanlı algoritmalarla KKPÇP çözümünü gerçekleştirmiştir. Problemin ele alınışında öncelik tabanlı ve permütasyon tabanlı gösterim olmak üzere iki parçacık gösterimini göz önünde bulundurularak, kaynak kısıtlı proje çizelgeleme problemleri için parçacık sürüsü optimizasyonu kalıpları geliştirilmiştir. Permütasyon tabanlı PSO'da parçacıkların uçtuğu esnada, permütasyon uygulanabilirliği ve öncelik kısıtlarını ele almak için hibrit parçacık-geliştirme mekanizması önerilmiştir. Parçacıklar tarafından temsil edilen uygulanabilir çözümler, parçacıkların uçuşu esnasında değerlendirilebilmesi amacıyla tüm aktivitelerin kaynak tahsisi ve başlangıç sürelerinin gösterimleri de dâhil olmak üzere permütasyon veya öncelik formuna uygun şemalara dönüştürülmesi gerektiği saptanmıştır. 480 adet 30 aktiviteli projelerden oluşan deneme problemlerinin çözüm sonucuna göre kaynak kısıtlı proje çizelgeleme problemi çözümünde permütasyon tabanlı PSO, öncelik tabanlı PSO'dan daha üstün performans göstermiştir ve PSO-tabanlı yöntem diğer üst sezgisel, genetik algoritma, tavlama benzetimi gibi yöntemlere göre daha kısa proje süreleri sunmuştur.

Kanıt vd. (2005) kaynak kısıtlarının yapı maliyetine etkisini ele alınmıştır. Yapım süreci boyunca kullanılması gereken kaynaklarda kısıt bulunması halinde, proje süresi ve maliyetinin bu kısıtlamadan nasıl etkilendiği araştırılmıştır. Bunun için yapımı gerçekleştirilmiş bir sağlık ocağı projesi incelenmiştir. Elde edilen sonuçlara göre, süre şartına bağlı programa uygulanan kaynak kısıtlı proje süresini uzatmakta, ancak projenin yükleniciye olan maliyetini azaltmaktadır. Proje süresindeki değişim $y = -14,29x^2 + 216,14x + 218,64$ denklemiyle; proje maliyetindeki değişim ise $y = 11,08x^2 - 18,77x + 265,89$ denklemiyle ifade edilebilmektedir. Denklemlerde x işçilik kısıtını belirtmektedir.

Talbot (1976) tarafından önerilen kapalı numaralandırma yöntemi, aday problemlerin dikkate alınma biçiminden dolayı incelenen diğer iki yaklaşımdan çok daha az bilgisayar belleği kullanımını gerektirmektedir. Stinson (1976) tarafından önerilen dalsınır çözüm yöntemi, bilgisayar belleğinin sınırlamadığı durumlarda minimum hesaplama süresinde çözümler üretmekte ve tercih edilen çözüm yaklaşımı olması muhtemel görünmektedir.

Davis (1978) sınırlı numaralandırma yönteminde minimum hesaplama süresi ile en uygun çözümü üretmesi olası görünmektedir. Her bir problem için en iyi beş bağımsız değişken regresyon modelinin kullanıldığı ve her problemi çözmek için gereken hesaplama süresini tahmin etmek için bir ek deney gerçekleştirilmiştir. Elde edilen regresyon sonuçlarına dayanarak, minimum hesaplama süresini veren yöntem 110 test probleminin 98'inde veya bunların yaklaşık yüzde 89'unda doğru olarak tahmin edilmiştir.

Kolisch ve Rainer (1994) klasik kaynak kısıtlı proje programlama yöntemi için yaygın olarak bilinen iki spesifik sezgisel yöntemin; seri ve paralel programlama şemalarının detaylı algoritmik tanımları ve teorik sonuçlarını ele almıştır. Seri ve paralel kaynak kısıtlı proje programlama yöntemlerinin, kaynak kısıtlı proje programlama problemlerinde nasıl bir etki oluşturduğu incelenmiştir. Geniş kapsamlı bir hesaplama çalışması, belirleyici tek geçişli ve olasılıklı çok geçişli (örnekleme) yöntemi olarak uygulandığında her iki şemayı da karşılaştırmaktadır.

Sistematik olarak oluşturulmuş 360 örneğe dayanan derinlemesine bir hesaplama çalışması yapılmıştır. Programlama şemalarının performansını incelemek için aşağıdaki istatistiksel model (beş faktörlü) kullanılmıştır. Oluşturulan denklemler Eşitlik 1'de sunulmuştur.

$$\begin{aligned} DEV_{abcno} &= \delta(PR_a, SS_b, Z_c, NC_m, RF_n, RS_o) + \varepsilon_{abcmno} \\ CPU_{abcmno} &= \theta(PR_a, SS_b, Z_c, NC_m, RF_n, RS_o) + \varepsilon_{abcmno} \end{aligned} \quad (1)$$

DEV_{abcmno} ve (CPU_{abcmno}) , m'ninci ve o'nuncu yüzdeler seviyelerine sahip örnekler olduğunda optimum çözümden (CPU-saniye cinsinden ortalama çalışma süresi) ortalama sapmayı gösterir. Problem parametreleri NC, RF ve RS, sırasıyla PR, SS ve Z prosedür parametrelerinin a'nıncı, b'ninci ve c'ninci seviyeleri ile çözülmektedir. Hatalar için karşılıklı olarak bağımsız oldukları ve her birinin aynı sürekli popülasyondan alındığı varsayılmaktadır. NC, RF ve RS'nin her bir kombinasyonu için toplam $3 * 4 * 3 * 10 = 360$ probleme eşit 10 örnek üretilmiştir. 3600 CPU-saniyelik bir zaman sınırı uygulanarak, bu örneklerin 308'i için 80386SX işlemci, matematiksel yardımcı işlemci ve 15 MHz işlemciye sahip kişisel bir bilgisayarda en uygun çözüm elde edilmiştir. 308 problemin her biri yöntem parametreleri, programlama şeması, öncelik kuralı ve örnek boyutunun her seviye kombinasyonu ile iyileştirilmiştir.

Seri programlama şeması (SPŞ) aktif iş programları üretirken paralel programlama şeması (PPŞ) erteleme olmayan iş programları meydana getirmektedir. PPŞ, zor problemlerde daha iyi performans göstermekte (yüksek kaynak faktörlü veya düşük kaynak gücü) SPŞ ise kolay problemlerde (düşük kaynak faktörlü veya yüksek kaynak gücü) daha iyi performans göstermektedir.

PPŞ, belirleyici tek geçişli ve olasılıklı çok geçişli (örnekleme) yöntemi 40'a kadar örnekleme için daha iyi olduğu ortaya çıkmıştır. Kaynak bazlı problem parametrelerinin; kaynak faktörü ve kaynak gücü, tek geçişli ve örnekleme sezgisel çözümü üzerinde kayda değer bir etki yaratmaktadır. Sonuç itibarıyla, yüksek kaynak faktörü ve düşük kaynak gücünün zayıf performans gösterilmesine neden olacağı vurgulanmıştır.

Merkle vd. (2002) kaynak kısıtlı proje çizelgeleme probleminin (KKPÇP) çözümünü karınca kolonisi optimizasyonu (KKO) ile gerçekleştirmiştir. KKO yaklaşımının genel fikri karınca kolonisinin yiyeceğe erişimini en kısa yolu izleyerek bulmasını taklit etmekte ve her bir çözüm için bir karınca kullanılmaktadır. Feromon katsayısı ve buharlaşma katsayılarının farklı değerler alması ile yakınsamanın değişimi incelenmiştir. KKO, 396 örnek problemde 130 en iyi çözümü bulabilmiştir, fakat incelenen 396 problemin yaklaşık 2/3'ü optimal çözülemese de genetik algoritma, tavlama benzetimi, tabu araması gibi diğer farklı örnekleme yöntemi algoritmalarıyla kıyasla daha iyi bir performans gösterdiği sonucuna varılmıştır.

Hartmann (1998) kaynak kısıtlı proje programlama problemini (KKPPP) genetik algoritma (GA) ile çözmüştür. Genetik algoritma, karmaşık problemlerin çözüm alternatifleri arasından en uygun çözümü bulmak için kullanılan üst sezgisel bir yöntemdir. Bu algoritma, bulunduğu çevrede güçlü olanları seçme, çaprazlama ve mutasyon gibi evrimsel süreçlerden faydalanarak bir sistem oluşturmaktadır. Çalışmada ProGen tarafından üretilen 30 ve 60 aktiviteli projeler kullanılmıştır. Her projede dört yenilenebilir kaynak bulunmaktadır. Her problem kümesi ağ karmaşıklığı, kaynak faktörü ve kaynak miktarının değiştirilmesiyle sistematik olarak oluşturulan 480 projeden oluşmaktadır. Permütasyona dayalı GA yaklaşımı 30 aktiviteli projelerde en iyi sonucu vermiştir. Permütasyon kodlaması için birkaç alternatif genetik operatör arasında, bir sıralama seçim stratejisi, 0,05'lik bir mutasyon olasılığı ve öncelik fizibilitesini koruyan iki noktalı bir çaprazlama operatörü seçilmiştir. Başlangıç popülasyonu rastgele öncelik kuralı yöntemi ile belirlenmiş ve her örnek için 1000 çizelge hesaplanmıştır. Sunulan yeni GA'nın diğer GA'lardan ve yaklaşımlardan daha iyi performans gösterdiğini ortaya koymuştur. Bu

sonuç, yeni GA yaklaşımını gerçek dünyadaki proje programlama problemlerine uygulamayı önermektedir.

Patterson (1984) birden fazla kısıtlı kaynaklı iş programlama probleminin çözümünü incelenmiştir. Tek projenin çözümü için üç optimizasyon yaklaşımı, tek kısıtlı, çok kısıtlı kaynak ve proje çizelgeleme problemi karşılaştırılmaktadır. Bu kıyaslanan yöntemlerin tamamı numaralandırma bazlı olmakla birlikte aday problemlerin değerlendirilmesi, dikkate alınma sırası ve alt kısmi programları tanımlamak ve atmak için kullanılan yöntemler açısından farklılıklar göstermektedir.

Özdamar (1995) kaynak kısıtlı proje programlama problemini (KKPPP) belirli bir amaç ve kısıtlara göre sınıflandırıp, zaman-maliyet tabanlı amaçlar ile yenilenebilir-yenilenemez kaynak kısıtlarını ele almışlardır. Literatürden derlenen ve optimum çözülmüş 78 problem kümesi ve 110 kıyaslama amaçlı problem kümesini iyi bilinen dağıtma kuralları ile problem kısıtlamalarını bir seçim temeli olarak kullanarak, karar verme sürecinden oluşan bir programlama algoritması ile analize tabi tutulmuştur.

Kısıtların çok sayıda sıfır-bir değişken ile tanımlanması araştırmacıları KKPPP çözümü için matematiksel programlama yerine dal-sınır yöntemini geliştirmeye yönlendirmiştir. Bir dal-sınır algoritmasının hesaplama başarısı, dallanma tekniğine ve alt sınırının dayanımına bağlı olmaktadır. Dallanma şeması, kaynak çakışmalarını gidermek için devam etmekte olan bazı aktivitelerin ertelenmesi olarak tanımlanmaktadır.

Christofides (1987) ,40 ile 25 aktiviteli problemleri 3 kaynak türü için, Alvarez-Valdes ve Tamarit (1989) ise; 38, 51 ve 103 aktiviteli problemleri 6 kaynak türü için çözmüşlerdir. Özdamar ve Ulusoy (1995) geniş kapsamlı ilk araştırmada bulunan dallanma şemasını kullanarak budama yöntemiyle sezgisel olarak çalışmaktadırlar. Sezgisel tarama her iki problem kümesinde de optimumdan ortalama %2 sapma ile sonuçlanmıştır. Deneylerde, ağacın sezgisel bir şekilde budanması ve alt sınır hesaplamaları gelecekteki kaynak çatışmalarına dayandığı ve erken budama ile sonuçlandığı, üretilen düğüm sayısının ise kısıtlı kaynak problemleri için rapor edilenlerden daha fazla olduğu gözlenmiştir.

Peteghem (2013) çok modlu kaynak kısıtlı proje programlama problemini (ÇMKKPPP) çözmek için literatür de bulunan meta sezgisel çözüm yöntemlerine genel bir bakış sağlanmakta ve bu yöntemler arasında eşit kriterlere dayalı karşılaştırmalar yapılmaktadır. Çalışmanın amacı üç yönlü olarak sunulmaktadır: ilk olarak, mevcut meta

sezgisel yöntemlere genel bir bakış verilmekte ve her yöntemin özelliklerine göre bir sınıflandırma yapılmaktadır. İkinci olarak, mevcut kıyaslama veri kümelerinin eksikliklerini gidermek için yeni bir veri kümesi önerilmektedir. Son olarak, üst-sezgisel yöntemleri projenin ağ ve kaynak parametrelerini göz önüne alarak karşılaştırmaktır.

ÇMKKPPP, kaynak kısıtlı proje programlama probleminin (KKPPP) genelleştirilmiş hali olarak tanımlanmaktadır. ÇMKKPPP' de kaynakların üç farklı kategorisi ayırt edilebilmektedir: insan gücü ve makineler gibi birim zaman başına yenilenebilir kaynaklar sınırlıdır, yenilenemez kaynaklar projenin tamamında sınırlıdır (ör. bütçe) çift kısıtlı kaynaklar hem birim zaman hem de tüm proje için sınırlı olmaktadır. ÇMKKPPP'nin amacı, her bir aktiviteye uygun bir yapım seçeneği atayıp kaynak ve öncelik kısıtlarına bağlı olarak en kısa proje süresinde mevcut projeyi planlamaktır. ÇMKKPPP, amacı uygulanabilir bir mod atama listesi oluşturmak olan mod atama problemi ve amacı problemin yapısını en aza indirgemek olan bir zamanlama problemi olarak iki alt probleme ayrılabilir.

Bukata (2014) kaynak kısıtlı proje programını (KKPPP) makul bir sürede nitelikli çözümler bulmak için paralel tabu arama algoritmasını (PTAA) ele almıştır. PTAA, bir yerel arama algoritması olup başlangıç çözümünden başlamakta ve yerel bir optimum değere ulaşılan kadar en iyi komşu hareketlerini uygulayarak bu çözümü yinelemeli olarak geliştirmektedir. Tabu arama (TA) meta-sezgisel yöntem ilk olarak 1986 yılında Glover(1986) tarafından önerilmiştir. TA üst-sezgisel yöntemi, TA'nin temel terimlerinin bir tabu listesi, aspirasyon kriterleri, çeşitlendirme ve yoğunlaşma olarak tanımlanmıştır.

Küçük problemlerin (yaklaşık 30 aktiviteye kadar) dal-sınır algoritması gibi kesin yöntemlerle optimal olarak güvenilir bir şekilde çözümlenebilmektedir. Sezgisel veya meta sezgisel tarama, özellikle grafik kartları veya grafik işleme birimleri (GİB) gibi paralel donanım kullanılarak daha hızlandırılabilir ve daha büyük problem örneklerinin tatmin edici bir şekilde çözümü için daha uygun olacağı düşünülmektedir.

Grafik kartları (grafik işleme birimi) için PTAA, çözüm kalitesi ve saniyede gerçekleştirilen çözüm sayısı açısından mevcut diğer TA yaklaşımlarından daha iyi performans gösterdiği belirtilmiştir. Grafik kartları için algoritma merkezi işlem birimi (MİB) için optimize edilmiş paralel / sıralı algoritmadan 10.5 / 42.7 kat daha hızlı olduğu belirlenmiştir. Paralel / sıralı, MİB algoritmasına kıyasla aynı kalitede çözümler sırasıyla 5,4 / 22 kat daha hızlı elde edilmektedir.

KKPPP'yi ele alan standart karşılaştırma örnekleri üzerinde yapılan deneylerde, önerilen çözümün avantajları ortaya koyulmaktadır. Çözümlerin kalitesinin çok iyi olduğu ve diğer TA uygulamalarından daha iyi performans gösterdiği gözlemlenmiştir. GİB algoritmasının çok etkili olduğu kanıtlanmış ve orta menzilli GİB, optimize edilmiş paralel merkezi işlem birimi (PMİB) sürümünden önemli ölçüde daha hızlı olduğu belirtilmiştir. NVidia Geforce GTX 650 Ti GBİ, J120 veri kümesi için saniyede bir milyondan fazla çözümü değerlendirebildiği belirlenmiştir. GBİ'lerin gelecekte yöneylem araştırmasında daha yaygın olarak kullanılması öngörülmüştür.

Bhaskar (2010) bulanık aktivite süresiyle kaynak kısıtlı proje programlama problemini (KKPPP) ele almıştır. Problemi çözmek için tekrarlamalı olmayan bir sezgisel yöntem önerilmiştir. Bu yöntem, yeni bir programlama şeması için öncelik kuralına dayanmaktadır. Bu öncelik kuralına performans programlama dizini (PPD) denilmektedir. PPD, aktivitelerin kritikliğinin yanı sıra mevcut kaynağın verimli bir şekilde kullanılmasına dayanmaktadır. Projenin kritik yolunu bulmak için mesafeye dayalı sıralama yöntemi kullanılmıştır.

Bhaskar çalışmasında PSPLIB'den 30 aktiviteli problem kümesini kullanılmıştır. Bu veri kümesi farklı boyutlarda birkaç problem içermektedir. Bu kümeden 20 problem alınmıştır ve ele alınan yönteme göre bulanık aktivite sürelerinde 20 problem üretilmiştir. Ağ karmaşıklığı, kaynak faktörü ve kaynak gücü parametreleri kullanılarak PSPLIB örnekleri oluşturulmuştur. PSPLIB'den seçilen örneklerin dört tür yenilenebilir kaynağı olmaktadır. Bhaskar tarafından seçilen 20 problem ilk 10 örnek için ağ karmaşıklığı 1,5, kaynak faktörü 0,25 ve kaynak gücü, ilk 10 örnek için 0,2 ve sonraki 10 örnek için 0,5'dir.

30 aktivitedeki tüm problemler için en uygun çözüm bilinmesine rağmen bulanık aktivite süreleri olan problemler için elde edilen çözümün sadece belirleyici aktivite süreleri olan problemler için en uygun çözümlerle karşılaştırılmasının yararlı olunmayacağı belirtildiğinden, problemler için alt sınır hesaplanmaktadır ve önerilen sezgisel tarama ile elde edilen sonuçlar alt sınırlarla karşılaştırılmaktadır. Alt sınırlar, kaynak kısıtlamaları kaldırılarak sorunlar çözülerek hesaplanmaktadır.

Beşikçi (2013), kaynak kapasitelerinin genel bir kaynak bütçesinden belirlendiği ve çoklu proje ortamındaki (ÇPO) kaynakların projeler arasında paylaşamadığı ve tahsis edilmesi gereken ÇPO için yeni bir yaklaşımı ele almaktadır. Bu yaklaşım kaynak portföy problemi (KPP) olarak adlandırılmaktadır. KPP, belirli bir toplam bütçeye tabi olan kaynak

tahsisi (KT) prensibi kapsamındaki faaliyetlerin kaynak gereksinimlerini sağlamak için kaynak seviyelerinin (kapasiteler) belirlenmesi ile ilgili olmaktadır. KPP için çözüm yaklaşımları bir dizi test problemi kullanılarak test edilmektedir, ağ karmaşıklığı (AK) ve maksimum kullanım faktörüdür (MKF).

Çalışmada, KPP'nin genel matematiksel formülasyonu önerilmiştir ve çözümü için iki farklı genetik algoritma (GA) yaklaşımında yeni bir iyileştirme sezgisel olarak tanımlanmış ve kullanılmıştır. Önerilen GA yaklaşımları arasındaki fark, kaynak tahsisi (KT) ve kaynak portföy (KP) alanlarını keşfetme yolu ile ilgili olmaktadır. Eşzamanlı GA yaklaşımı KT ve KP boşluklarını aynı anda aramaktadır. Deneysel sonuçlar, iki fazlı GA yaklaşımının, çözüm süresi ve kalitesi açısından önerilen çözüm yaklaşımları arasında genel olarak en iyi sonuçları verdiğini göstermektedir.

Önerilen çözüm yönteminin önemli bir yönü, çeşitli yönetim düzeylerinde karar vericiye yararlılığı olarak belirtilmektedir. Bu yöntem, çoklu proje problem ortamında kaynak kapasitesi kararlarıyla ilgili farklı türde bilgiler vermektedir. Genel kaynak kapasitelerinin tercih değerleri, kaynakların mevcut projelere göre göreceli önemini belirtmekte ve bu nedenle, bu kaynakların mevcudiyetinde bir aksaklık olması durumunda kaynakların kritikliğini göstermektedir. Diğer önemli sonucun, KT değerlerinin sorunlu ortam için kaba kapasite programlamasına karşılık gelmesi ve çözümün aynı zamanda her bir proje için, temel bir çizelge olarak kullanılabilen ayrıntılı programlama sağlanması olarak belirtilmektedir.

Laurent (2017), klasik kaynak kısıtlı proje programlama problemini (KKPPP) genişleten çeşitli bölgeler arasında kaynak havuzlaması olan yeni bir problem uzantısı olan çoklu bölge KKPPP'yi ele almıştır. İki farklı kabul kriteri ile yerel arama (YA), tavlama benzetimi (TB) ve yinelemeli yerel arama (YYA), TB tipi kabul kriteri ve daha iyi yürüyüş (DİY) olmak üzere dört yöntemle çözüm aramıştır. En uygun çözümün bilindiği klasik KKPPP'nin 30, 60, 90 ve 120 aktiviteli literatür örneklerini çözerek çözümleme yöntemleri test edilmiştir. YYA ve TB de %90'dan daha fazla en iyi çözüm elde edilmiştir. İkinci olarak, önce küçük örnekler çözülmüş, sonra literatür örnekleri probleme uyarlanmıştır ve çözülmüştür. Deneyler 2,40 GHz @ Intel (R) Xeon (R) CPU E7-8870 işlemci üzerinde hesaplanmıştır. Matematiksel model IBM ILOG CPLEX Optimization Studio v 12.4 ile çözülmüştür. Üst-sezgiler JAVA 1.7'de uygulanmıştır. En iyi sonuçlar TB ve YYA üst-sezgileri ile elde edilmiştir. İlk önceliğin, önerilen yöntemin büyük örneklerde geliştirilmesi olduğu belirtilmiştir. Bir veya daha fazla vektör içeren çözüm yapısı

kullanıldığı takdirde sonuçlar üzerindeki etkisinin de görülebileceği belirlenmiştir. Diğer bir bakış açısının ise, bölgelerdeki farklı görev dağılımlarını keşfetmek için nüfus tabanlı üst-sezgisel yöntemlerin kullanılması olduğu belirtilmiştir.

Araúzo (2010), proje yönetimi literatürünün büyük ölçüde bireysel projelerin yönetilmesiyle ilgilenilmesine rağmen, uygulamada firmalar genellikle dinamik ve karmaşık çok projeli ortamlarda çalıştığını belirtmiştir. Bu nedenle firmaların değişen koşulları atlatması ve uyum sağlama yeteneğinden yararlanan çoklu proje ortamlarında bağlantılı dinamik programlama ve kontrol için faktör tabanlı bir yaklaşım ele alınmaktadır. Çoklu faktör sistemi (ÇFS): proje yöneticileri ve kaynak yöneticileri olmak üzere iki tür faktöre sahiptir. Mevcut yaklaşımda, yinelemeli fiyat düzenlemesi kaynak kapasitesi çatışmalarına dayanmaktadır. ÇFS projeleri dinamik olarak kaynaklara tahsis etmektedir ve değer, kârlılık, zamanlama ve uygulamalı bilgi açısından mevcut portföy üzerindeki etkisini dikkate alarak proje kabul / ret kararını vermektedir. Firma için hangi kaynakların en değerli olduğunu keşfetmenin nasıl mümkün olduğunu göstermektedir.

Sistem basit vaka çalışması (BVÇ) ve karmaşık dinamik senaryo (KDS) gibi farklı senaryolarda sistem performansını kontrol etmek için uygulanmış ve simüle edilmiştir. BVÇ’de sırasıyla C1, C2 ve C3 yeterliliklerine sahip üç farklı kaynağı (R1, R2 ve R3) ele alınmıştır. 5 projeden (P1, P2, P3, P4, P5) oluşan bir portföyü ve her bir projeyi tamamlamak için gereken görevler gösterilmiştir. Her görev, ilgili yetkinlik ve tamamlanması beklenen standart süre ile tanımlanmıştır. Bu deneyde P5 projesi kabul edilmiş ve yürütülmüştür ve toplam değer 55.700’den 69.369 para birimine yükseltilmiştir. Bu, sistemin küresel performansı gerçek zamanlı olarak artırmak için R2 kaynağının esnekliğini kullanabildiğini göstermiştir. KDS’da her 20 birimde bir sisteme gelen 12 proje ele alınmıştır. DD1 ve DD2 başlangıç tarihleri yer almaktadır. Kaynaklar ve yeterlilikler BVÇ çalışma örneğine benzemektedir. Bu senaryoda, sistem performansının uygun olduğunu; ÇFS, küresel verimliliği arttırmak için kaynak esnekliğini kullanılabildiğini göstermiştir. Projeler ve kaynaklar aracı olarak modellenmiştir. Projeler, programlanan planlı çalışmalarını yerine getirmek için kaynak isterken, kaynaklar yeteneklerini ve iş gücünü sunmaktadır. Bu çerçevede, proje portföy yönetimine (PPY) ve yeni projeleri kabul etme / reddetme kararında değerlendirmeye olanak tanımıştır.

Wang (2017), belirsiz faaliyet süreleriyle kaynak kısıtlı çoklu proje programlama problemi (KKÇPPP) için öncelik kurallarının performansı ele alınmıştır. İncelediği öncelik kuralları aşağıda Çizelge 2.1’de verilmiştir.

Aktivite süresinin rastgele bir değişken olduğu varsayılmış ve belirsizliklerin bulunduğu bir ortamda öncelik kurallarının performansını analiz etmek için iki yeni dayanıklılık ölçütü önerilmiştir. Problemi çözmek ve proje özellikleri ile öncelik kurallarının performansı arasındaki ilişkiyi araştırmak için faktöriyel deney tasarlanmıştır. Kalite ve dayanıklılık arasında bir denge ilişkisi araştırılmış ve hem proje hem de portföy yöneticileri açısından en iyi öncelik kuralları önerilmiştir.

Belirsizliğin öncelik kurallarının uygulanması üzerindeki etkisini göstermek için 5 belirsizlik düzeyi kullanılmıştır. 1260 proje üzerinde faktöriyel deney yapılmıştır. İlk olarak, öncelik kurallarının genel performansı analiz edilmiştir ve kalite ile dayanıklılık önlemleri arasındaki denge ilişkisine dayanarak, proje ve portföy yöneticileri için ilgili Pareto sınırları elde edilmiştir. Genel olarak, yüksek belirsizlik seviyelerinin performansı, küçük harfe göre ortalama değere daha konsantre olduğu belirlenmiştir. Sonuç olarak, öncelik kurallarının performansının kullanılan hedeflere göre önemli ölçüde farklı olduğu gözlemlenmiştir. İkinci olarak, problem özelliklerinin öncelik kurallarının ortalama performansı üzerindeki etkisi incelenmiştir. Son olarak, en iyi öneri dört tür önlem dikkate alınarak verilmiştir ve sonuçlar ayrı proje ve portföy yöneticileri ile karşılaştırılmıştır. En erken bitiş tarihinin (EEBT) proje yöneticileri için en iyi kural olduğu , maksimum toplam ardılın (MTA) ise portföy yöneticileri için en iyi olduğu belirlenmiştir.

Çizelge 2.1: Wang (2017) tarafından incelenen önceliklerin listesi

Minimum gecikme (MİNG)
Maksimum gecikme(MAKG)
En küçük projeden en küçük aktivite (EKPEKA)
En büyük projeden en büyük aktivite (EBPEBA)
Minimum toplam iş kapasitesi (MİNTİK)
Maksimum toplam iş kapasitesi (MAKTİK)
MİNTİK ve en erken geç başlama süresi (EEGBS)(2 aşamalı kural)
MAKTİK ve en erken erken başlama süresi (EEGBS)(2 aşamalı kural)
İlk gelen ilk hizmet (İĞİH)
En kısa aktivite önceliği (EKUÖ)
Maksimum (en uzun) aktivite önceliği (MAÖ)
Rastgele (RAST)
En erken teslim tarihi (EETT)
Son gelen ilk hizmet (İĞİH)
Maksimum programlama baskısı (MPB)
Minimum geç bitiş süresi (MİNGBS)
Minimum en kötü olay durgunluğu (MİNEKOD)
Ağırlıklı aktivite kritikliği ve kaynak kullanımı (AAKKK)
Maksimum toplam ardıllar (MTA)
Maksimum kritik ardıllar (MKA)

Brucker (1998) Şimdiye kadar makine programlamasında yaygın olarak kabul edilen sınıflandırma ile uyumlu bir sınıflandırma şeması oluşturulmamıştır. Ayrıca, farklı semboller proje programlama araştırmacıları tarafından aynı konuyu belirtmek için kullanılmıştır. Dolayısıyla, bir yandan makine programlama diğer yandan proje programlama için proje çizelgelemede ortak bir formül ve bir sınıflandırma şeması oluşturma açısından her iki alanda da bir boşluk vardır. Nitekim, proje programlamasında gittikçe artan sayıda makale yayınlanmaktadır ve bilim çevresinde gerçekte neyin yeni olduğunu ve ne ile ilgili olduğunu takip etmek daha da zor olmaya başlayacaktır. Sırasıyla kaynak ortamının tanımını, aktivitelerin özelliklerini ve amaç fonksiyonunu makine programla ile uyumlu olacak şekilde ve şimdiye kadar ele alınan en önemli modelleri gruplandırma için uyumlu olacak bir sınıflandırma şeması sunulmaktadır. Ayrıca, birleştiren bir formül gösterimi önerilmektedir. Bu raporun ikinci amacı da son gelişmelerin bazılarını incelemektir. Tek modlu ve çok modlu durum için kesin ve sezgisel algoritmaları, zaman-maliyet ödünleşim problemini, minimum ve maksimum zaman gecikmeli problemleri, proje süresinin azaltılması dışındaki hedeflerle ortaya çıkan problemler ve stokastik aktivite süreleri ile ilgili problemler daha ayrıntılı incelenmektedir.

2.1 Problemin Formül ve sınıflandırma şeması

Proje programlama üzerine birçok farklı problem düşünülmüş ve bunun için genel bir sınıflandırma şeması ve ortak bir formül hazırlanmıştır. Bu şema makine programlama ve kaynak-kısıtlı makine programlamada yaygın olarak kabul edilen şema ile uyumlu olması bakımından önemlidir çünkü makine programlama modelleri proje programlama modelleri için özel bir durumdur. Bir projenin 1' den n' e (1; ;n) kadar aktivitelerden (işlerden) oluştuğu varsayılır. Kolaylık olması adına, genelde başlangıç (0) aktivitesine tek bir kukla ile başlanır ve ilave edilen (n+1) aktivitesi tek bir kukla ile sonlanır. Genelde, ok diyagramı (AON) ile tanımlanan projede oklar aktiviteleri ,düğüm noktaları ise olayları ve öncelik ilişkilerini sırasıyla temsil eder.

Öncelik kısıtlarının grafiği $G = (V ; E)$ olarak gösterilir aktiviteler arasındaki öncelik kısıtları alternatifler arasında $i \rightarrow j$ veya $(i; j)$ olarak belirtilir. Öncel (j) direkt öncel olan aktivite kümesini tanımlarken, ardıl (j) direkt ardıl olan aktivite kümesini tanımlar . J aktivitesinin işlem süresi P_j olarak verilir. Burada bir takım yenilenebilir R_p , yenilenemez R_v ve çift kısıtlı kaynaklar mümkündür. Planlama süresinin her periyodu için tekrar elde edilebilir. Mevcut bir kaynak yenilenebilir olarak tarif edilir. Planlama süresinin tamamı için mevcut bir kaynak yenilenemez olarak tarif edilir. Yenilenebilir ve

yenilenemez kaynakları kapsayabileceğinden çift kısıtlı kaynak kavramı atlanmaktadır. Periyot başına j aktivitesinin kullanacağı yenilenebilir k kaynağı r_{jk}^p ile gösterilirken , her periyotta mevcut k kaynağının elemanlarının sayısı (sabit) R_k^p ile gösterilir. Çok modlu durumda modların grubu, yani j aktivitesinin alternatif kümesi M_j olarak gösterilir. J aktivitesinin m modundaki tamamlanma süresi P_{jm} olarak gösterilir. Periyot başına (tüm kaynakların tüketimi) yenilenebilir (yenilenemez) k kaynaklı j aktivitesinin kullanımı $r_{jkm}^p(r_{jkm}^v)$ ile gösterilirken tüm planlama süresi için yenilenemeyen mevcut k kaynağının elemanlarının sayısı R_k^v ile gösterilir. Tek modlu durumda, yani $|M_j| = 1$ için ve $R^v = \emptyset$ için j olmasına rağmen ve kolaylık adına bu modda m indisini ve üst simge olan p atlanmaktadır.

J aktivitesinin başlangıç zamanı ve bitiş zamanı sırasıyla S_j ve C_j ile gösterilir. Sonuç olarak, $S = (S_1; \dots; S_n)$ bir programlamadır ve $C = (C_1; \dots; C_n)$ tamamlanma sürelerinin vektörüdür. Zaman kısıtlarını açmadan uygulanabilen dizi S_T ile gösterilir. Kaynak kısıtlarını açmadan uygulanabilen kaynak programları grubu S_R ile gösterilir ve $S = \frac{S_R}{S_T}$ uygulanabilen programlarını gösterir. Zaman içerisinde periyotlara endeksli bir t 'dir. Son olarak, i ve j aktiviteleri arasındaki minimum ve maksimum zaman gecikmeleri sırasıyla d_{ij}^{min} ve d_{ij}^{max} olarak gösterilir. Genelde, parametrelerin tam sayısal değer olduğu kabul edilir.

Çizelge 2 de küçük ilavelerle birlikte formüller özetlenmiştir. Makine programlama literatürlerinde de kullanılan $\alpha|\beta|\gamma$ – şeması genişletilir.

α : kaynak ortamı : makine programlama problemleri ve proje programlama problemlerini ayırt etmek için α -alanı PS (proje programlama) veya MPS (çok-modlu proje programlama) olarak belirtilir. Blazewicz vd.(1983) tarafından kullanılan formüle göre kaynak-kısıtlı makine programlaması için PS de m, σ, p ilave edilebilir.

Ayrıca çok-modlu proje programlama olması durumunda yenilenemeyen kaynaklar dikkate alınabilmektedir. MPS için benzer olarak $m, \sigma, p; \mu, \tau, \omega$ ilave edilebilir.

PS : proje programlama

MPS : çok-modlu proje programlama

PS m, σ, p m : kaynaklar, σ : g adet her bir kaynağın mevcut miktarı, her aktivitenin ihtiyaç duyduğu kaynak miktarı çoğu p birimleri

MPS $m, \sigma, p; \mu, \tau, \omega$ çok-modlu proje programlama m : yenilenebilir kaynaklar , σ : her bir kaynağın mevcut miktarı , her aktivitenin ihtiyaç duyduğu kaynakların çoğu p birimleri ;

μ : yenilenemeyen kaynaklar, τ : her bir kaynağın mevcut miktarı , her aktivitenin ihtiyaç duyduğu maksimum ω miktarı

Eğer $m, \sigma, p; \mu, \tau, \omega$;parametrelerin değerleri girişte belirtilir ise, PSm için , PSm, σbu değerlerin tamamı m, σ, p girdi de belirtilir ise rakamsal değerler ile ilgili parametrelerin yerine yazılır.

Temel Formüller

V : aktivite kümesi

n : gerçek aktivitelerin sayısı

E : öncel kümesi veya geçici kısıtlar kümesi

$G=(V, E)$: öncelik veya geçici kısıtların yönlü grafiği (şebeke diyagramı)

$i \rightarrow j, (i, j)$: öncelik kısıtları

(j) : j aktivitesinin direkt öncellerinin kümesi

(j) : j aktivitesinin direkt ardıllarının kümesi

p_j : j aktivitesinin işlem süresi(tamamlanma süresi)

R^p : yenilenebilir kaynaklar kümesi

R_k^p : yenilenebilir k kaynağının (temin edilebilir mevcut birimlerinin) sabit miktarı

r_{jk}^p : (j aktivitesinin periyod başına) yenilebilir k kaynağının kullanım miktarı

M_j : j aktivitesinin modlarının kümesi

p_{jm} : m modunda ki j aktivitesinin işlem süresi (tamamlanma süresi)

R^v : yenilenemez kaynakların kümesi

R_k^v : yenilenemeyen k kaynağının toplam (temin edilebilir mevcut birimlerinin) miktarı

r_{jkm}^p : m modunda gerçekleştirildiğinde j aktivitesinin periyod başına yenilenebilir k kaynağı kullanım miktarı

r_{jkm}^v : m modunda gerçekleştirildiğinde j aktivitesinin yenilenemez k kaynağı tüketim miktarı

S_j : j aktivitesinin başlangıç zamanı

$S = (S_1, \dots, S_n)$: iş programı

C_j : j aktivitesinin tamamlanma süresi

$C = (C_1, \dots, C_n)$: tamamlanma süreleri vektörü

S_T : süre-uygulanabilir iş programları kümesi (süre kısıtlarını ihlal etmeyen)

S_R : kaynak-uygulanabilir iş programları kümesi (kaynak kısıtlarını ihlal etmeyen)

$S=S_R \cap S_T$: uygulanabilir iş programı kümesi

$r_k(S,t)$: t zamanında S iş programında k kaynağının tüketim miktarı

d = proje süresinin son teslim tarihi (süresi)

T,t : sürelerin endekslenmesi için zaman tercihi

$t=1,2,\dots,T$: periyotlar

$[t-1,t]$: t periyoduna tekabül eden zaman aralığı

$\frac{d_{ij}^{min}}{d_{ij}^{max}}$: i ve j aktivitelerinin başlaması arasındaki minimum ve maksimum zaman

gecikme(ara)

Aynı şekilde parametrelere değer girilmez PS ve MPS için sırasıyla PS ve MPS gösterimi kullanılabilir.

Örnekler:

$PS_{m,1,1}$: m adet kaynak türü var, mevcut her bir kaynak için 1 birim temin edilebilmekte her aktivitede ki kaynak için en fazla 1 birimi kullanmaktadır.

$PS_{m,\infty}$: m adet kaynak türü var, temin edilebilen kaynak miktarı sonsuzdur.(yani belirgin bir kaynak kısıtı yoktur. Örneğin; kaynak dengeleme)

PS_1 : tek kaynak

β : Aktivite özellikleri: genel öncel kısıtlar vs. için makine programlamada yerleşmiş olan formüller ve gösterim kullanılır. p_j işlem süreleri gibi,

$p_j=1$ tüm işlem süreleri (aktivite süreleri) 1'e eşittir.

p_j = stokastik işlem süreleri

d = projenin son teslim tarihi (süresi)

prec : aktiviteler arasındaki öncelik kısıtlardır

zincirler(bağlantılar), ağaçta,ağaçtan.ağaç... aktiviteler arasındaki öncelik ilişkileri zincirler(bağlantılar), ağaçta,ağaçtan.ağaç aracılığıyla belirlenir.

temp : genel zamansal kısıtlamalar aktiviteler arasındaki minimum ve maksimum başlangıç-başlangıç zamanı gecikmeleri ile verilir.

Y : Amaç fonksiyonu : birçok durumda olduğu gibi makine programlama için formüllere uyarak amaç fonksiyonları tanımlanabilir.

$C_{max}, L_{max}, \sum w_j C_j$ vb. gibi klasik amaç fonksiyonlarının yanı sıra sonraki kriterler göz önüne alınabilir. Örneğin :

$\sum c_j^F \beta^{c_j}$ net bugünkü değer (c^F nakit akışı, β iskonto faktörü)

$\sum c_k f(r_k(S,t))$ kaynak dengeleme (c_k : k kaynak birimi başına maliyet , $r_k(S,t)$: t zamanında verilen S programının k kaynağını kullanımı miktarı)

$\sum c_k \max r_k(S, t)$: kaynak yatırımı

Bu modellerin bazıları aşağıdaki gibi sınıflandırılabilir:

PS|*prec*| C_{max} : Bu model biçimleri kaynak-kısıtlı proje programlama problemleri arasındaki ana problemi oluşturur. Temelde, proje tamamlanma süresini minimize ederken, kaynak kısıtlarını ve önceliklerini gözlemlemek zorunludur. Fakat bu problem hesaba dayalı olması bakımından hala oldukça zorlayıcıdır.

MPS|*prec*| C_{max} : Bu sınıfın modelleri kaynak-kaynak ve zaman-kaynak ödünleşimini ele almaktadır. Dolayısıyla proje yönetiminin gerçekte neye daha çok yakın olduğu gözlemlenebilmektedir.

PS|*temp*| C_{max} : Uygulamaların bir çoğunda aktiviteler arasındaki minimum zaman gecikmelerinin yanı sıra, maksimum başlangıç-başlangıç zamanı gecikmeleri de gözlemlenmelidir. Burada tam olarak geçerli bir problem olan NP tamamlanmıştır.

PS|*temp*| $\sum c_k f(r_k(S, t))$: Bazı uygulamalarda yenilenebilir kaynaklar sınırlıdır ve zamanla kullanılan kaynakların seviyesini dengelemek için bir program bulunmalıdır.

2.2 Tek- modlu durum

Bu bölümde temel kaynak-kısıtlı proje programlama problemi PS|*prec*| C_{max} çözümü için hesaba dayalı ve sezgisel yöntemler özetlenmiştir. $V = \{0, 1, \dots, n, n+1\}$ kümesinden oluşan bir proje varsayılırsa; burada aktiviteler $j=0$, $(j=n+1)$ başlangıç (sınır) aktiviteleri kurgusaldır. Varsayılan bu şebeke döngüsel olmayan ve düğüm üzerindeki bir aktivite tarafından tanımlanır. Aktivitelerin yarıda kesilip daha sonra tekrar devam etme izni yoktur. Burada yenilenebilir kaynaklar sınırlıdır. Varsayılan tüm veriler tamsayıdır. Amaç öncelik ilişkilerinin getirdiği kısıtlamalar ve sınırlı kaynak temin edilebilirliğini karşılayan en kısa süreli iş programını bulmaktır.

En kısa proje süresi izin verilen T üst sınırıyla (zaman pencerelerini türetilir.) için öncelik ilişkileri kullanılarak, yani $[EC_j, LC_j]$ aralığı, erken tamamlanma zamanı EC_j ve geç tamamlanma zamanı LC_j ile ileri ve geri tekrarlanan $j \in V$ aktivitelerinin, önce uygulanan tamamlanma süresi devam eder. Benzer olarak, $[ES_j, LS_j]$ aralığı ES_j , ilk başlangıç zamanı ve LS_j son başlangıç zamanından itibaren sırasıyla aşağıdan ve yukarıdan bağlanır, öncelikle uygulanabilir başlangıç zamanlarını belirtmek için hesaplanabilir. Genelde PS|*prec*| C_{max} 0-1 program aralığı olarak formül oluşturulur değişkenlerden faydalanılır, eğer j aktivitesi t süresinde tamamlanmış ise $x_{jt}=1$ (değilse $x_{jt}=0$ olur).

Dal-kesim yöntemlerinin çeşitli kombinasyonları optimizasyon problemlerinin çözünebilirliğini önemli ölçüde geliştirmiştir. Geçerli eşitsizliklerin üretimi ve bu eşitsizliklerin bir LP çözücü aracılığıyla ilerlemesi devamlı testlere dayalı kısıtların ilerlemesinin etkili bir başlangıcı için düşünülebilir. Programlamada ki başarılı uygulamalar programlama problemlerini ayırmak için (örneğin: iş planlama) ve mevcut durumdaki $PS|prec|C_{max}$ 'ı kapsamaktadır. Bu gibi çabaların amacı çekici olmayan ya da olanaksız düğümlerin sezgisel ve dal-sınır yöntemlerini erken tespit yoluyla hızlandırmaktır. Tsang (1993) tarafından sınırlı yayılıma ayrıntılı bir giriş yapılmıştır.

2.3 Dal-sınır yöntemi

$PS|prec|C_{max}$ için dal-sınır algoritmalarının çeşitliliği Johnson'ın (1967) ilk çalışmasından başlayarak geliştirilmiştir. Kullanılan kısmi işlerin birçoğu, numaralı ağacın düğüm noktaları ilgilidir. Bu dallanma süreci farklı yollarda uzayan kısmi işlerden oluşur. Alt dallardaki baskınlık kuralları ve ani seleksiyon uzayan kısmi işler için alternatiflerin sayısını azaltmaya neden olur. Şemalar ve budama yöntemleri farklı dallanma methodu kullanır. Genelde derin ilk arama bellek ihtiyaçlarını düşük tutmak için kullanılır.

Öncelik Ağacı : Patterson vd. (1989) öncelik ağaç rehberliğinde bir algoritma önermiştir. Bu yöntemde kukla başlangıç aktivitesi 0 ile başlar. Dal-sınır ağacının her bir g seviyesinde mevcutta programlanmış SJ_g ve seçilebilir EJ_g aktivitelerini oluşturur. Yani çoktan planlanmış öncel aktivitelere karar verilmiştir. Sonra uygun bir aktivite olarak J_g seçilir. SJ_g için şebeke ve kaynak kısıtları göz önüne alındığında uygulanabilir başlangıç zamanı bir önceki seviyede belirlenmiş olan başlangıç zamanından daha az olmayacak şekilde hesaplanmıştır. Bu işlemin ardından bir sonraki seviyeye geçilir. En sondaki kukla aktivitesine erişildiğinde tam bir iş programı elde edilir. Bu durumda önceki seviyede gerileme meydana gelir. Sonra denenmemiş uygun bir aktivite seçilir. Tüm uygulanabilir aktiviteler denenirse çözüm elde edilmiş olur ve bir seviye aşağı inilerek diğer derin çözüm gerçekleştirilir. Öncel ağacının kökten yaprağa her dalı aktiviteler arasındaki öncel ilişkilerinin permütasyonudur. Bu algoritma Sprecher tarafından iyileştirilmiştir.

Gecikme Alternatifleri: Bu algoritma Christofides vd. (1987) tarafından gecikme alternatiflerinin işleyişine dayalı olarak Demeulemeester ve Herroelen (1992) tarafından geliştirilmiştir. Daha önceki ağaç algoritmalarının aksine dal-sınır ağacının her bir g seviyesi sabit bir zaman anı ile ilgilidir t_g 'de (karar noktası) aktiviteler başlatılabilir. Sonuç olarak uygun aktivitelerin farklı bir tanımı bu algortmada kullanılır. Henüz

programlanmamış bir aktivite j olarak ifade edilir. Uygun bir t_g süresinde, i öncellerinin tamamı bir $C_i \leq t_g$ tamamlanma süresi ile planlanır. Ayrıca bir j aktivitesi S_j başlangıç zamanıyla $S_j \leq t_g < S_j + p_j$ eşitsizliğine sahipse t_g süresinde olduğu ifade edilebilir. Dal-sınır ağacının mevcut g seviyesindeki işlemi aşağıdaki gibidir. Bu yeni karar noktası t_g , t_{g-1} sürecinde aktivitelerin en erken tamamlanma süresi olarak tanımlanır. Yenilenebilir kaynakların sabit ve uygun seviyelerinden dolayı yalnızca bitiş süreleri planlanmış aktivitelerin planlanmamış diğer aktivitelerden başlayarak dikkate alınması gerekir. Kullanılan aktivitelerin FJ_g dizisi karar noktasından önce veya zamanında bitirilir uygun aktivitelerin EJ_g dizisi hesaplanır. Uygun aktivitelerin tamamı başlatıldığında bu aktivitelere JIP_g dizisi eklenmesi sırasında bir kaynak çakışmasına neden olmuş olabilir. Bu yüzden sonraki tanıma göre minimal gecikme alternatifler dizisi hesaplanır.

A gecikme alternatifi DA_g, JIP_g 'nin bir altkümesidir. Öyle ki her bir yenilenebilir $k \in R$ kaynağı için $\sum_{j \in JIP_g} \frac{r_{jk}}{DA_g} \leq R_k$ 'dir. DA_g 'nin alt kümesi uygun değil ise A gecikme alternatifi DA_g minimal olarak adlandırılır, bir gecikme alternatifidir. A minimum gecikme alternatifi seçilir ve mevcut kısmi programlamadan gecikmeli aktiviteler kaldırılır. Eğer kaynak çakışması meydana gelmiyorsa, sadece en küçük gecikme alternatifi grupta boş bırakılır. Bir j aktivitesinin başlangıç zamanı geciktirilebilir çünkü bu bilgi gerileme sırasında düzeltilebilir. Daha sonra bir sonraki seviye dallandırılır ve sonraki karar noktası hesaplanır. Eğer bu programlama şu an tamamlanırsa gerileme de uygulanır ve sonraki minimum gecikme alternatifleri test edilir. Bu uygulama önceki ağaç algoritmalarından farklıdır çünkü dal-sınır ağacının (tek) yerine aktiviteler kümesine her bir seviyede başlanır. Ayrıca burada aktiviteler seçilmeden önce hangi aktivitenin başlatılabileceği anında belirlenebilir. Son olarak öncelik ağaç algoritmalarının aksine bu yaklaşım, daha düşük bir seviyede olan, bu mevcut seviyede planlanan kararların geri çekilmesine izin vermektedir.

Genişleme Alternatifleri: Stinson vd. (1978) kısmi iş programı oluşturmak için geniş alternatiflerin kullanımını önermiştir. Önceki algoritmada olduğu gibi dal-sınır ağacının her bir g seviyesi bir t_g karar noktasıyla ilişkilidir. Süreç içerisindeki aktiviteler JIP_g , bitmiş aktiviteler FJ_g ve uygun aktiviteler EJ_g dizisidir. Daha sonra mevcut kısmi programlama, kaynak kısıtlarını ihlal etmeden karar noktasında, uygun aktivitelerin bir alt kümesinden başlayarak genişletilir. Daha doğru bir ifadeyle, geniş bir alternatif EAg bir

alt kümedir. Her bir $k \in R$ kaynağı için $\sum_{j \in JIP_g} \frac{r_{jk}}{DA_g} \leq R_k$ geçerlidir ve ayrıca eğer $JIP_g \neq \emptyset$ ise $EA_g \neq \emptyset$ dir. Algoritmayı sonlandırmayı garanti etmek için, eğer süreç içerisinde aktivite yoksa, sadece genişleme alternatifleri bulunabilir. Fakat, süreç içerisinde aktiviteler varsa (mevcut ise), optimum sonucun bulunduğunu garantilemek için boş kümenin her zaman genişleme alternatifleri olduğu test edilmelidir. Dal-sınır ağacının mevcut g seviyesinde ki bu yöntem aşağıdaki gibidir: Uygun ve geniş aktiviteler dizisinin yeni karar ve hesap noktasının belirlenmesidir. Son olarak, geniş bir alternatif olan EA_g ' nin seçilmesi ve uygun aktivitelerin başlaması dallanmanın bir sonraki seviyesinden öncedir. Bu gerileme mekanizması önceki algoritmanın birine denktir. Bu yöntem daha önceki algoritmalarından farklıdır: Daha öncekiler, mevcut seviyeden daha düşük bir seviyede başlatılıp aktivitelerin gecikmesi ihtimal içerirken, bu algoritma en alt seviyenin planlanmasında gecikmelere izin vermemektedir. Sonuç olarak, bu arama maksimal geniş alternatifleri sınırlamayabilir, sadece minimal gecikme alternatifleri göz önüne alındığında eniyilemeyi kaybetmez.

Blok Uzantıları: Mingozi vd. (1998) aşağıdaki fikirlere göre nispeten farklı bir yaklaşımda bulunmuşlardır. Tanımlı zamanlarda optimal bir program vardır.

$t_0 = 0 < t_1 < t_2 \dots < t_l$ ve uygun aktiviteler dizisi $A_1, A_2, \dots, A_l$ öyle ki;

I. Her bir t_i ($i > 0$) birkaç aktivitenin bitiş süresidir.

II. A_i aktivitelerinin tümü birlikte $[t_{i-1}, t_i]$ ($i=1, \dots, l$) sırasında ilerleyebilir.

III. Eğer bir aktivitesi $J \in A_i$, $[t_{i-1}, t_i]$ süresi içerisinde tamamlanmazsa, $[t_i, t_{i+1}]$ süresi içerisinde işlenebilecektir.

IV. Herhangi bir aktivitenin tüm öncellerinin hangi t_i süresinde başlayacağı t_i süresinden önce planlanır.

Birlikte işlenebilecek aktiviteler dizisi A_i ile bir blok böyle bir aralıktan oluşur $[t_{i-1}, t_i]$. Ayrıca kısmi bir programlama, III. ve IV. uygun şartlardaki bir dizi blokların yardımıyla tanımlanır. Daha sonra tekrar kısmi planlara yeni bloklar eklenerek dallandırılır.

Planlama şemaları: Dal-sınır algoritması mesleki ve çoklu işlem iş programlama problemlerinde kullanımı yaygınlaştırılmıştır. Ayrıca Bartusch vd. (1988) bu yöntemi uygulamıştır. Kısmi programlamaların yerine uygun programların bir grubu plan şemaları ile gösterilebilir. Plan şemaları aşağıdaki gibi ifade edilebilir.

Bir S programında rastgele iki aktivite ya paralel bir şekilde $i // j$ ya da ikisinden biri diğerine bağlı olacak şekilde gösterilir $i \rightarrow j$ veya $j \rightarrow i$. Ancak j başlangıç süresinden önce biterse eğer, o zaman $i \rightarrow j$ şeklinde kabul edilir. $i // j$; i ve j en az ortak bir zaman birimi içinde paraleldir anlamına gelmektedir. Bu ilişkiler genişletilerek zaman çizelgeleri oluşturulur. $i \rightarrow j$ ya da $j \rightarrow i$; $i - j$ ayrımı ile genişletilir. $i - j$; ya $i \rightarrow j$ veya $j \rightarrow i$ anlamına gelir. Ayrıca $i - j$ ayrımı ve $i // j$ paralelliği $i \sim j$ bağıntısını esnetmek için genişletilebilir (gevşetilebilir?). $i \sim j$ henüz kararsız olduğu anlamına gelir, $i - j$ veya $i // j$ iki bağıntı arasındaki herhangi ikisinden biri düzenlenir. Bir dizi ortaklıklar, ayrıklıklar, paralellik ilişkileri ve esnek ilişkiler sırasıyla C , D , N VE U ile gösterilir. C , D , N , U herhangi iki farklı aktivite için bir planlama şemasıdır i, j tam olarak aşağıdaki ilişkilerden biridir: $i \rightarrow j \in C$, $j \rightarrow i \in C$, $i - j \in D$, $i // j \in N$ veya $i \sim j \in U$. Bir dizi uygulanabilir programın (herhangi biri boş olabilir) programlama şeması (C, D, N, U) şeklinde tanımlanır, yani uygulanabilir programların ve aralarındaki ilişkilerin tamamı C , D , N ile gösterilir.

Eğer verilen tüm öncel bağıntıların kümesi C_0 ise, aktivitedeki çiftlerin tamamı D_0 'dır, kaynak kısıtlarından dolayı paralel bir süreç olamaz ve geride kalan çiftlerin tümü $i \sim j$, U_0 'dır, sonra $(C_0, D_0, \emptyset, U_0)$ uygulanabilir tüm programların kümesini temsil eder. $(C_0, D_0, \emptyset, U_0)$ ağaç numaralarının köküne karşılık gelir.

Bir programlama şeması için $(C_0, D_0, \emptyset, U_0)$ tüm bağıntılarda uygulanamayan bir programın olmadığı gösterilebilir veya baskın bir şekilde uygulanabilir bir program hesaplanabilir (cf.[117]). Her ikisi de $O(n^3)$ süresinde hesaplanabilir. Böylece programlama şemalarının bu formu $(C, D, N, \emptyset)$ numaralandırma ağacının yaprakları olarak ele alınabilir ve serbest bir bağıntı olan $i \sim j$; $i - j$ ve da $i // j$ biri ile yer değiştirerek dallanabilir.

Programlama şemasında bağıntıların dahil edilmesi, ağaç numaralarının her bir düğümünde geçici bir analiz oluşturmasına izin verir. Bu analizle yeni bağlantılar, ayrılmalar veya paralellik ilişkiler ortaya çıkarılır. Ayrıca zaman penceresindeki aktiviteler için en düşük sınır hesaplar geliştirilebilir.

En küçük engellenmiş gruplar : Igelmund ve Radermacher(1983) engellenmiş gruplara bağlı olarak bir dallanma şeması tanıttı. Ayrıca en alt sınırlar, kısmi numaralı algoritmalarda arama ağacının en geniş kısımlarını budamak için; Demeulemeester, Herroelen (1992, 1997) ve Sprecher (1996) baskın kuralları başarılı bir şekilde

kullanmışlardır. En güçlü baskın kuralların arasında kesi kümesi kuralı vardır ve kural daha önce değerlendirilmiş kısmi programların depolanmış bilgilerinden yararlanır. Arama sürecinde iken bu kural, mevcut kısmi programlama ile depolanmış bilgiyi kıyaslar. Eğer mevcut kısmi programlamadan kanıtlanabilir herhangi bir çözüm elde edilirse, daha önce değerlendirilmiş kısmi bir programlamadan elde edilen bir çözümden daha iyi olamaz, bu depolanmış bilgi gerilemede uygulanabilir.

2.4 Alt sınırlar

Genelde alt sınırlar $PS|prec|C_{max}$ 'ın optimal çözüm değeri için (gevşetilmiş problemin çözümü) için kısıtların bazıları gevşetilerek serbest hesaplanabilir. Kaynak kısıtlarının serbestliği kritik yol süresinin elde edilmesine yol açar, basit bir alt sınır sağlar. Stinson vd. (1978) CP 'de kritik yola bağlı olmayan bir i aktivitesi ekleyerek bu sınırı geliştirmişlerdir. i aktivitesinin (CP ile paralel işlenebilir) zaman kümelerinin süresinin maksimum değerinin hesaplamışlardır. Kritik yol uzunluğuna (süresine) maksimum $\{0, p_i - e_i\}$ ilave edilerek yeni bir alt sınır elde edilebilir. Tek bir aktivite ilave etmenin yerine, Demeulemeester ve Herroelen CP ile ayrı bir P düğüm noktası aracılığıyla CP için [aktif bir programlama yöntemi olan P'yi kullanarak] bir alt sınır hesaplanır ve kritik yolun uzatılabilir olduğunu ileri sürmüştür. Genelde bu alt sınır CP boyunca optimal çözüm değeri ile uyumlu değildir. Fakat iki yollu gevşetme, ikili görevlerin iş problemleri için gelişmiş bir grafiksel yöntem ile optimal bir çözüm olabilir. Bu yaklaşım ayrıca CP için optimal bir çözüme olanak sağlar [CP 'de aktiviteler P zaman pencereleri ile uyumludur. Mingozzi vd. (1998) aşağıdaki lineer programa bağlı bir sınırı kısmi olarak, öncel kısıtlara ve ayırmalara izin verir. Yapabilecekleri paralel olarak işlenecek maksimum aktivite kümesini göz önüne alırlar. Kümelerin tamamında karakteristik vektörler $a_1, a_2, \dots, a_q$ oluşturulur. Sonra lineer programlamanın genişletilmiş bu formunda

$$\min \sum_{j=1}^q x_j \quad (1)$$

$$\sum_{j=1}^q a_{ij} x_j \geq p_i, i = 1, \dots, n, \quad (2)$$

$$x_j \geq 0, j = 1, \dots, q, \quad (3)$$

x_i zaman birimlerinin sayısını belirtir, a_j tarafından temsil edilen tüm faaliyetler ortak olarak işlenir. Eşitlik (1-3) tamsayı versiyonu bir dizi paketleme problemi oluşturur. Mingozzi vd. (1998) kaynak-kısıtlı proje programlama problemi için paketleme problem kümesini sezgisel bir şekilde çözerek alt sınırları oluşturmuşlardır. Baar vd. (1997) lineer programa doğrudan sütun oluşturma tekniklerini uygulayarak çözmüşlerdir. Sütun ve

aktivitelerin sayısı ile katlanarak büyümesine rağmen bu yöntem oldukça hızlıdır. Brucker ve Knust (1999) aktiviteler için zaman pencerelerini hesaba katarak bu yaklaşımı geliştirmişlerdir. Bu zaman pencereleri, tamamlanma zamanı için hayali bir T üst sınır kullanarak öncel kısıtlardan türetilir. Mevcut sütunlar aktiviteler kümesine uyum sağlar ve verilen bir zaman penceresinde birlikte işlenebilir. Burada amaç tüm zaman pencereleri ile ilgili programlamada önlem almaktır. Eğer bu şekilde bir program mevcut değil ise, T bir alt sınırdır. T 'nin en büyük özelliği ikili arama sağlamasıdır. Bu "yıkıcı iyileştirme" tekniği ayrıca Klein ve Scholl (1997) tarafından kullanılmış olup, uygun olmayan bir T üst sınır sağlaması için çeşitli yöntemlerle test edilmiştir. Sınırlara dayalı diğer lineer programlama dal- sınır algoritması ile ilişkili olarak Christofides vd. (1987) tarafından tanıtılmıştır.

İlk sezgisel yöntemler programlama yöntemlerine dayalı öncelik kurallarıdır. Şu ana kadar çoklu öncelik kuralları deneysel olarak test edilmiş ve önerilmiştir. Önceliğe dayalı sezgisel tarama; sezgisel olmanın, kolaylıkla uygulanabilmenin ve hesap bakımından hızlı olmanın avantajına sahiptir. Fakat önceliğe dayalı sezgisel tarama, ortalama en iyi amaç fonksiyon değerinden uzaklaşması bakımından çok başarılı değildir. Dolayısıyla son zamanlarda yapılan çalışmalar kesik dal-sınırı, sezgisel programlamaya dayalı hesaplar, ayrık ark kavramı, lokal kısıtlara dayalı analizler,örnekleme teknikleri gibi sezgileri daha detaylı incelemek için, araştırmacılar ilgilerini bu yöne çevirmişlerdir.

Demeulemeester ve Herroelen (1992), Patterson(1984) tarafından oluşturulan problem kümesinde dal-sınır algoritmasında 51 aktiviteye kadar 110 örnek problemten oluşan problemleri çözmüştür. Demeulemeester ve Herroelen (1992), (IBM PS/2 Model 70 A21, 25 MHz), bilgisayarda 0.21 s ortalama bir zaman hesabı ile problemi çözmüşlerdir, Stinson vd. ise bu algoritmayı yaklaşık 12 kat daha hızlı çözerek daha üstün gelmişlerdir.

Kolisch vd. (1995), etkin proje bir jeneratörü ProGen adında bir parametre geliştirmişlerdir. Daha sonra bu parametre, kaynak-kısıtlı proje programlama için önerilen algoritmaların daha iyi değerlendirilmesi için bir araç olarak, oldukça yaygın bir şekilde kullanılmıştır. Aynı zamanda bu parametre 30, 60, 90 ve 120 aktiviteli testlerden ve her biri 480 tür örnekten oluşur. Bu 30 aktiviteden oluşan test, Demeulemeester ve Herroelen (1992) tarafından bir kişisel bilgisayarda kullanılmıştır (IBM PS/2 Model 55SX, 386SX, 15 MHz). Patterson-kümesi ortalama 1.06 saniyede çözümlenirken, 480 problemler örneğinin sadece 415'i , problem başına 1000 saniye çözüm süresi sınırı çözülebilmektedir.

Mingozi vd. (1998), lineer programlama formülasyonuna dayalı sınırlar ile Stinson vd. (1978) tarafından tanıtılan kritik dizi sınırından ve Demeulemeester ve Herroelen (1992) tarafından sunulan algoritmadan daha kompetiftir ve sonraki uygulamalara kadar en iyilerinden biri olarak bilinmektedir. Demeulemeester ve Herroelen (1997), alt sınırına uyarak ve işaretlenmemiş olan tamsayılardan biri 32 bit ile 8 bit ebatında dört kaynağı göstererek bu yaklaşımlarını geliştirmişlerdir. İlk kez 24 Mbyte 'a kadar izin veren 30 aktivitenin tamamı çözülebildi. Kişisel bir bilgisayarda (80486, 25 MHz) ana işlemci süresi yaklaşık 34 saniyedir. Sprecher kendi dal-sınır algoritması ile Demeulemeester ve Herroelen 'in hesaplama zamanı ve hafıza gereksinimleri arasındaki ödünleşimi belirten algoritmanın geliştirilmiş biçimini kıyaslamıştır. Demeulemeester ile Herroelen (1997), geliştirmiş oldukları algoritma ile, 30 aktiviteli bir problemin 480 kriterinin 479'unu 24 Mbyte hafıza kullanarak ortalama 12.33 saniyede çözerlerken , Sprecher'in algoritması ise aynı problem üzerinde 400 Kbyte hafıza kullanarak ortalama 12.35 saniyede çözülmüştür.

Brucker vd. (1998), dal- sınır algoritması ile diğer algoritmalarından daha az problem çözmüşlerdir fakat dal-sınır algoritmasının asıl avantajı hafıza gereksiniminin diğerlerinden çok daha az olmasıdır. 90 aktiviteli problemler için en fazla 10 Mbyte bellek kullanılır.

Araujo vd. (2019) önleyici olmayan KKPPP'yi ve türevleri için otomatik karışık tamsayı doğrusal programlama (KTDP) formülasyon stratejileri önermiştir. En karmaşıktan en genel duruma kadar KKPPP'nin problem varyantları: tek modlu kaynak kısıtlı proje programlama problemi (TMKKPPP); çok modlu kaynak kısıtlı proje programlama problemi (ÇMKKPPP); çok modlu kaynak kısıtlı çoklu proje programlama problemi ele alınmıştır.

Her yinelemede güçlendirilmiş Chvátal-Gomory kesimlerini ayırmak için fizibilite ve optimallik koşulları dikkate alınarak yoğun bir çakışma grafiği oluşturulmuştur ve Chvátal-Gomory kesimleri olmak üzere 5 kesim ailesi içeren paralel bir kesme düzlemi algoritması geliştirilmiştir. Her yinelemede fizibilite ve optimallik koşullarını göz önünde bulunduran yoğun bir çatışma grafiği oluşturulmuş ve bu kesinti üreticileri tarafından güçlendirme prosedürlerinde kullanılmıştır. Tüm kesimler, özellikle kesme düzleminde bir arada olduklarında alt sınırları iyileştirmeye katkıda bulunmuştur. Öncelikli kesintiler tüm varyantlar için en etkili olarak belirlenmiştir. Güçlendirilmiş Chvátal-Gomory kesintileri, bir grup çoklu proje örneğinde özellikle etkili olmuştur. Bu sonuçlar, kesim

jeneratörlerinin bir durum özelliğine dayalı ayarlamasının faydalı olabileceğini göstermektedir.

Araujo vd. (2019), 5 farklı kesim ailesini ayırmak için bir kesme düzlemi algoritmasının yanı sıra kaynakla ilgili kısıtlamaları güçlendirmek için yeni bir ön işleme rutini önerilmiştir. Önerilen stratejiler, doğrusal gevşeme sınırlarını önemli ölçüde geliştirerek, son teknoloji ürünü bir karma tamsayı doğrusal programlama çözücüsünün, KKPPP'lerin farklı varyantlarının önceden açık 754 örneği için kanıtlanabilir en uygun çözümleri bulmasına olanak tanımıştır.

Yöntemlerle üretilen geliştirilmiş doğrusal programlama formülasyonları ile PSPLIB'den (Kolisch and Sprecher, 1996). 247, MISTA Challenge'dan (Wauters vd. 2016) 1 ve MMLIB'den (Van Peteghem ve Vanhoucke, 2014) 506 örnek olmak üzere toplam 754 örnek çözülmüştür. Scientific Linux ANSI C 99 sürüm 6,3 yazılımında 2,67 GHz ve 512 GB RAM'e sahip Xeon X5650 Westmere işlemcilerle çözümler elde edilmiştir. Tüm algoritmalar GCC 5.4.0 sürümünde kodlanmış ve GUROBI sürüm 8.0.1 (Gurobi Optimization, 2016) ile derlenmiştir.

Ön işleme ve kesme düzlemleri rutinleri dâhil olmak üzere KKPPP için geliştirilmiş dal ve düğüm seçim stratejileri ile eksiksiz bir dal-sınır algoritmasının geliştirilmesi de gelecek vaat eden bir yol olarak belirlenmiştir. KKPPP için kompakt formülasyonların doğrusal programlama gevşemesini iyileştirmek için yeni karma tamsayı doğrusal programlama tabanlı yöntemler önerilmiştir.

Banihashemi vd. (2020) KKPPP için Veri Zarflama Analizi (VZA) yaklaşımı ile zaman-maliyet-kalite-çevresel etki ödünleşimini ele almıştır. Çalışmanın amacı, zaman-maliyet-kalite ve çevresel etkilerin ödünleşim probleminin optimizasyonu için farklı uygulama modlarında proje faaliyetlerinin verimliliğini değerlendirmek olarak belirtilmiştir.

En iyi yürütme modunu seçmek ve hedefler arasında bir denge bulmak için farklı yürütme modlarına sahip proje faaliyetlerinin değerlendirilmesi için paralel bir VZA yöntemi sunmuştur. Proje aktivitelerinin niteliğine göre, çıktılar istenen (kalite) ve istenmeyen (zaman, maliyet ve çevresel etkiler) olarak kategorize edilmiştir ve VZA modeline göre analiz edilmiştir. Verimli yürütme modlarını sıralamak için ideal ve ideal olmayan sanal birimler yöntemi kullanılmıştır. Önerilen model, geçerliliğini kanıtlamak için gerçek bir kırsal su temini inşaatı projesinde uygulanmıştır.

Bulgular, her bir aktivitede verimli yürütme modunun kullanılmasının dört proje hedefi (zaman, maliyet, kalite ve çevresel etkiler) arasında optimal bir ödünleşime yol açtığını göstermiştir. Çalışma, proje yöneticilerine ve uygulayıcılara, zaman-maliyet-kalite-çevresel etkileri hesaba katarak proje faaliyetlerinin en verimli yürütme modlarını seçmelerinde yardımcı olabilecektir.

Bold ve Goerigk (2021) belirsiz aktivite süreleri ile kaynak kısıtlı proje programlama problemini (KKPPP) ele almaktadır. Aktivite sürelerinin bütçelenmiş bir belirsizlik kümesinde olduğu varsayılmıştır ve bir karar vericinin problem belirsizliğine bağlı olarak kaynak çatışmalarını çözmesi gerektiği, ancak belirsiz aktivite süreleri öğrenildikten sonra aktivite başlangıç zamanlarını belirleyebildiği sağlam iki aşamalı bir yaklaşım izlenmiştir. İki aşamalı ayarlanabilir sağlam optimizasyon probleminin kompakt bir muadilinin türetilmesini sağlayan ikinci aşama problem için yeni bir formülasyon sunulmuştur.

Belirsiz KKPPP'yi çözmek için iki ana yaklaşım benimsenmiştir. Birincisi, problemi, bir programlama politikasına göre her yeni bilgi elde edildiğinde programlama kararlarının verildiği dinamik bir optimizasyon problemi olarak görmektir (Igelmund ve Radermacher, 1983; Möhring ve Stork, 2000). Li ve Womer (2015), belirsiz KKPPP için uyarlanabilir bir kapalı döngü zamanlama politikası bulmak için yaklaşık dinamik programlamayı kullanmaktadır. İkinci yaklaşım, proaktif olarak aktivite sürelerindeki gecikmelere karşı koruma sağlayan sağlam bir temel hat programlaması geliştirmeyi amaçlamaktadır.

Belirtilen yeni kompakt formülasyon, kaynak çatışması olmayan bir proje için, en kötü durum gecikmeli üretim aralığını en üst düzeye çıkarmanın ikinci aşama karşıt alt probleminin yeniden formüle edilmesi dikkate alınarak elde edilmiştir. Bu alt problemin yeniden formüle edilmesi, orijinal proje şebekesinin birden çok kopyasından yapılan artırılmış bir proje şebekesi üzerindeki en uzun yol probleminin eşdeğeri olarak tanımlanmıştır.

Çalışmada bu yeni formülasyonun performansı, farklı özelliklere ve zorluklara sahip 1440'ın üzerinde örnek üzerinde incelenmiştir. Sonuçlar, önerilen formülasyonun, standart optimizasyon yazılımıyla, bu sorunu çözmek için mevcut en iyi algoritmadan önemli ölçüde daha hızlı çözülebileceğini göstermiştir. Çalışmada önerilen bu yaklaşım kullanılarak, aynı zaman sınırı içinde yaklaşık %50 daha fazla örneğin optimal olarak

çözülebildiği ve daha küçük bir ortalama boşluk ve çözüm süresinin elde edilebildiği belirtilmiştir. Çalışmada kullanılan test örnekleri, PSPLIB'den alınan 30 aktivite içeren deterministik KKPPP örneklerinden oluşmaktadır. (Kolisch ve Sprecher, 1997). Önerilen tüm modeller, 16 GB RAM ile sınırlı 2.30 GHz Intel Xeon CPU'nun 4 çekirdeği üzerinde çalışan Gurobi 9.0.1 kullanılarak çözülmüştür.

Yapılan deneyler, bu kompakt formülasyonun, standart optimizasyon yazılımı kullanılarak, bu problemi çözmek için mevcut son teknoloji algorithmadan önemli ölçüde daha hızlı çözülebileceğini ve aynı kıyaslama setinde yaklaşık %50 daha fazla örnek için optimum sonuca ulaşılabilirliğini göstermektedir.

Chakraborty vd. (2019) değiştirilmiş değişken komşu arama sezgisel algoritması (DDKASA) kullanılarak ÇMKKPPP ele alınmıştır. Çalışmanın amacı, belirleyici yenilenebilir ve yenilenemez kaynak gereksinimleri olduğu bilinen aktivitelere sahip projeler için ÇMKKPPP'ne hızlı ve optimuma yakın bir çözüm sağlamaktır.

Zamanlama problemlerine uyan gelişmiş bir optimizasyon tekniği olarak DDKASA kullanılmıştır. Deneysel çalışma için, PSPLIB'den 10-30 aktivite içeren bir dizi projeye sahip standart bir 3929 çok modlu karşılaştırma örneği kümesi kullanılmıştır. Daha iyi bir karşılaştırma için aynı zamanda MMLIB50, MMLIB100 ve veri kümelerinden yeni geliştirilmiş 4320 çoklu MMLIB + modu örneklerinden oluşan standart bir küme de kullanılmıştır. 50.000 deneme sınırıyla bu veri kümelerinde önerilen algoritma 106,34 için daha iyi bir çalışma süresi sağlamıştır ve 1601 örnek problemin çözümünde, özellikle daha fazla sayıda aktiviteye sahip projeler için önerilen algoritmanın verimli olduğunu göstermiştir. Problemlerin çözümü, 16 GB RAM ve 3.40 GHz CPU'ya sahip bir Intel core i7 işlemci üzerinde gerçekleştirilmiştir. Tüm yöntemler Matlab, R2015b'de kodlanmış ve çözülmüştür. PSPLIB'den (Kolisch ve Sprecher, 1997), 10, 12, 14, 16, 18, 20 ve 30 kuklasız aktivite içeren örnekler dahil olmak üzere bir dizi kıyaslama problemi seçilmiştir.

Çalışmanın katkısı, hem yenilenebilir hem de yenilenemez kaynaklara sahip olma seçeneği ile ÇMKKPPP için güncellenmiş bir paralel programlama oluşturma planı önermek olmuştur. Gerektiğinde çok sayıda aktif aktivite arasında doğru aktiviteyi belirlemek için dinamik bir mod seçim politikası geliştirmiştir. Optimal veya optimuma yakın iş çizelgesi oluşturmak için, çalışma aynı zamanda başlangıçta tamponlanmamış bir başlangıç programını temsil eden aktiviteler arasında komşu dağıtılması için değiştirilmiş bir yaklaşım önermiştir. En uygun veya optimuma yakın sonuçları bulmak için gelişmiş

değiştirme ve seçme modu ile bu ilk arabelleğe alınmamış programların nasıl keşfedileceğini göstermiştir.

Gelecekteki araştırmalarda (i) ÇMKKPPP için daha büyük MMLIB veri setini çözmeyi, (ii) süre ve kaynak kesintileri altında ÇMKKPPP'leri için buluşsal ve meta-sezgisel yaklaşımlar geliştirmeyi ve (iii) ÇMKKPPP'leri diğer türler altında çözmek için belirsizliğe toleranslı yaklaşımların önerisini içereceği belirtilmiştir.

Coelho ve Vanhoucke (2020) kaynak kısıtlı proje programlama problemini (KKPPP) ele alınmıştır. Çalışmanın amacı, bir KKPPP örneğini neyin zor kıldığına dair iç görü ve anlayış oluşturmak ve literatürde hâlihazırda mevcut olan algoritmalarla çözülmesi imkânsız olan küçük bir örnek kümesinden oluşan yeni bir veri kümesi önermek olmuştur.

Problemin zorluğunu, çok sayıda mevcut örnekte farklı algoritmaları test ederek ve hangilerinin çözülmesinin zor olduğunu tespit ederek analiz edilmiştir. Bu amaçla, kolay KKPPP örneklerini çok zor olanlara dönüştürmek için 5 temel blok, iyi performans gösteren algoritmalar ve büyük miktarda hesaplama gücü kullanan üç aşamalı bir yaklaşım önerilmiştir.

İlk aşamada, algoritma yeni bir zorluk göstergesini kullanmıştır ve zorluklarını artırmak için mevcut KKPPP örneğinde yinelemeli olarak küçük değişiklikler yapmıştır. Bu çekirdek yöntemi, temelde farklı iki dizide uygulanmıştır ve bunların her biri, çok çeşitli KKPPP örnekleri üzerinde test edilmiştir.

Çalışmada toplamda yaklaşık olarak 45 yıllık hesaplama süresi (tek çekirdekli işlemci kullanıldığında) alan dört hesaplama deneyi kurulmuştur ve bu deneyler sonunda tümü en fazla 30 aktiviteye sahip olan 623 örnekten oluşan küçük bir setle sonuçlanmıştır.

Bu araştırmanın, gelecekteki araştırmaları farklı şekillerde teşvik edebileceğine inanılmaktadır ve yeni algoritmalar geliştirirken radikal atılımlara yol açacağı öngörülmektedir. Makalede önerilen üç aşamalı yaklaşım, 623 örneğin nispeten büyük bir kümesini elde etmek için en iyi performans gösteren yaklaşım olduğu belirtilmiştir. Bu örneklerin literatürde bulunan diğer örneklerden daha zor olduğunu bulmak için gelecekteki araştırmaların derinlemesine bir analize odaklanması gerektiğine inanılmaktadır.

KKPPP için algoritmaları test etmek ve doğrulamak için bir araç Vanhoucke ve Coelho'da (2018) sunulmuştur. Çözüm Güncellemesi adlı bu araç literatürde bilinen alt sınırlar ve en iyi bilinen üst sınırlar için mevcut tüm veri kümelerini hesaplamaktadır ve

araştırmacının keşfetmeye değer örneklerinin filtrelemesini sağlamaktadır. Gelecekte araştırma çalışmaları yapılırken bu aracın denenmesi tavsiye edilmektedir. www.projectmanagement.ugent adresinden indirilebilmektedir.

Kadri ve Boctor (2017) sıraya bağlı aktarım süreleri ile kaynak kısıtlı proje programlama problemini (SBASKKPPP) iki noktalı çaprazlama operatörü kullanan yeni bir genetik algoritma ile çözümü ele alınmıştır. Yeni geliştirilen algoritmanın performansını daha önce literatürde yayımlanan çalışmaların performansı ile karşılaştırarak elde edilen iyileşme karşılaştırılmıştır.

Önerilen GA, 2014 MATLAB sürümünde uygulanmıştır ve Intel çekirdek i7 (2.0 gigahertz) işlemci ve 6 gigabayt RAM içeren bir kişisel bilgisayarda test edilmiştir. Çalışmada ki test örnekleri, proje planlama örnekleri kitaplığındaki (PSPLIB) örneklerle dayanmaktadır. Çalışmada GA, Krüger ve Scholl (2009) tarafından çok geçişli buluşsal yöntem ile Poppenborg ve Knust (2014) tarafından sunulan tabu arama algoritması (TAA) ile çözülmüştür. Yapılan analizlerin sonucunda 5000 çözümlü GA'nın ortalama % 2,18 oranında iyileşme sağladığını ve bu iyileşmenin aktivite sayısındaki artışla arttığını göstermektedir. Bu nedenle, ortalama yüzde iyileşme 30 aktiviteye sahip örnekler için %0,37 iken, 120 aktiviteli örnekler için %3,62'ye yükselmiştir. Önerilen GA'nın test edilen 408 aktivitenin 220'si (% 54) için daha iyi bir çözüm, 361 aktivitenin (% 88,5) için daha iyi veya eşit bir çözüm ve 47'si (% 11,5) örnek için daha kötü bir çözüm sağladığı sonucuna varılmıştır.

Sunulan yeni GA, literatürde yayımlanan diğer yöntemlerden daha iyi performans gösterdiğini ortaya koymuştur. SBASKKPPP'yi çözmek için daha verimli optimal ve sezgisel yöntemler geliştirmek için daha fazla araştırmaya ihtiyaç olduğunu göstermiştir. Daha büyük ve daha zor problemleri çözme kapasitesine sahip bir optimal çözüm yönteminin geliştirilmesi, önerilen GA algoritmasının ve diğer sezgisel yöntemlerin performansının daha eksiksiz bir şekilde değerlendirilmesine izin verileceğini ortaya koymuştur.

Hauder vd. (2020) aktivite ve zaman esnekliği ile KKPPP'ni ele alınmıştır. Üretim süresinin en aza indirilmesinin yanı sıra, iki yeni alternatif hedef sunulmuştur: seçilen aktivitelerin dengeli uzunluğunun maksimize edilmesi (zaman dengesi) ve dengeli kaynak kullanımının sağlanması olarak tanımlanmaktadır.

Geliştirilen entegre esnek proje programlama problemi için yeni karma tamsayı (MIP) ve kısıt programlama (CP) modelleri önerilmektedir. MIP ve CP modelleri OPL'de uygulanmıştır ve bunları çözmek için CPLEX 12.9.0 MIP çözücü ve CP Optimizer kullanılmıştır. Tüm deneyler, 28 mantıksal işlemci, Microsoft Windows 10 Education ile Intel (R) Xeon (R) CPU E5-2660 v4, 2.00 GHz sanal makine üzerinde gerçekleştirilmiştir.

Hauder vd. (2020) çoklu proje (RCPSP-AC) için, proje örnekleri Tao ve Dong (2017) 'da verilen projeleri kullanmıştır. Toplamda $j = 30$ düğümlü bir proje yapısından ve 5 iç içe veya düğümden oluşan bir örnek sunulmuştur. Tao ve Dong'un (2017) ardından, her biri 15 örnekten oluşan 5 örnekle grup elde etmek için, bu proje yapısı 2, 3, 4 ve 5 ile çarpılarak 30, 60, 90, 120 ve 150 düğümlü örnek gruplar elde edilmiştir.

Gelecekteki araştırmalarda, bir zamanlama bağlamında esnekliğin ilerlemesiyle ilgili çeşitli konuların ele alınması amaçlanmaktadır. RCPSP-AC ve kaynak dengesini maksimize eden RCMPSP-ACTF için daha iyi çözümler sağlayan yeni yöntemlerin ele alınması önerilmektedir. Kaynakların aniden kesilmesi gibi yıkıcı olaylarla ilgili olarak elde edilen optimizasyon sonuçlarının sağlamlığı, kuruluşların rekabet gücünü önemli ölçüde etkileyebilmektedir. Bu zorluğun üstesinden gelmek için, bozulma değişkenleri ve sağlam çözümler elde etmek için ilgili yöntemlerin araştırılması önerilmektedir.

Issa ve Tu (2019) , A, B, C ve D kategorileri olarak kaydedilen dört proje aktivitesi göz önünde bulundurularak KKPPP ve ÇMKKPPP' nin çalışmalarını ele almıştır. A kategorisi, X eksenini boyunca sabit süre, Y eksenini boyunca sabit kaynaklar kullanılarak gerçekleştirilebilen aktivitelere atıfta bulunmaktadır ve kesintiye uğratılmaması gerekmektedir. B kategorisi, A kategorisinin aynı kaynak türü kullanılarak gerçekleştirilebilen ancak kesintiye uğratılabilen aktiviteler için geçerli olmaktadır. C kategorisi, esnek kaynaklar kullanılarak esnek süreler boyunca gerçekleştirilebilen ve kesintiye uğratılamayan faaliyetleri ifade etmektedir. D kategorisi, esnek kaynaklar kullanılarak esnek sürelerde gerçekleştirilebilen ve kesintiye uğratılabilen aktivitelere atıfta bulunmaktadır. Aktiviteler ayrı ayrı A, B veya C kategorisi altında sınıflandırıldığında KKPPP' leri ve ÇMKKPPP'leri çözmek için birçok algoritma geliştirilmiştir.

Issa ve Tu (2019), KKPPP ve ÇMKKPPP'deki araştırma boşluğunun devam ettiğini ve ele alınması gerektiğini ortaya koymuştur. Mühendislik projelerini daha pratik ve verimli bir şekilde planlamak ve programlamak için aynı anda kullanılan A, B ve D

aktivite kategorileri altında eşsiz üretim (EÜ)'de KKPPP ve ÇMKKPPP'ni çözmek için yöntemler veya buluşsal yöntemlerin geliştirilmesi gerektiği önerilmiştir.

Liu vd. (2019) kaynak kısıtlı proje programlama problemi (KKPPP) için yeni çaprazlama ve mutasyon operatörleri içeren rekabetçi bir genetik algoritma (GA) önerilmiştir. Sınırlı kaynak koşullarında, proje yöneticilerinin en iyi programı hızlı bir şekilde almalarına yardımcı olmayı amaçlamaktadır.

Önerilen GA paradigmasında, yeniden birleştirilecek ebeveynleri seçmek için yeni bir seçim operatörü gibi çeşitli değişiklikler sunmaktadır. Belirli bir çaprazlama sırasına sahip değiştirilmiş iki noktalı çaprazlama operatörü ve doğrusal olarak azalan olasılığa dayalı bir mutasyon operatörü kullanılmaktadır.

Çalışmada GA, 2,4 GHz hızında kişisel bir bilgisayar IBM ThinkPad T440 üzerinde çalışan C++ ile kodlanmıştır. Proje programlama problemi kütüphanesinden (PSPLIB) alınan J30, J60, J90 ve J120 boyutundaki standart kıyaslama problemleri kullanılarak test edilmiştir ve literatürdeki son teknoloji ürünü 19 üst-sezgisel yöntem ile karşılaştırılmıştır.

PSPLIB, sırasıyla 30, 60, 90 ve 120 aktivite ve J30, J60, J90 ve J120 olmak üzere dört tip alt grup içeren Kolisch ve Sprecher (1997) tarafından geliştirilen ProGen problem oluşturucu tarafından oluşturulmuştur. J30, J60 ve J90 alt kümelerinde toplam 480 örnek bulunurken, 600 örnek J120 alt kümesinde bulunmaktadır. Hesaplama sonuçları, önerilen genetik algoritmanın J30 alt kümesi için en uygun çözümleri, J60 alt kümesi için en iyi çözümleri ve 500.000 denemede J120 alt kümesi için neredeyse en iyi çözümleri sağlayabileceğini göstermektedir.

Sonuçlar GA'nın diğer algoritmalarla karşılaştırıldığında 50.000 denemede yeterince iyi olmadığını göstermekte olup bu da algoritmanın yakınsama yeteneğinin daha da iyileştirilmesi gerektiğini, dolayısıyla yeni ve daha iyi bir çaprazlama veya mutasyon süreci tasarlamak için kodlama şemasının karakterinin nasıl kullanılacağına daha fazla araştırmaya ihtiyaç duyulduğu belirtilmektedir.

Kadam ve Kadam (2014) kaynak kısıtlı proje programlama problemini (KKPPP) genetik algoritma (GA) ile çözmüştür. Kaynak ve öncelik kısıtlarını ihlal etmeden proje süresini minimize etmek için JAVA dilinde kodlanan GA ile aktivite öncelikleri optimize edilmiştir. GA'da tek noktadan çapraz geçiş işlemi ve Flip Bit tabanlı mutasyon işlemi kullanılmıştır. Algoritma, PSPLIB'de bulunan Kolisch ve Sprecher (1997) tarafından

geliştirilen ProGen tarafından oluşturulan J30, J60, J90 ve J120 aktivitelek bir dizi örnek üzerinde test edilmiştir. Önerilen algoritma için popülasyon büyüklüğü: 5000, iterasyon sayısı 1000, çaprazlama olasılığı 0,5 ve mutasyon olasılığı ise 0,01 olarak tayin edilmiştir.

Analiz sonuçları Çizelge 2.2 'de sunulmuştur. Tablonun ilk sütunu, örnek kümesini göstermektedir. Diğer sütun, 1000 nesil çözüm için J30, J60, J90 ve J120'nin tüm örneklerinde önerilen algoritma ile elde edilen sonucu temsil etmektedir.

Çizelge 2.2 : Önerilen algoritma ile elde edilen sonuçlar

Örnekler	Ortalama Sapma
J30	0,08
J60	10,31
J90	10,43
J120	25,32

Çizelge 2.3'te , önerilen algoritmanın performansını, optimum çözümden ortalama sapma açısından Hibrit GA, Hibrit Dağılım Araması, Parçacık Sürüş Optimizasyonu gibi diğer algoritmalarla karşılaştırmaktadır.

Çizelge 2.3 : Önerilen algoritmanın performansının diğer algoritmalarla karşılaştırılması

Algoritma	Ortalama Sapma			
	J30	J60	J90	J120
Bu makalede	0,08	10,31	10,43	25,32
[46]	0,04	10,10	11,02	27,65
[48]	13,37	10,81	10,34	32,45
[47]	0,13	11,51	11,51	34,65
[43]	0,27	11,73	11,30	35,22
[44]	0,29	12,03	-	35,71
[45]	0,10	11,59	-	34,07

Kadam ve Kadam (2014) önerilen algoritmanın diğer mevcut yöntemlerle karşılaştırıldığında nispeten iyi performans gösterdiğini göstermektedir.

Kong ve Dou (2020), çoklu zaman kısıtlamaları (ÇZK) altında kaynak kısıtlı proje programlama problemi (KKPPP) olarak adlandırılan yeni bir pratik zamanlama problemi

sunulmaktadır. ÇZK'lı KKPPP'de, aktivitenin süre kısıtlaması, geçici kısıtlama ve kaynak takvimleri kısıtlamasına tabi olarak projenin toplam üretim süresinin en aza indirilmesi amaçlanmıştır. Aktivitenin süre kısıtlaması gerçek hayattaki projelerde yaygın olarak mevcut olup ilk olarak bu çalışmada KKPPP olarak önerilmiştir. Çalışmada, yeni problem için bir kısıt programlama optimizasyon modeli geliştirilmiştir. Bu yeni model CPLEX Studio IDE 12.9.0'da programlanmıştır ve Windows 10'un 64 bit sürümü altında 8 GB RAM ile Intel Core i7 2.60GHz dizüstü bilgisayarda deneyler gerçekleştirilmiştir.

CP optimizasyon algoritmasının ÇZK'lı KKPPP'yi çözmedeki etkinliğini ve doğruluğunu test etmek için farklı örnekler seçilmiştir. Diğer birkaç yaklaşımla karşılaştırıldığında, CP optimizasyonun programlama topluluğunda yeterince bilinmediği fakat programlama problemlerini çözmekte güçlü olduğu belirtilmiştir. CP optimizasyon yazılımı, modellemeyi kolay ve hızlı hale getiren bildirim temelli bir matematiksel model kullanmaktadır. En uygun çözümü bulmak için hibrit yöntemlere dayalı kesin bir algoritma kullanmaktadır. CP optimizasyonun, gerçek hayattaki problemler için iyi bir performansa sahip olduğu belirtilmiştir (Laborie vd., 2018). Bu nedenle, CP optimize edici, ÇZK'lı KKPPP'yi çözmek için iyi bir seçim olduğu belirtilmektedir.

İlk olarak, CP'nin nasıl kullanılacağını tanıtmak için 10 gerçek aktivite içeren bir örnek seçilmiştir. Sonra iki karşılaştırmalı deney yapılmıştır. İlk deney, aynı projeyi ÇZK'lı KKPPP'yi CP ile farklı yöntemler kullanarak çözülmüştür. İkinci deney, yöntemin farklı boyutlardaki projelerle uğraşmak için CP kullanılmıştır. Deneyler, CP'nin hızlı olduğunu ve yüzlerce aktiviteye sahip projeler için diğer metasezgisel yöntemlere kıyasla dakikalar içinde optimuma yakın bir çözüm sağladığını göstermektedir. ÇZK'lı KKPPP'yi, projelerdeki gerçek hayattaki kısıtlamaları çözmek için pratik bir karar destek sistemi geliştirmeye katkıda bulunmuştur.

Kosztván vd. (2019) esnek projelerde ÇMKKPPP ele alınmıştır. Esnek görev bağımlılıklarını ve kararsız, tamamlayıcı görev tamamlamayı barındıran alternatif proje planları için puanlar sağlayan ve aynı zamanda yeni ancak planlanmamış görevleri yerine getiren matris tabanlı bir yöntem önerilmiştir. Önerilen yöntem, esnek ve geleneksel yaklaşımları birleştirebilmektedir.

Önerilen algoritma, hibrit proje yönetimi yaklaşımını destekleyen hızlı ve verimli bir yöntem olmuştur. Algoritma, zaman, maliyet ve kalite gibi faktörler üzerinde önceden tanımlanmış tercihlere göre en uygun çözümü bulabilmektedir. Önerilen algoritma,

planlanmamış görevleri yerine getirebilmektedir. Bu nedenle, önerilen yöntem sürekli değişen bir ortamda kullanılabilir.

Kosztván vd., hibrit bir yaklaşımın esnek proje yönetimi yaklaşımlarına uygun bir alternatif olabileceğini göstermektedir. Geleneksel ve esnek yaklaşımlar, daha uygulanabilir projeleri belirlemek ve proje süresinin çizelgeleme performansını artırmak için birleştirilebilmektedir. Önerilen algoritma, değiştirilmiş MMLIB proje veri tabanında ve ProGEN tarafından üretilen oluşturulan projelerde test edilmiştir.

Önerilen model, geleneksel çoklu KKPPP'ni genişletmektedir; ancak bu modelde yalnızca yenilenebilir kaynaklar (örneğin insan kaynakları) ve yenilenemeyen bir kaynak (örneğin maliyet talebi) dikkate alınmaktadır. Proje yönetiminde yenilenebilir, yenilenemez ve yarı yenilenebilir kaynaklar da önemli parametreler olabilmektedir. Bu nedenle, bu uzantının gelecekteki araştırmalarda dikkate alınacağı belirtilmiştir. Bu yöntemin bir başka olası uygulaması risk yönetimi ve risk analizi olarak belirtilmiştir. Tamamlayıcı ve planlanmamış görev tamamlamaları, yönetim veya müşteri taleplerindeki değişiklikleri esnek görev bağımlılığı ile modelleyebilmektedir. Bu durumlarda, daha uygun bir matris tabanlı model belirlenebilmektedir ve geleneksel, esnek proje yönetimi yaklaşımlarının risk azaltma etkinliği, farklı türdeki proje planları için karşılaştırılabilir.

Lin vd. (2019), çoklu beceri kaynak kısıtlı proje programlama problemi (ÇBKKPPP) için genetik programlama hiper-sezgisel (GP-HS) yaklaşım ele alınmıştır. GP, genetik algoritma (GA) temelinde geliştirilmiş verimli bir algoritma olarak tanımlanmaktadır. GA ile benzer şekilde, GP; başlatma, seçim, geçiş, mutasyon aşamalarını içermektedir. Bununla birlikte, GP, çözümü temsil etmek için değişken boyutlu LISP ağacını (Lon vd., 2017) benimseyerek doğrusal olmayan kodlama stratejisi kullanmaktadır. GP'nin amacı, en optimum değere sahip ağacı bulmak olarak belirtilmiştir.

Çalışmada ilk olarak, çözümü kodlamak için tek bir görev dizisi vektörü kullanılmıştır ve uygulanabilir programlar oluşturmak için onarım tabanlı bir kod çözme şeması önerilmiştir. İkinci olarak, bir dizi düşük düzeyli buluşsal yöntem oluşturmak için 10 basit sezgisel kural tasarlanmıştır. Üçüncüsü, GP, sezgisel alan üzerindeki düşük seviyeli buluşsal yöntemleri esnek bir şekilde yönetebilen yüksek seviyeli bir strateji olarak kullanılmıştır. Parametre ayarlarının etkisini araştırmak için deney tasarımı yöntemi (DTY) kullanılmıştır. Son olarak, GP-HS'nin performansı, 36 örnekten oluşan akıllı çok

amaçlı proje programlama ortamı (AÇAPPO) karşılaştırma veri kümesinde değerlendirilmiştir.

ÇBKPPPP'yi ele almak için önerilen GP-HS'nin performansını değerlendirmek için, Myszkowski vd. (2015) tarafından oluşturulan yeni AÇAPPO karşılaştırma veri kümesi üzerinde sayısal deneyler gerçekleştirilmiştir. AÇAPPO, 100 görevden 200 göreve kadar değişen boyutlarda 36 proje örneğinden oluşmaktadır. ++ GP-HH algoritması, Visual C 6.0'da uygulanmıştır ve 2.3 GHz ve 8 GB RAM'e sahip bir çekirdek i5-7360 U işlemcide çalıştırılmıştır. Her örnek için, algoritma bağımsız olarak 20 kez çalıştırılmıştır.

Makalenin katkıları şu şekilde özetlenmiştir: (1) bir dizi düşük düzeyli buluşsal yöntem oluşturmak için 10 basit buluşsal yöntem kullanılmıştır; (2) GP, düşük seviyeli buluşsal yöntemleri yönetmek için yüksek seviyeli strateji olarak kullanılmıştır; (3) uygulanabilir programları verimli bir şekilde oluşturmak için basit bir kodlama şeması ve yeni bir onarıma dayalı kod çözme şeması geliştirilmiştir; (4) DTY'ye dayalı sayısal testler, parametre ayarlarının etkisini araştırmak için gerçekleştirilmiştir. Mevcut algoritmalarla yapılan karşılaştırmalar, önerilen algoritmanın çözüm kalitesi ve yakınsama yeteneği açısından rakip algoritmalarından daha iyi performans gösterdiğini göstermektedir.

Öztürk ve Öner (2020) çok modlu kaynak kısıtlı proje programlama problemini (ÇMKPPPP) çözmek için karışık tamsayı doğrusal programlama modelleri (KTDPM) ele alınmıştır. Aktivitelerin farklı (çoklu) modlarda yürütülebildiği kaynak kısıtlı proje programlama problemleri (KKPPPP) için iki yeni sürekli zamanlı KTDPM geliştirilmiştir.

Çalışmada olay ve kaynak akışı konseptlerine dayalı iki yeni KTDPM önerilmiştir. Bunlar , Çok Modlu Başlangıç / Bitiş Olayına Dayalı Model ve Çok Modlu Akış Tabanlı Modellerdir. Önerilen modellerin her ikisi de, Kolisch vd. (1995) tarafından Proje Programlama Problem Kitaplığı'ndan (PPPK) alınan sayısız problem çözülerek test edilmiştir. Karşılaştırma için ayrıca, Pritsker vd. (1969) tarafından önerilen ayırık zaman (AZ) modeli de dahil edilmiştir. Tüm modeller, Intel Core i5 işlemci, 4 GB RAM ve 2.30 GHz CPU'lu bir bilgisayarda LINGO optimizasyon yazılımı kullanılarak çözülmüştür.

Makalede PPPK'dan tek ve çok modlu KKPPPP için aktivite sayısının sırasıyla 10, 12, 14, 16, 18 ve 20 olduğu çok modlu problem setleri kullanılmıştır. Her set 10 farklı problem içermektedir. Tüm problemlerde iki ayrı kaynak ve tüm aktivitelerin tanımlanmış 3 ayrı modu yer almaktadır. Zaman aralığı arttıkça akış temelli modelin problemleri ayırık zaman modellerine göre daha hızlı çözülmektedir.

Pellerin vd. (2019) KKPPP'ni çözmek için çoklu meta sezgisel yaklaşımların kombinasyonuna dayalı hibrit meta sezgisel algoritma geliştirmiştir.

Çalışmada ilk olarak, KKPPP'nin çözümü için son 20 yılda geliştirilen hibrit meta sezgisel algoritmaların temel unsurları gözden geçirilmiştir. İkinci olarak, PSPLIB veri örneklerindeki farklı hibritlerin deneysel sonuçları rapor edilmiştir ve mevcut olan en başarılı KKPPP hibritlerini belirlemek için hesaplamalı bir karşılaştırma yapılmıştır. Son olarak, bu karşılaştırmadan, en iyi KKPPP hibritlerinin ortak temel stratejileri incelenmiştir.

Çalışmada PSPLIB proje kitaplığından (Kolisch ve Sprecher, 1996; Kolisch, Sprecher ve Drexler, 1995) kıyaslama örnekleri setlerinde test edilmiş hibrit yöntemler ele alınmıştır. J30 seti için, hibrit yöntemlerin çoğunun optimuma yakın çözümler üretebileceği belirtilmiştir. J60, J90 ve J120 için sonuçlar 50.000'den fazla program için önemli ölçüde iyileştirilmiştir. Bu iyileştirmenin, araştırmacıları, 200 ve 500 aktiviteye kadar çok daha büyük problem setlerini çözmek ve daha büyük program limitleri için yöntemlerini test etmek, verimli ve etkili KKPPP hibrit meta sezgisellerini daha da geliştirmeye teşvik etmesi gerektiği belirtilmiştir.

KKPPP hibritleri, çalışma zamanından önce veya arama sırasında uyarlamalı veya kendi kendine uyarlanabilir kontrol yoluyla ayarlanmadığı sürece çok fazla parametre içermemesi gerektiği belirtilmiştir.

Rahman vd. (2019), KKPPP'ni çözmek için bir GA tabanlı memetik algoritma (MA) sunulmuştur. Algoritma, kritik yol tabanlı buluşsal yöntem ve Nawaz, Ensore ve Ham (NEH) buluşsal yönteminin bir varyantı tarafından başlatılmıştır. Benzer bir blok sırası geçişini ve sıraya dayalı yerel aramadaki bir değişkeni içermektedir. Algoritmanın yerel optimalden uzaklaşmasına yardımcı olan bir otomatik yeniden başlatma şeması sunulmuştur. Önerilen MA için uygun parametreler setini belirlemek için bir DTY kullanılmıştır.

Çalışmada PSPLIB'den alınan J30, J60 ve J90 veri kümelerinin her biri 480 ve J120 veri kümesi 600 problem örneği içermektedir. RG30 veri kümesi 1800 ve RG300 veri kümesi 480 örnek içermektedir. Bu problem örneklerinin karmaşıklığı üç parametreye bağlıdır: (1) ağ karmaşıklığı (AK); (2) kaynak faktörü (KF); ve kaynak gücü (KG). Tüm sayısal deneyler, 3.40 GHz CPU ve 16 GB RAM'e sahip bir Intel corei7 işlemci üzerinde gerçekleştirilmiştir. Algoritma C ++ ile kodlanmıştır.

Sonuçlar, önerilen algoritmanın, özellikle J30, J60 ve J90 setleri için yüksek kaliteli çözümlere yakınsama kabiliyetine sahip olduğunu göstermiştir. Gelecekteki çalışmalarda, J120 ve RG300 problem setlerinin algoritmanın performansını hızlandırmak için daha fazla araştırmaya ihtiyaç olduğu belirtilmiştir. Geliştirilen MA, projeleri daha verimli yürütmek ve mevcut kaynakları daha verimli kullanmak ve böylece proje yürütme maliyetlerini düşürmek için proje yöneticileri tarafından kullanılabileceği belirtilmiştir. Önerilen yaklaşımın proje yöneticilerinin makul hesaplama süresi içinde tatmin edici bir program elde etmelerine yardımcı olmaktadır. Sunulan algoritmaların ÇMKKPPP'i gibi gerçek hayattan diğer proje programlama problemlerine gelecekteki çalışmalarda genişletebileceği belirtilmiştir.

Ren vd. (2020), uçağın hareket eden montaj hattının kaynak aktarım süresine sahip kaynak kısıtlı proje programlama problemi için dallara bağlı gömülü GA ile çözümünü ele almıştır. Uçak montaj hattı üretim programlama problemini KKPPPTS olarak soyutlayıp, dallara ve sınırlara gömülü bir GA önerilmiştir.

Önerilen algoritmanın özelliklerine göre, sadece GA'da kodlanabilen değil, aynı zamanda dal ve sınır algoritmasında arama ağacının yapısına da dönüştürülebilen mutlak öncelik tabanlı bir kodlama yöntemi tasarlanmıştır. Karşılık gelen çaprazlama ve mutasyon operatörü, çapraz geçiş ve mutasyon işleminden sonra uygulanabilir çözümleri garanti etmek için de geliştirilmiştir. İlk dal genetik algoritma tarafından üretilmiştir ve çözümün hesaplama doğruluğunu iyileştirmek için dal ve sınır algoritması ile ilk daldan aramaya devam edilmiştir.

Çalışmada çözüm yaklaşımları C# ile kodlanmış ve Intel Core i7 2.40 GHz CPU ve 8 GB RAM ile kişisel bir bilgisayarda test edilmiştir. CPLEX'in sürümü ILOG CPLEX 12.5.1.0 ve hesaplama ortamı Microsoft Visual Studio 2013, Windows 10, 64 bit'tir. Her bir örnek için GA ve BEGA simülasyonu 15 bağımsız kez çalıştırılmıştır. Tüm test GA ve BEGA'nın durdurma koşulu (yani, maksimum hesaplama süresi, maksimum yineleme sayısı), yineleme sayısı 50 olarak ayarlanmıştır.

Test verileri, PSPLIB kitaplığının Kolisch ve Sprecher (1997) tarafından oluşturulan standart karşılaştırma veri seti probleme uygun biçime dönüştürülerek elde edilmiştir. İş boyutu $n \in \{10, 12, 14, 16, 18, 20, 30, 60, 90, 120\}$ olarak ayarlanmıştır. Öncelik kısıtlamaları, standart karşılaştırma verilerine dayanmaktadır. Farklı türdeki kaynakların kaynak aktarım süreleri, 1-15 aralığında rastgele oluşturulmuştur.

Makalede önerilen kaynak aktarım süreleri dikkate alınarak baskınlık kuralları ve dinamik alt sınırın, dallanma ve sınırlama algoritmasının hesaplanmasını hızlandırmada etkili olduğu kanıtlanmıştır. GA ile karşılaştırıldığında, dal ve bağlı gömülü GA'nın daha iyi sonuçlar elde edebileceği belirlenmiştir. Hesaplama sonuçları, küçük ölçekli problemlerle uğraşırken, dallara bağlı gömülü GA'nın iyi performans gösterdiğini ve ortalama sapmanın optimal çözümden sadece yaklaşık %2 olduğunu göstermiştir. Orta ve büyük ölçekli problemlerde, algoritma ayrıca KKPPPTS'yi çözmek için önerilen mevcut algoritmadan daha iyi performans gösterdiği belirtilmiştir.

Riedler vd. (2017), yüksek çözünürlüklü, kaynakla sınırlı bir proje zamanlama problemi için yinelemeli bir zaman bölümü iyileştirme algoritması ele alınmıştır. Planlamanın yüksek çözünürlükte yapılması gereken, kanser tedavisi için partikül terapisinden kaynaklanan, kaynakları kısıtlı bir proje planlama problemini ele alıyoruz. Zaman indeksli formülasyonlar veya ayrık olay formülasyonları gibi geleneksel karışık tamsayı doğrusal programlama tekniklerinin, bu gibi durumlarda ciddi sınırlamalara sahip olduğu, yani çok hızlı büyüme veya zayıf doğrusal programlama gevşemelerine sahip olduğu bilinmektedir. Zamanı, zaman dilimlerine ayırmaya dayalı bir gevşeme öneriyoruz. Bu gevşeme yinelemeli olarak çözülür ve buluşsal yöntemler kullanılarak uygulanabilir çözümler türetmek için temel oluşturur. Bu ilkel ve ikili çözümlere ve sınırlara dayanarak, zaman aralıkları art arda rafine edilir. Bu parçaları birleştirerek, kısa sürede iyi yaklaşık çözümler sağlayan ve sonunda en uygun çözüme yaklaşan bir algoritma elde ederiz. Zaman aralığı ayrıntılandırmasını gerçekleştirmek için çeşitli stratejiler araştırılır. Yaklaşım, geleneksel formülasyonlara ve bir meta-sezgiselliğe kıyasla mükemmel performans gösterir.

Kıyaslama örnekleri, kanser tedavi merkezi MedAustron'da partikül tedavilerinin planlanmasını gerektiren gerçek dünya hasta planlama senaryosu tarafından motive edilir. Genel olarak, her tedavi seansı, sırayla gerçekleştirilmesi gereken beş aktiviteden oluşur. Modellenen kaynaklar partikül ışını, ışınlama odaları, radyo onkologları ve anesteziştir. Prensipte olarak, kaynakların kısa zaman dilimleri dışında tüm zaman ufku için mevcut olduğu varsayılır.

Karşılaştırmalı örneklerimizin temel özelliği, etkinliklerin sayısıdır. Her biri etkinlik sayısı başına 15 bulut sunucusundan oluşan iki grup karşılaştırma örneği oluşturulmuştur. Test çalıştırmaları, 7200 saniyelik bir zaman sınırı ve 4GB RAM bellek

sınırı kullanılarak 2.53 GHz ile Intel Xeon E5540 üzerinde gerçekleştirilmiştir. MILP modelleri, tek bir iş parçacığı ile Gurobi 7 kullanılarak çözülmüştür.

Sallam, Chakraborty ve Ryan (2020), KKPPP'yi çözmek için iki aşamalı çok operatörlü diferansiyel evrim (DE) algoritması olan TS-MODE'yi geliştirmiştir. TS-MODE, keşif aşamasıyla başlayıp ve popülasyon çeşitliliği ve çözümlerin kalitesine dayalı olarak, bu yaklaşım dinamik olarak en uygun DE'ye daha fazla önem vermektedir ve ardından aynı işlemi kullanma aşamasında tekrar etmektedir.

Makalenin amacı, klasik KKPPP'yi çözebilen verimli bir algoritma sunmaktır. Bu algoritma, tek bir algoritmik çerçevede birden fazla DE mutasyon stratejisinin gücünü kullanmaktadır ve her operatör tarafından üretilen çözümlerin kalitesi ve çeşitliliği olmak üzere iki kritere göre en uygun DE operatörüne daha fazla ağırlık vermektedir. Üretilen çözümün kalitesini daha da iyileştirmek için, çok sayıda yerel arama tekniği (ileriye doğru iyileştirme ve komşu aramalar) kullanılmaktadır.

Çalışmada, TS-MODE'un performansını değerlendirmek için, PSPLIB'den alınan J30 (480 örnek), J60 (480 örnek) ve J120 (600 örnek) (Kolisch ve Sprecher, 1997) aktivite veri kümeleri kullanılmıştır. Bu deneyler Windows 10, 3.4 GHz Core I7 işlemci ve 16 GB RAM içeren bir bilgisayarda gerçekleştirilmiştir. Önerilen algoritma MATLAB'da kodlanmıştır.

Mevcut algoritmalarla karşılaştırıldığında, önerilen TS MODU'nun KKPPP'ni çözmede etkili olduğunu göstermiştir. Gelecekteki araştırmalarda, parametre ayarlaması ve ortamı için kendi kendine uyarlanabilir bir mekanizma tasarlamaya, DE operatörlerinin seçiminde peyzaj ve takviye tabanlı yaklaşımların kullanımını araştırmaya ve çok modlu KKPPP'ni çözmek için önerilen TS-MODE'u genişletme hedeflenmiştir.

Snauwaert ve Vanhoucke (2020), KKPPP için geniş ve derinlikli yeteneklere sahip yeni bir algoritma ele almıştır. ÇMKKPPP'ni verimli bir şekilde çözebilmek için genişlik ve derinlik yeteneğine sahip bir GA geliştirilmiştir. Probleme özgü yöntemi, yeni bir temsil, yeni bir geçiş ve kişiye özel yerel aramaları birleştirmektedir.

Çalışmanın amacı, verimli bir işgücünün özelliklerini bir üretim süresi hedefi altında incelemektir. Projelerin programlanması ve atanması için üç yerel arama sunulmaktadır. Çözüm yaklaşımları C++ ile kodlanmış ve 2,3 GHz Intel Xeon Gold 6140 işlemci üzerinde test edilmiştir. Algoritmanın parametrelerinin ince ayarını yapmak için tüm veri kümesini temsil eden küçük bir 81 örnek alt kümesi kullanılmıştır. Elde edilen

parametreler: 75 çözüm vektörünün bir popülasyon boyutu, 0,8'lik çaprazlama ve 0.2'lik mutasyon oranı ile 15,000 iş çizelgesi oluşturulduktan sonra hesaplamalar kesilmiştir.

Veriler, problemi karakterize eden önceden tanımlanmış proje parametreleri kullanılarak oluşturulacaktır. Bu parametreler, 6.250 projeden oluşan bir veri seti üretilebilecek şekilde farklı değerlere ayarlanacaktır.

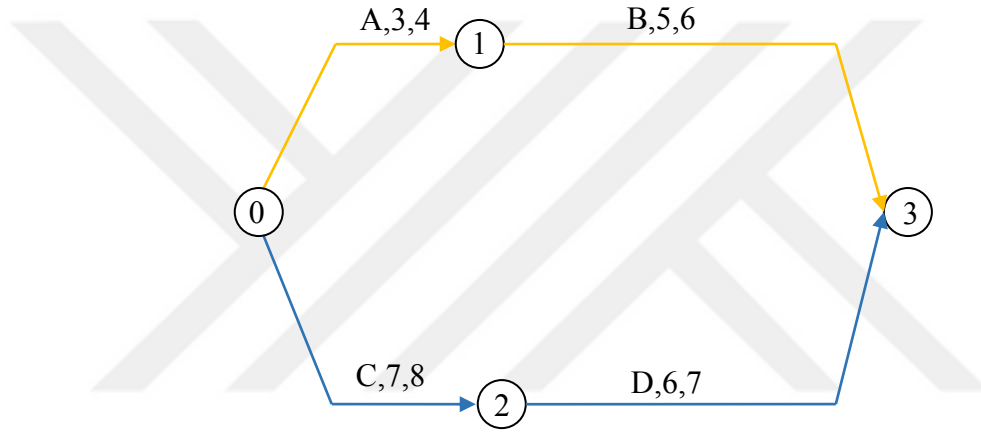
Gelecekteki araştırmalarda ÇMKKPPP 'ne öğrenmenin entegrasyonuna odaklanabileceği belirtilmiştir. Çalışanların verimliliği çok dinamik bir kavram olma eğiliminde olduğu için, bir aktivitenin farklı beceriler gerektirebileceğini hesaba katarak öğrenmeden etkilenen ayarlanabilir bir derinliği dahil etmek ÇMKKPPP 'ne umut verici bir katkı olabileceği belirtilmiştir.



3. VAKA ÇALIŞMALARI

3.1 Örnek Problem 1

KKPÇP'nin çözümü için hazırlanan hesap cetveli uygulamasının oluşturulması dört aktiviteli bir problem üzerinden anlatılmıştır. Şekil 3.1'de ok diyagramı verilen projede 4 aktivite yer almakta ve aktiviteler A, B, C ve D olarak adlandırılmıştır. Şebekeyi oluşturan ve toplam dört tane olan düğüm noktaları 0, 1, 2 ve 3 olarak adlandırılmıştır. Aktivite süreleri ve aktivitelerin ihtiyaç duyduğu kaynak miktarı aktiviteyi temsil eden okların üzerindeki etiketlerde belirtilmiştir.



Şekil 3.1: Örnek proje-1

Projenin analiz edilebilmesi için problemin hesap tablosu uygulamasına tanıtılması gerekmektedir. Aktivite isimleri Şekil 3.2'de gösterildiği gibi uygulamanın D sütununda girilmektedir. Her bir aktivitenin başlangıç ve bitiş düğüm noktaları sırası ile B ve C sütunlarında tanımlanmıştır. Aktivite süreleri E sütununa, aktivitelerin kaynak ihtiyaçları ise F sütununda her aktivite için ilgili satırlara Şekil 3.2'de gösterildiği gibi tanımlanmıştır. Bu kısım projede yer alan aktivitelerle ilgili veri girişinin yapıldığı sütunlardır.

Bu aşamada projeyi oluşturan aktivitelerin hangi başlangıç ve bitiş düğüm noktaları arasında olduğu belirtilmiştir. Örneğin A aktivitesinin başlangıç düğüm noktası 0 ve bitiş düğüm noktasının 1 olduğu sırası ile B3 ve C3 düğüm noktalarına girilen 0 ve 1 değerleri ile tanımlanmıştır. Diğer aktivitelerin başlangıç ve bitiş düğüm noktaları aynı yöntemle tanımlanmıştır.

Aktivite süreleri ve Şekil 3.1'de sunulan şebeke diyagramı kullanılarak tüm düğüm noktalarının ileri gidişle Erken Olay Zamanı (EOZ) ve Geç Olay Zamanı (GOZ) hesaplanmıştır. Ok diyagramı şebekelerinin çözümünde ilk önce ileri gidişle EOZ hesaplanır. İlk düğüm noktası olan sıfıncı düğüm noktasının EOZ'si 0'dır. Bir sonraki olay zamanı olan 1. düğüm noktası için EOZ hesaplaması sıfıncı düğüm noktasının erken olay zamanı olan EOZ0'a A aktivitesinin süresi eklenerek gerçekleştirilir. Belirtilen işlemin hesap tablosu üzerinde tanımlanması Şekil 3.1'de gösterilmektedir.

	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
	Baş	Bit	Aktivite	Süre	Kaynak	EOZ _i	GOZ _i	EOZ _j	GOZ _j	TB	Baş	Bit		Mak işçi	12
	0	1	A	3	4	0	0	3	8	5	0	3			
	1	3	B	5	6	3	8	13	13	5	3	8			
	0	2	C	7	8	0	0	7	7	0	0	7			
	2	3	D	6	7	7	7	13	13	0	7	13			
DN	EOZ	GOZ													
	0	0	0												
	1	3	8												
	2	7	7												
	3	13	13												

Şekil 3.2: Veri giriş ekranı.

TOPLA																
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2		Baş	Bit	Aktivite	Süre	Kaynak	EOZ _i	GOZ _i	EOZ _j	GOZ _j	TB	Baş	Bit		Mak işçi	12
3		0	1	A	3	4	0	0	3	8	5	0	3			
4		1	3	B	5	6	3	8	13	13	5	3	8			
5		0	2	C	7	8	0	0	7	7	0	0	7			
6		2	3	D	6	7	7	7	13	13	0	7	13			
7																
8		DN	EOZ	GOZ												
9		0	0	0												
10		1	=C9+E3		8											
11		2	7		7											
12		3	13	13												

Şekil 3.3: Erken Olay Zamanı hesaplamasının hesap tablosunda gerçekleştirilmesi.

TOPLA															
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1		Baş	Bit	Aktivite	Süre	gecikme	yalın süre	Kaynak	EOZ _i	GOZ _i	EOZ _j	GOZ _j	TB	Başlangıç	Bitiş
2		0	1	A	3	0	3	4	0	0	3	8	5	0	3
3		1	3	B	5	0	5	6	3	8	13	13	5	3	8
4		0	2	C	7	0	7	8	0	0	7	7	0	0	7
5		2	3	D	6	0	6	7	7	7	13	13	0	7	13
6															
7		DN	EOZ	GOZ											
8			0	0	0										
9			1	3	8										
10			2	7	7										
11															

Şekil 3.4: Geç Olay Zamanı hesaplamasının hesap tablosunda gerçekleştirilmesi.

TOPLA													
=DÜŞEYARA(B3;\$B\$9:\$D\$12;2;YANLIŞ)													
	A	B	C	D	DÜŞEYARA(aranan_değer; tablo_dizisi; sütun_indis_sayısı; [aralık_bak])						K	L	M
1													
2		Baş	Bit	Aktivite	Süre	Kaynak	EOZ _i	GOZ _i	EOZ _j	GOZ _j	TB	Baş	Bit
3		0	1	A	3	4	=DÜŞEYARA	0	3	8	5	0	3
4		1	3	B	5	6	3	8	13	13	5	3	8
5		0	2	C	7	8	0	0	7	7	0	0	7
6		2	3	D	6	7	7	7	13	13	0	7	13
7													
8		DN	EOZ	GOZ									
9		0	0	0									
10		1	3	8									
11		2	7	7									
12		3	13	13									

Şekil 3.5: Aktivitelerin Erken ve Geç Başlangıç Zamanlarının hesaplanması

2 numaralı düğüm noktasına sadece 0 numaralı düğüm noktasından geliş vardır ve 0-2 hattında C aktivitesi yer almaktadır. Bu durumda $EOZ_2 = EOZ_0 + SÜRE_C$ biçiminde formüleleştirilebilir. Son düğüm noktası olan 3 numaralı düğüm noktasına hem 1 numaralı hem de 2 numaralı düğüm noktalarından gelinebilmektedir. EOZ hesaplanırken tüm gelişlerin en uzun süreni alınır. Böylece $EOZ_3 = \text{MAKSİMUM}(EOZ_1 + SÜRE_B; EOZ_2 + SÜRE_D)$ formülü ile ifade edilir. Son düğüm noktasının hesaplanması ile ileri gidiş tamamlanmış ve tüm düğüm noktalarının EOZ'leri hesaplanmıştır.

Bu aşamadan sonra düğüm noktalarının GOZ'leri hesaplanmıştır. Son düğüm noktası olan üçüncü düğüm noktasının GOZ'si için $GOZ_3 = EOZ_3$ eşitliği tanımlanır. Bir önceki olay zamanı olan 2. düğüm noktasının GOZ hesabı Şekil 3.4'te belirtildiği gibi üçüncü düğüm noktasından D aktivite süresinin çıkarılmasıyla $GOZ_2 = GOZ_3 - SÜRE_D$ olarak hesaplanmıştır. Birinci düğüm noktasının GOZ hesaplaması 3. düğüm noktasının GOZ değerinden B aktivite süresinin çıkarılmasıyla bulunmuştur ($GOZ_1 = GOZ_3 - SÜRE_B$). Sıfırcı düğüm noktasına iki dönüş vardır ve GOZ hesaplanırken tüm dönüşlerin en kısa süreni alınmaktadır. Sıfırcı düğüm noktası $GOZ_0 = \text{MİNİMUM}(GOZ_2 - SÜRE_C; GOZ_1 - SÜRE_B)$ olarak hesaplanmıştır.

Olay zamanlarının gösteriminde i indisi aktivitenin başlangıç düğüm noktası, j indisi ise aktivitenin bitiş düğüm noktasını ifade etmek için kullanılmıştır. Aktivitelerin EOZ ve GOZ hesaplandıktan sonra aktivitelerin I, J, K ve L sütunlarındaki EOZ_i ve GOZ_j verileri Şekil 5'te formülü açık şekilde belirtilen düşeyara komutu ile hesaplanmıştır. Düşeyara tablonun en sol sütunundaki değeri arar ve daha sonra aynı satırda belirtilen sütundan bir değer vermektedir. Şekil 3.5'te A aktivitesinin EOZ_i değerini veren ifade gösterilmektedir. Formülde belirtilen B3 satırı aranan değerdir, yani başlangıç düğüm noktasıdır. Tablonun alt satırında girilen düğüm noktası (DN), EOZ ve GOZ sütunları düşey ara formülünde girilen tablo dizisini tanımlamaktadır. Formülde 2 verisi ikinci sütun

olan EOZ sütununu belirtmektedir. Formülde belirtilen YANLIŞ verisi tam eşleşme istendiği için girilmiştir.

Formül tüm aktiviteler için tanımlanmış ve EOZ_i , GOZ_i , EOZ_j ve GOZ_j değerleri sırası ile I, J, K ve L sütunlarında gösterilmiştir. Aktivitelerin başlangıç zamanı EOZ_i değerine eşit kabul edilerek erken başlangıç zamanları belirlenmiştir. Başlangıç zamanlarına aktivite süreleri eklenerek bitiş süreleri hesaplanmıştır. M sütununda TB olarak belirtilen toplam bolluk süreleri $GOZ_j - EOZ_i - \text{Aktivite Süresi}$ ifadesi ile hesaplanmıştır. Şekil 3.6'da A aktivitesinin toplam bolluğunun hesaplanması gösterilmektedir. Formülde L3 hücresi A aktivitesinin GOZ_j 'si, I3 hücresi EOZ_i ve son olarak E3 hücresi ise A aktivitesinin süresidir. Aynı formül tüm aktiviteler için uygulanarak B, C ve D aktivitelerinin de toplam bolluğu bulunmuştur.

N ve O sütununda belirtilen Başlangıç ve Bitiş zamanları Şekil 3.7'de belirtildiği gibi hesaplanmıştır. A aktivitesinin başlangıç zamanı, F sütununda Gecikme olarak tanımlanan A aktivitesinin gecikmesi süresi ile EOZ_i değerini toplanması ile hesaplanmıştır. Belirtilen ifade $BAŞLANGIÇ_A = EOZ_{iA} + GECİKME_A$ ile formülleştirilebilir. Aynı işlem diğer aktiviteler içinde yapılarak tüm aktivitelerin başlangıç zamanı hesaplaması yapılmıştır.

O sütununda Bitiş olarak tanımlanan bitiş zamanları Şekil 3.8'de belirtildiği gibi aktivitenin başlangıç zamanına aktivite süresinin eklenmesi ile hesaplanmıştır. Aynı işlem B,C ve D aktiviteleri içinde uygulanarak tüm aktivitelerin bitiş süreleri bulunmuştur.

Şekil 3.9'da Gantt şemasının hazırlanışı gösterilmektedir. C14 hücresinden başlayarak sağ taraftaki diğer hücrelere iş gününü temsil eden rakamlar girilmiştir. Her satır bir aktiviteye ait olacak şekilde aktivitenin hangi iş günlerinde yürütüldüğünü belirleyen formül tanımlanmıştır. Şekil 3.9'da tanımlanan $EĞER(VE.....)$ formülü iş gününün aktivitenin başlangıç zamanından büyük, tamamlanma zamanına küçük eşit olması durumunda ilgili hücreye aktivitenin kaynak talebini, aksi durumda ise 0 değerini yazdırmaktadır. Koşullu biçimlendirme ile sıfırdan büyük değere sahip hücrelere yeşil renk verilerek Şekil 3.9'da gösterilen Gantt Şema görüntüsü elde edilmiştir.

TOPLA													
	A	B	C	D	E	F	G	H	I	J	K	L	M
1													
2		Baş	Bit	Aktivite	Süre	Kaynak	EOZ _i	GOZ _i	EOZ _j	GOZ _j	TB	Baş	Bit
3		0	1 A		3	4	0	0	3	8	=J3-G3-E3	0	3
4		1	3 B		5	6	3	8	13	13	5	3	8
5		0	2 C		7	8	0	0	7	7	0	0	7
6		2	3 D		6	7	7	7	13	13	0	7	13
7													
8		DN	EOZ	GOZ									
9		0	0	0									
10		1	3	8									
11		2	7	7									
12		3	13	13									

Şekil 3.6: Toplam bolluk sürelerinin hesaplanması

=I3+F3														
B	C	D	E	F	G	H	I	J	K	L	M	N	O	
Baş	Bit	Aktivite	Süre	gecikme	yalın süre	Kaynak	EOZ _i	GOZ _i	EOZ _j	GOZ _j	TB	Başlangıç	Bitiş	
0	1 A		3	0	3	4	0	0	3	8	5	=I3+F3	3	
1	3 B		5	0	5	6	3	8	13	13	5	3	8	
0	2 C		7	0	7	8	0	0	7	7	0	0	7	
2	3 D		6	0	6	7	7	7	13	13	0	7	13	
DN	EOZ	GOZ												
0	0	0												
1	3	8												
2	7	7												
3	13	13												

Şekil 3.7: Başlangıç sürelerinin hesaplanması

Örnek 15.01.20 [Uyumluluk Modu] - Microsoft Excel

Dosya

Giriş

Ekle

Sayfa Düzeni

Formüller

Veri

Gözetim Geçir

Görünüm

Yapıştır

Kes

Kopyala

Biçim Boyacı

Pano

Arial

10

A⁺

A⁻

Şekil 3.8: Bitiş sürelerinin hesaplanması

TOPLA																					
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
=EĞER(VE(\$N3<\$S14;\$O3>=\$C\$14);\$H3,0)																					
Baş	Bit	Aktivite	Süre	gecikme	yalın süre	Kaynak	EOZ _i	GOZ _i	EOZ _j	GOZ _j	TB	Başlangıç	Bitiş		Mak işçi	12		aktivite	öncelik		
0	1 A		3	0	3	4	0	0	3	8	5	0	3					A	1 A		
1	3 B		5	0	5	6	3	8	13	13	5	3	8					B	2 B		
0	2 C		7	0	7	8	0	0	7	7	0	0	7					C	3 C		
2	3 D		6	0	6	7	7	7	13	13	0	7	13					D	4 D		
DN	EOZ	GOZ																			
0	0	0																			
1	3	8																			
2	7	7																			
3	13	13																			
Aktivite/Süre																					
A	EĞER(VE	4	3	6	8	8	6	8													12
B		4	3	6	8	8	6	8													20
C		8	8	8	8	8	8	8													56
D																					

Şekil 3.9: Aktivite/Süre hesaplanması

Gantt şemasının oluşturulmasından sonra günlük kaynak talebi Şekil 3.10'da 20. satırda gösterildiği gibi hesaplanmıştır. Temin edilebilen kaynak miktarının aşılması

halinde o gün yürütülen aktiviteleri belirleyen formül Şekil 3.10'da belirtildiği gibi 23 ve 26. Satırlardaki tüm hücrelere tanıtılır. Şekil 3.10'da 4, 5, 6 ve 7. Günlerde B ve C aktivitelerinin 8. günde ise B ve D aktivitelerinin yürütülmesi nedeniyle kaynak çakışmasına yol açmıştır.

C23		=EĞER(VE(C\$20>SR\$2:C15>0),\$B23;"")																			
	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1																					
2	Baş	Bit	Aktivite	Süre	gecikme	yalın süre	Kaynak	EOZİ	GOZİ	EOZJ	GOZJ	TB	Başlangıç	Bitiş		Mak işçi	12		aktivite	öncelik	
3	0	1	A	3	0	3	4	0	0	3	8	5	0	3					A	1	A
4	1	3	B	5	0	5	6	3	8	13	13	5	3	8					B	2	B
5	0	2	C	7	0	7	8	0	0	7	7	0	0	7					C	3	C
6	2	3	D	6	0	6	7	7	7	13	13	0	7	13					D	4	D
7																					
8	DN	EOZ	GOZ																		
9	0	0	0																		
10	1	3	8																		
11	2	7	7																		
12	3	13	13																		
13																					
14	Aktivite/Sür	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
15	A	4	4	4																	
16	B	6	6	6	6	6	6	6	6												
17	C	8	8	8	8	8	8	8	8												
18	D									7	7	7	7	7	7						
19																					
20		12	12	12	14	14	14	14	13	7	7	7	7	7							
21																					
22	çakışan akt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
23	A																				
24	B				B	B	B	B	B												
25	C				C	C	C	C	C												
26	D									D											

Şekil 3.10: Çakışan Aktivitelerin belirlenmesi

Şekil 3.11'de kaynak çakışması görülen aktivitelerden hangisine öncelik verileceği gösterilmiştir. Formül, hücrenin boş olmaması halinde hangi aktivite adının olduğunu bulup ilgili aktiviteye hangi önceliğin atandığını tabloda arama yaparak bulmakta ve öncelik değerini ilgili hücreye yazmaktadır. Kaynak önceliklerinin T sütununa aktivite isimlerinin U sütununa kullanıcı tarafından girilmesi gerekmektedir. Örneğin A aktivitesinin önceliği 1 rakamı ile birinci öncelik olarak B aktivitesinin önceliği ise 2 olarak numaralandırıldığı durum için 2 aktivite arasında kaynak çakışmasının olması halinde B aktivitesinin önceliği daha düşük olduğu için B aktivitesi ertelenecektir. Hesapların sonucu aşağıdaki hesap cetvelinde gösterildiği gibi 4, 5, 6 ve 7. günlerde çakışan B ve C aktivitelerinin öncelik sırası, 8. günde de B ve D aktivitelerinin öncelik sırası 29 ve 32. satırlarda verilmiştir.

TOPLA		=EĞER(C23<>"",DÜŞEYARA(C23;T3\$3:SÜ\$6;2;YANLIŞ);"")																												
	B	C	D	E	[EĞER(mantıksal_sınama;[eğer_doğruysa_değer];[eğer_yanlışsa_değer])																M	N	O	P	Q	R	S	T	U	V
	Baş	Bit	Aktivite	Süre	gecikme	yalın süre	Kaynak	EOZ1	GOZ1	EOZ2	GOZ2	TB	M	Başlangıç	Bitiş		Mak İşçi		12	S	aktivite	öncelik								
2		0	1 A	3	0	3	4	0	0	3	8	5	0	3	8						A	1	A							
3		1	3 B	5	0	5	6	3	8	13	13	5	3	8							B	2	B							
4		0	2 C	7	0	7	8	0	0	7	7	0	0	7							C	3	C							
5		2	3 D	6	0	6	7	7	7	13	13	0	7	13							D	4	D							
6																														
7																														
8	DN	EOZ	GOZ																											
9		0	0	0																										
10		1	3	8																										
11		2	7	7																										
12		3	13	13																										
13																														
14	Aktivite/Süre	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20									
15	A	4	4	4																										
16	B				6	6	6	6	6																					
17	C	8	8	8	8	8	8	8	8																					
18	D									7	7	7	7	7	7															
19																														
20		12	12	12	14	14	14	14	14	7	7	7	7	7	7															
21																														
22	çakışan akt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20									
23	A																													
24	B				B	B	B	B	B																					
25	C				C	C	C	C	C																					
26	D									D																				
27																														
28	Ç önceliği																													
29	A	SEYARA/C																												
30	B				2	2	2	2	2																					
31	C				3	3	3	3	3																					
32	D									4																				

Şekil 3.11: Aktivite önceliğinin tespit edilmesi

TOPLA		=MAK(C29:C32)																			
	B	C	D	E	MAK(sayı1; [sayı2]; ...)	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
5		0	2 C		7	0	7											C			
6		2	3 D		6	0	6	7	7	13	13	0	7	13				D		3 C	
7																				4 D	
8	DN	EOZ	GOZ																		
9		0	0	0																	
10		1	3	8																	
11		2	7	7																	
12		3	13	13																	
13																					
14	Aktivite/Sür	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
15	A	4	4	4																	
16	B				6	6	6	6	6												
17	C	8	8	8	8	8	8	8	8												
18	D									7	7	7	7	7	7						
19																					
20		12	12	12	14	14	14	14	13	7	7	7	7	7							
21																					
22	çakışan akt	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
23	A																				
24	B				B	B	B	B	B												
25	C				C	C	C	C	C												
26	D									D											
27																					
28	Ç önceliği																				
29	A																				
30	B				2	2	2	2	2												
31	C				3	3	3	3	3												
32	D									4											
33																					
34																					
35	Ert. Akt				0	0	3	3	3	3	4	0	0	0	0	0	0	0	0	0	0
36					C	C	C	C	C	D											

Şekil 3.12: En düşük önceliğin belirlenmesi

Çakışma önceliği hesaplanan aktivitelerin hangisinin erteleneceği Şekil 3.12’de sunulan ekran görüntüsünde belirtildiği gibi MAK komutu ile hesaplanmıştır. Bu formül bir rakam kümesindeki en büyük değeri vermektedir. C aktivitesinin önceliği B aktivitesinden daha düşük olduğu için kaynak çakışmasının olduğu 4, 5, 6 ve 7. günlerde C aktivitesinin ertelenmesi gerekmektedir. Projenin 8. günü için D aktivitesinin ertelenmesi önerilmektedir. İlk etapta bu ihmal edilecektir çünkü C aktivitesinin 4 gün ertelenmesinin ardından kaynak çakışmasının değişme ihtimali bulunmaktadır.

Şekil 3.12’de gösterildiği üzere kaynak aşımı 5 günlük zaman dilimi içindedir ve bunun ilk 4 gününde C Aktivitesi en düşük önceliğe sahiptir. Bu nedenle ilk denemede C aktivitesi 4 gün geciktirilmiş ve oluşan iş programı Şekil 3.13’te sunulmuştur. Ertelenecek aktivite için E sütununda aktivitenin bulunduğu ilgili satıra gecikme süresi girilir. D

sütununda aktivite süresi $Süre = Süre' + Gecikme$ olarak tanımlanmıştır. Burada $Süre'$ olarak tanımlanan aktivitenin kaynak kullanımını içeren süresi, Gecikme aktivitenin geciktirileceği süre ve $Süre$ ise aktivitenin toplam süresini ifade etmektedir. Aktivitenin kaynak kullanımına ne zaman başlayıp ne zaman tamamlanacağını belirten süreler M ve N sütununda Başlangıç ve Bitiş başlıkları altında hesaplanmıştır. E4 hücresine 4 rakamı girilerek C aktivitesi 4 gün geciktirilmiştir. C aktivitesi 4 gün geciktirilince M ve N sütunundaki Başlangıç ve Bitiş süreleri de değişmektedir. C aktivitesinin geciktirilmemiş hali olan $GEÇİKME_C = 0$ durumunda başlama ve tamamlanma zamanı sırası ile 0 ile 7. günler olmaktadır. C aktivitesi 4 gün geciktirilince aktivitenin başlama ve tamamlanma zamanı sırası ile 4 ile 11. günler olmaktadır. C aktivitesinin bolluğu olmadığı için ertelendikçe projenin tamamlanma süresi de uzamaktadır. Mevcut durumda projenin tamamlanma süresi 13 günden 17 güne uzamış ve kaynak kısıtlarının ihlali önlenemediği için aktivitelerin geciktirme sürecine devam edilecektir.

1	Baş	Bit	Aktivite	Süre	Gecikme	Yalın Süre	Kaynak	EOZ	GOZ	EOZ	GOZ	TB	Başlangıç	Bitiş	Mak İsp	12	Aktivite	Öncelik									
2	0	1	A	3	0	3	4	0	0	3	12	9	0	3			A	1 A									
3	1	3	B	5	0	5	6	3	12	17	17	9	3	8			B	2 B									
4	0	2	C	11	4	7	8	0	0	11	11	0	4	11			C	3 C									
5	2	3	D	6	0	6	7	11	11	17	17	0	11	17			D	4 D									
6																											
7	DN	EOZ	GOZ			12	12																				
8	0	0	0			30	30																				
9	1	3	12			56	56																				
10	2	11	11			42	42																				
11	3	17	17																								
12																											
13																											
14	Gün	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
15	A	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19		4	4	4	6	14	14	14	14	8	8	8	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
20																											
21	Çalışan Aktivite	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
22	A					B	B	B	B																		
23	B					C	C	C	C																		
24	C																										
25	D																										
26	Çalışma Önceliği																										
27	A					2	2	2	2																		
28	B					3	3	3	3																		
29	C																										
30	D																										
31	Erteleme	0	0	0	0	0	3	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	Aktivite					C	C	C	C																		

Şekil 3.13: C aktivitesinin 4 gün ertelenmesi sonucunda oluşan iş programı

Şekil 3.13'te sunulan iş programında kaynak kısıtlarının aşılmadığı dönemin 4 gün olduğu ve C aktivitesinin en düşük önceliğe sahip olduğu görülmektedir. Bu durumda ikinci denemede C aktivitesi 4 gün daha ertelenip C aktivitesinin ertelenme süresinin 4 günden 8 güne çıkarılması gerekmektedir. C aktivitesinin 8 gün ertelenmesi ile elde edilen iş programı Şekil 3.14'te sunulmuştur.

2	Baş	Bit	Aktivite	Süre	Gecikme	Yalın Süre	Kayın	EOZ	GOZ	EOZ	GOZ	TB	Başlangıç	Bitiş	Mak İpç	12	Aktivite	Öncelik	1 A								
3	0	1	A	3	0	3	4	0	0	0	16	13	0	3			A	1 A									
4	1	3	B	5	0	5	6	3	16	21	21	13	3	8			B	2 B									
5	0	2	C	15	0	15	16	0	0	15	15	0	8	15			C	3 C									
6	2	3	D	9	0	9	7	15	15	21	21	0	15	21			D	4 D									
7	DN	EOZ	GOZ					12	12																		
8	0	0	0					30	30																		
9	1	3	16					56	56																		
10	2	15	15					42	42																		
11	3	21	21																								
12																											
13																											
14	Gün	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
15	A	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19		4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20																											
21	Çalışan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
22	Aktivite																										
23	A																										
24	B																										
25	C																										
26	D																										
27	Çalışma																										
28	Önceliği																										
29	A																										
30	B																										
31	Erteleme	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Şekil 3.14: A-B-C-D alternatifinin ikinci deneme sonucunda oluşan iş programı

Örnek problem 1 için 6 farklı öncelik tayin etme seçeneği vardır. Birinci alternatif olan A-B-C-D öncelik sıralamasının çözümü yukarıda belirtildiği gibi hesaplanmıştır. İkinci alternatifte de 4 aktivitenin öncelik sırası değiştirilmiştir. Aşağıdaki tabloda belirtildiği gibi bu alternatifte B aktivitesinin öncelik sırası C aktivitesinden sonra gelmektedir. Burada B aktivitesi ertelenecektir. Alternatifler arasında en kısa süren öncelik sırası aranılan en iyi çözüm olacaktır. Şekil 3.15'te görüldüğü gibi 5 günlük periyot içerisinde kaynak aşımı vardır. İlk denemede B aktivitesi 5 gün ertelenmiştir. İlk deneme sonunda B aktivitesinin 5 gün geciktirilmiş olmasına rağmen kaynak kısıtlarının aşıldığı Şekil 3.15'te sunulan iş programında görülmektedir. Kaynak kısıtlarının aşılma süresi 5 gündür.

Bu durumda ikinci denemede B aktivitesi 5 gün daha ertelenmesi gerekmekte olduğu için B aktivitesinin ertelenme süresi 5 günden 10 güne çıkarılması gerekmektedir. B aktivitesinin 10 gün ertelenmesi ile elde edilen iş programı Şekil 3.16'da sunulmuştur.

Alternatifler arasında en kısa süren öncelik sırası aranılan en iyi çözüm olacaktır. Şekil 3.15'te görüldüğü gibi 5 günlük periyot içerisinde kaynak aşımı vardır. İlk denemede B aktivitesi 5 gün ertelenmiştir. İlk deneme sonunda B aktivitesinin 5 gün geciktirilmiş olmasına rağmen kaynak kısıtlarının aşıldığı Şekil 3.15'te sunulan iş programında görülmektedir. Kaynak kısıtlarının aşılma süresi 5 gündür.

Bu durumda ikinci denemede B aktivitesi 5 gün daha ertelenmesi gerekmekte olduğu için B aktivitesinin ertelenme süresi 5 günden 10 güne çıkarılması gerekmektedir. B aktivitesinin 10 gün ertelenmesi ile elde edilen iş programı Şekil 16'da sunulmuştur.

1	Baş	Bit	Aktivite	Süre	Geçikme	Yalın Süre	Kaynak	EOZİ	GOZİ	EOZİ	GOZİ	TB	Başlangıç	Bitiş	Mak İsp. 12	Aktivite	Öncelik	1 A									
2	0	1	A	3	0	3	4	0	0	3	3	0	0	3		A	1 A										
3	1	3	B	10	0	5	6	3	3	13	13	0	8	13		B	3 B										
4	0	2	C	7	0	7	8	0	0	7	7	0	0	7		C	2 C										
5	2	3	D	6	0	6	7	7	7	13	13	0	7	13		D	4 D										
6																											
7	DN	EOZİ	GOZİ					12	12																		
8	0	0	0					30	30																		
9	1	3	3					56	56																		
10	2	7	7					42	42																		
11	3	13	13																								
12																											
13																											
14	Gün	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
15	A	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	B	0	0	0	0	0	0	0	0	8	8	8	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
17	C	5	5	5	5	5	5	5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19		12	12	12	8	8	8	8	7	13	13	13	13	13	0	0	0	0	0	0	0	0	0	0	0	0	0
20																											
21	Çalışan																										
22	Aktivite	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
23	A																										
24	B									B	B	B	B	B													
25	C																										
26	D									D	D	D	D	D													
27	Çalışma																										
28	Önceliği																										
29	A																										
30	B										3	3	3	3	3												
31	C																										
32	D										4	4	4	4	4												
33	Ertelenecek	0	0	0	0	0	0	0	0	0	4	4	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0

Şekil 3.15: A-B-C-D alternatifinin birinci deneme sonucunda oluşan iş programı

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	
1	Baş	Bit	Aktivite	Süre	Geçikme	Yalın Süre	Kaynak	EOZİ	GOZİ	EOZİ	GOZİ	TB	Başlangıç	Bitiş		Mak İsp. 12			Aktivite	Öncelik								
2		0	1 A		3	0	3	4	0	0	3	3	0	0	3				A	1 A								
3		1	3 B		15	10	5	6	3	3	18	18	0	13	18				B	3 B								
4		0	2 C		7	0	7	8	0	0	7	12	4	0	7				C	2 C								
5		2	3 D		6	0	6	7	7	12	18	18	5	7	13				D	4 D								
6																												
7	DN	EOZİ	GOZİ					12	12																			
8		0		0				30	30																			
9		1	3	3				56	56																			
10		2	7	12				42	42																			
11		3	18	18																								
12																												
13																												
14	Gün		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
15	A	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
17	C	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	
18	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
19																												
20		12	12	12	8	8	8	8	8	7	7	7	7	7	7	6	6	6	6	6	6	0	0	0	0	0	0	
21	Çalışan																											
22	Aktivite	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	
23	A																											
24	B																											
25	C																											
26	D																											
27	Çalışma																											
28	Önceliği																											
29	A																											
30	B																											
31	Ertelenecek	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Şekil 3.16: A-C-B-D alternatifinin ikinci deneme sonucunda oluşan iş programı

Şekil 3.17’de sunulan iş programında üçüncü alternatifin ilk deneme sonucu gösterilmiştir. Tüm aktivitelerin öncelik sırası değiştirilmiştir. Ertelenecek aktivite olarak B belirlenmiştir. Aşağıda Şekil 3.17’deki iş programı tablosunda gösterildiği gibi kaynak aşımı 5 günlük dönem içesindedir ve ilk deneme de 5 gün ertelenmiştir. Kaynak kısıtlarının aşımı hala devam ettiği için ikinci denemede B aktivitesi 5 gün daha ertelenmesi gerekmektedir ve B aktivitesinin ertelenme süresi 5 günden 10 güne çıkarılmıştır. B aktivitesinin 10 gün ertelenmesi ile elde edilen iş programı Şekil 3.18’de sunulmuştur.

14	Gün	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
15	A	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	C	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
18	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19		12	12	12	8	8	8	8	7	13	13	13	13	13	0	0	0	0	0	0	0	0	0	0	0	0	0
20																											
21	Çalışan																										
22	Aktivite	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
23	A																										
24	B																										
25	C																										
26	D																										
27	Çalışma																										
28	Önceliği																										
29	A																										
30	B																										
31	C																										
32	D																										
33	Ertelenecek	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	Aktivite																										

Şekil 3.17: A-C-D-B alternatifinin birinci deneme sonucunda oluşan iş programı

Şekil 3.19’da sunulan iş programında dördüncü alternatifin ilk deneme sonucu gösterilmiştir. Tüm aktivitelerin öncelik sırası değiştirilmiştir. Ertelenecek aktivite olarak B belirlenmiştir. Şekil 3.19’da sunulan iş programı tablosunda gösterildiği gibi kaynak aşımı 5 günlük periyot içesindedir ve ilk deneme de 3 gün ertelenmiştir.

13	Gün	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
14	A	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	C	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
17	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18		12	12	12	8	8	8	8	7	7	7	7	7	7	6	6	6	6	6	6	6	6
19																						
20																						

Şekil 3.18: A-C-D-B alternatifinin ikinci deneme sonucunda oluşan iş programı

14	Gün	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
15	A	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	C	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
18	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19		12	12	12	8	8	8	8	14	13	13	13	13	7	7	7	7	7	7	7	7	7
20																						
21	Çalışan Aktivite	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
22																						

Şekil 3.19: C-D-A-B alternatifinin birinci deneme sonucunda oluşan iş programı

Şekil 3.19 ‘da görüldüğü gibi kaynak kısıtlarının aşımı hala devam ettiği için ikinci denemede B aktivitesi 7 gün daha ertelenmesi gerekmekte ve bu nedenle B aktivitesinin ertelenme süresi 3 günden 10 güne çıkarılmıştır. B aktivitesinin 10 gün ertelenmesi ile elde edilen iş programı Şekil 3.20’de sunulmuştur.

Şekil 3.21’de sunulan iş programında beşinci alternatifin ilk deneme sonucu gösterilmiştir. Tüm aktivitelerin öncelik sırası değiştirilmiştir. Ertelenecek aktivite B olarak belirlenmiştir. Şekil 3.21’de sunulan iş programı tablosunda gösterildiği gibi kaynak aşımı 5 günlük periyot içerisinde ve ilk deneme de 4 gün ertelenmiştir. Şekil 3.21’de görüldüğü gibi kaynak kısıtlarının aşımı hala devam ettiği için ikinci denemede B aktivitesi 6 gün daha ertelenmesi gerekmekte olduğu için B aktivitesinin ertelenme süresi 4 günden 10 güne çıkarılmıştır. B aktivitesinin 10 gün ertelenmesi ile elde edilen iş programı Şekil 3.22’de sunulmuştur.

14	Gün	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
15	A	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	C	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
18	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19		12	12	12	8	8	8	8	7	7	7	7	7	7	6	6	6	6	6	6
20																				

Şekil 3.20: C-D-A-B alternatifinin ikinci deneme sonucunda oluşan iş programı

14	Gün	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
15	A	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	C	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
18	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19		12	12	12	8	8	8	8	13	13	13	13	13	7	7	7	7	7	7
20																			

Şekil 3.21: C-A-B-D alternatifinin birinci deneme sonucunda oluşan iş programı

14	Gün	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
15	A	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	B	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	C	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
18	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19		12	12	12	8	8	8	8	7	7	7	7	7	7	6	6	6	6	6
20																			

Şekil 3.22: C-A-B-D alternatifinin ikinci deneme sonucunda oluşan iş programı

Şekil 3.23'te sunulan iş programında altıncı alternatifin ilk deneme sonucu gösterilmiştir. Tüm aktivitelerin öncelik sırası değiştirilmiştir. Ertelenecek aktivite olarak B belirlenmiştir. Aşağıda Şekil 3.23'te sunulan iş programı tablosunda gösterildiği gibi kaynak aşımı 5 günlük periyot içesindedir ve ilk deneme de 5 gün ertelenmiştir.

Şekil 3.23'te görüldüğü gibi kaynak kısıtlarının aşımı hala devam ettiği için ikinci denemede B aktivitesi 5 gün daha ertelenmesi gerekmekte olduğu için B aktivitesinin ertelenme süresi 5 günden 10 güne çıkarılmıştır. B aktivitesinin 10 gün ertelenmesi ile elde edilen iş programı Şekil 3.24'te sunulmuştur.

14	Gün	1	2	3	4	5	6	7	8	9	10	11	12	13
15	A	4	4	4	0	0	0	0	0	0	0	0	0	0
16	B	0	0	0	0	0	0	0	0	6	6	6	6	6
17	C	8	8	8	8	8	8	8	0	0	0	0	0	0
18	D	0	0	0	0	0	0	0	7	7	7	7	7	7
19		12	12	12	8	8	8	8	7	13	13	13	13	13

Şekil 3.23: C-A-D-B alternatifinin birinci deneme sonucunda oluşan iş programı

13	Gün	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
14	A	4	4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	B	0	0	0	0	0	0	0	0	0	0	0	0	0	6	6	6	6	6
16	C	8	8	8	8	8	8	8	0	0	0	0	0	0	0	0	0	0	0
17	D	0	0	0	0	0	0	0	7	7	7	7	7	7	0	0	0	0	0
18		12	12	12	8	8	8	8	7	7	7	7	7	7	6	6	6	6	6

Şekil 3.24: C-A-D-B alternatifinin ikinci deneme sonucunda oluşan iş programı

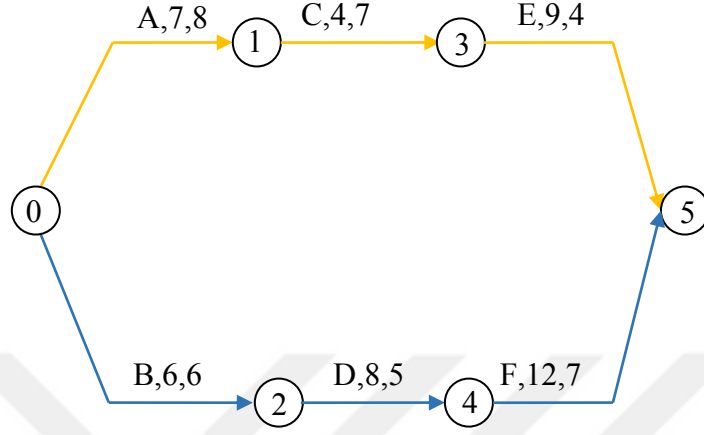
Çizelge 3.1'de belirtilen 6 öncelik sıralamasının tamamı denenmiş ve kaynak kısıtları dikkate alınarak oluşturulan çözümde proje süresi 18 gün olarak hesaplanmıştır.

Çizelge 3.1: Problem 1'in öncelik sıralama alternatifleri

Alternatif	Öncelik Sıralaması	Gün
A1	A-B-C-D	21
A2	A-C-B-D	18
A3	A-C-D-B	18
A4	C-A-B-D	18
A5	C-A-D-B	18
A6	C-D-A-B	18

3.2 Örnek Problem 2

Şebeke diyagramı Şekil 3.25 'te verilen 6 aktiviteli, 5 düğüm noktalı şebekenin sonsuz kaynak tanımı ile gerçekleştirilen çözümü Çizelge 3.2'de gösterilmiştir.



Şekil 3.25: Problem 2'nin şebeke diyagramı

Çizelge 3.2: Problem 2'nin kaynak kısıtı olmadan çözümü

Aktivite	Süre	Kaynak	EOZ _i	GOZ _i	EOZ _i	GOZ _i
A	7	8	0	0	13	25
B	6	6	0	0	6	6
C	4	7	13	25	17	29
D	8	5	6	6	25	25
E	9	4	17	29	38	38
F	12	7	25	25	38	38

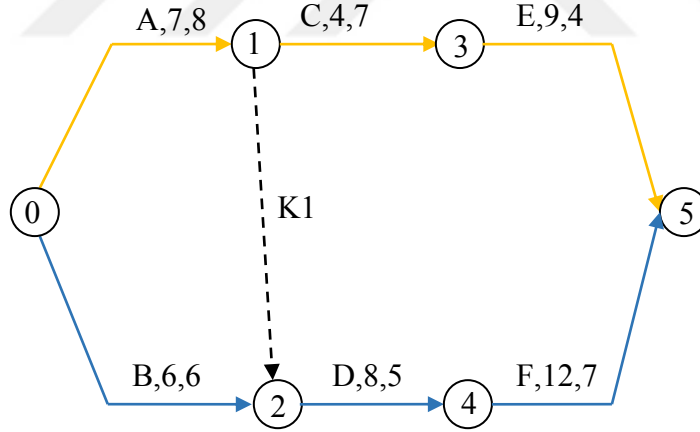
Problem 2, A-C-E ve B-D-F hatlarından oluşmaktadır. Belirtilen hatlardan kaç farklı aktivite önceliklerinin tayin edilebileceği hesaplanacaktır. Şebekenin çözümü için Çizelge 3.3'te sunulan öncelikler denenmiş ve elde edilen proje süreleri aynı Çizelgede sunulmuştur. Projenin en kısa tamamlanma süresi 37 gündür.

Çizelge 3.3: Aktivite önceliklerinin oluşturulması

Aktivite Öncelik Sırası	Proje Süresi	Aktivite Öncelik Sırası	Proje Süresi
ACEBDF	37	BADCFE	46
ACBEDF	37	BACDFE	46
ACBDEF	37	BACDEF	38
ACBDFE	37	BADCEF	46
ABCD FE	46	BDACEF	46
ABCDEF	38	BDACFE	46
ABDCEF	46	BDFACE	46
ABDCFE	46	BDAFCE	46
ABDFCE	46	BADFCE	46
ABCEDF	38	BACEDF	38

3.3 Örnek Problem 3

Şebeke diyagramı Şekil 3.26’da verilen 7 aktiviteli, 6 düğüm noktalı şebekenin sonsuz kaynak tanımı ile gerçekleştirilen çözümü Çizelge 3.4’te gösterilmiştir.



Şekil 3.26: Problem 3’ün şebeke diyagramı

Çizelge 3.4: Problem 3'ün kaynak kısıtı olmadan çözümü

Aktivite	Süre	Kaynak	Eozi	Gozi	Eozj	Goj
A	7	8	0	0	7	17
B	6	6	0	0	17	17
K1	0	0	7	17	17	17
C	4	7	7	17	11	25
D	8	5	17	17	25	25
E	9	4	11	25	37	37
F	12	7	25	25	37	37

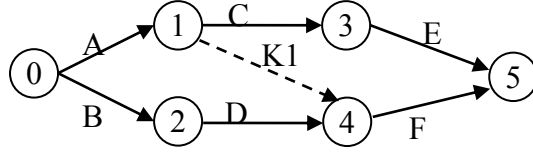
Problem 3, A-C-E, A-K1-D-F ve B-D-F hatlarından oluşmaktadır. Belirtilen hatlardan kaç farklı aktivite önceliklerinin tayin edilebileceği hesaplanacaktır. Şebekenin çözümü için Çizelge 3.5'te sunulan öncelikler denenmiş ve elde edilen proje süreleri aynı Çizelgede sunulmuştur. Projenin en kısa tamamlanma süresi 37 gündür.

Çizelge 3.5: Aktivite önceliklerinin oluşturulması

Aktivite Öncelik Sırası	Proje Süresi	Aktivite Öncelik Sırası	Proje Süresi
ABDCEF	46	BDAFCE	46
ABDCFE	46	ACEBDF	37
ABDFCE	46	ACBDEF	37
BADCEF	46	ABCDEF	38
BADCFE	46	BACDEF	38
BDACEF	46	ACBDFE	37
BDACFE	46	ABCD FE	46
ACBEDF	37	BACDFE	46
ABCEDF	38	BADFCE	46
BACEDF	38		

3.4 Örnek problem 4

Şebeke diyagramı Şekil 3.27'de verilen 7 aktiviteli, 6 düğüm noktalı şebekenin sonsuz kaynak tanımı ile gerçekleştirilen çözümü Tablo 5'te gösterilmiştir. Tablo 5'te gösterilen hesap cetvelinde bazı veriler girilerek en optimum proje süresi bulunması amaçlanmıştır.



Şekil 3.27: Problem 4'ün şebeke diyagramı

Çizelge 3.6: Problem 4'ün kaynak kısıtı olmadan çözümü

Aktivite	Süre	Kaynak	Eozi	Gozi	Eozj	Gozj
A	7	8	0	0	13	13
B	6	6	0	0	6	19
K1	0	0	13	13	21	34
C	4	7	13	13	37	37
D	8	5	6	19	21	34
E	9	4	37	37	46	46
F	12	7	21	34	46	46

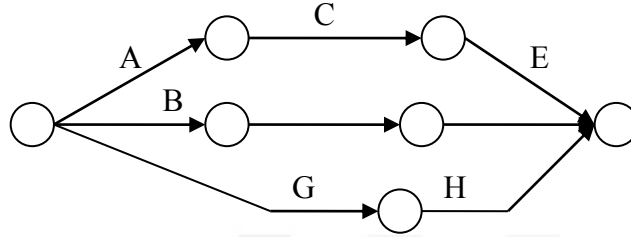
Problem 4, A-B-D ve C-E-F hatlarından oluşmaktadır. Belirtilen hatlardan kaç farklı aktivite önceliklerinin tayin edilebileceği hesaplanacaktır. Şebekenin çözümü için Çizelge 3.7'de sunulan öncelikler denenmiş ve elde edilen proje süreleri aynı Çizelgede sunulmuştur. Projenin en kısa tamamlanma süresi 37 gündür.

Çizelge 3.7: Aktivite önceliklerinin oluşturulması

Aktivite Öncelik Sırası	Proje Süresi	Aktivite Öncelik Sırası	Proje Süresi
ABDCEF	46	BDAFCE	46
ABDCFE	46	ACEBDF	37
ABDFCE	46	ACBDEF	37
BADCEF	46	ABCDEF	38
BADCFE	46	BACDEF	38
BDACEF	46	ACBDFE	37
BDACFE	46	ABCD FE	46
ACBEDF	37	BACDFE	46
ABCEDF	38	BADFCE	46
BACEDF	38		

3.5 Örnek problem 5 ve 6

Şebeke diyagramı Şekil 3.28 ve 3.29’da sunulan 8 aktiviteli, düğüm noktalı şebekenin sonsuz kaynak tanımı ile gerçekleştirilen çözümü Çizelge 3.8 ve 3.9’da gösterilmiştir. Çizelge 3.8 ve 3.9’da gösterilen hesap cetvelinde proje verileri girilerek en kısa proje süresi bulunması amaçlanmıştır.



Şekil 3.28: Problem 5’in şebeke diyagramı

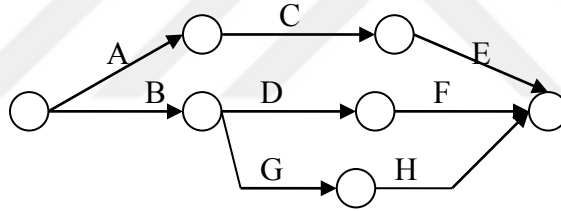
Çizelge 3.8: Problem 5’in kaynak kısıtı olmadan çözümü

Aktivite	Süre	Kaynak	Eozi	Gozi	Eozj	Goj	Öncelik
A	3	5	0	3	3	6	4
B	4	7	0	4	4	7	1
C	7	6	3	10	10	13	6
D	5	8	4	9	9	12	2
E	8	10	10	21	21	21	8
F	9	9	9	21	21	21	3
G	11	11	0	11	11	11	5
H	10	12	11	21	21	21	7

Problem 5, A-C-E, B-D-F ve G-H hatlarından oluşmaktadır. Belirtilen hatlardan kaç farklı aktivite önceliklerinin tayin edilebileceği hesaplanacaktır. Şebekenin çözümü için Çizelge 3.9’da sunulan öncelikler denenmiş ve elde edilen proje süreleri aynı Çizelgede sunulmuştur. Projenin en kısa tamamlanma süresi 53 gündür.

Çizelge 3.9: Problem 5'in aktivite öncelikleri ve süreleri

Alternatif	Gün	Alternatif	Gün	Alternatif	Gün
BDFGHACE	57	BDACFGHE	54	BDACFGEH	54
BDFGAHCE	57	BADCFGHE	54	BADCFGEH	54
BDFAGHCE	54	ABDCFGHE	54	ABDCFGEH	54
BDAFGHCE	54	BACDFGHE	53	BACDFGEH	53
BADFGHCE	54	ABCDGHE	53	ABCDGEH	53
ABDFGHCE	54	ACBDFGHE	53	ACBDFGEH	53
BDFGACHE	57	BDFGACEH	57	BDFACEGH	54
BDFAGCHE	54	BDFAGCEH	54	BDAFCEGH	54
BDAFGCHE	54	BDAFGCEH	54	BADFCEGH	54
BADFGHCE	54	BADFGCEH	54	ABDFCEGH	54
ABDFGCHE	54	ABDFGCEH	54	BDACFEGH	54
BDFACGHE	54	BDFACGEH	54	BADCFEGH	54
BDAFCGHE	54	BDAFCGEH	54	ABDCFEGH	54
BADFCGHE	54	BADFCGEH	54	BACDFEGH	53
ABDFCGHE	54	ABDFCGEH	54	ABCDFEGH	53
BACEDFGH	53	ABCEDFGH	53	ACBEDFGH	53
ACEBDFGH	53				



Şekil 3.29: Problem 6'nın şebeke diyagramı

Çizelge 3.10: Problem 6'nın kaynak kısıtı olmadan çözümü

Aktivite	Süre	Kaynak	Eozi	Gozi	Eozj	Gozj	Öncelik
A	3	5	0	0	3	10	6
B	4	7	0	0	4	4	1
C	7	6	3	10	10	17	7
D	5	8	4	4	9	16	2
E	8	10	10	17	25	25	8
F	9	9	9	16	25	25	3
G	11	11	4	4	15	15	4
H	10	12	15	15	25	25	5

Problem 6, A-C-E ve B-D-F-G-H hatlarından oluşmaktadır. Belirtilen hatlardan kaç farklı aktivite önceliklerinin tayin edilebileceği hesaplanacaktır. Şebekenin çözümü için

Çizelge 3.11’de sunulan öncelikler denenmiş ve elde edilen proje süreleri aynı Çizelgede sunulmuştur. Projenin en kısa tamamlanma süresi 53 gündür.

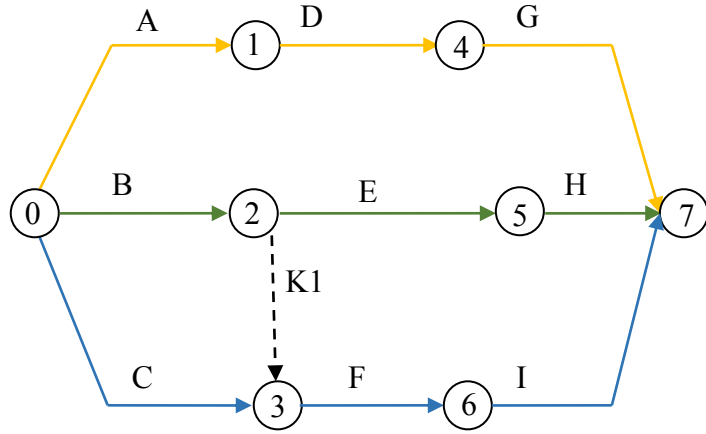
Çizelge 3.11: Problem 6’nın aktivite öncelikleri ve süreleri

Alternatif	Gün	Alternatif	Gün	Alternatif	Gün
BDFGHACE	54	BDACFGHE	54	BDACFGEH	54
BDFGAHCE	54	BADCFGHE	54	BADCFGEH	54
BDFAGHCE	54	ABDCFGHE	54	ABDCFGEH	54
BDAFGHCE	54	BACDFGHE	54	BACDFGEH	54
BADFGHCE	54	ABCDFGHE	54	ABCDFGEH	54
ABDFGHCE	54	ACBDFGHE	54	ACBDFGEH	54
BDFGACHE	54	BDFGACEH	54	BDFACEGH	54
BDFAGCHE	54	BDFAGCEH	54	BDAFCEGH	54
BDAFGCHE	54	BDAFGCEH	54	BADFCEGH	54
BADFGHCE	54	BADFGCEH	54	ABDFCEGH	54
ABDFGCHE	54	ABDFGCEH	54	BDACFEGH	54
BDFACGHE	54	BDFACGEH	54	BADCFEGH	54
BDAFCGHE	54	BDAFCGEH	54	ABDCFEGH	54
BADFCGHE	54	BADFCGEH	54	BACDFEGH	54
ABDFCGHE	54	ABDFCGEH	54	ABCDFEGH	54
ACBDFEGH	54	BDACEFGH	54	BADCEFGH	54
ABDCEFGH	54	BACDEFHG	54	ABCDEFHG	54
ACBDEFHG	54	ABCEDFGH	54	ACEBDFHG	54
BACEDFGH	54	ACBEDFGH	54		

Şekil 3.28 ve 3.29’da verilen şebeke de her bir aktivitenin değişen öncelik sırasına göre Çizelge 3.9 ve 3.11’de belirtilen kombinasyona tabi tutularak 560 ve 336 adet kombinasyon üretilecektir. Çizelge 3.9 ve 3.11’de ilk 56 alternatif hesaplanmıştır. 560 ve 336 alternatif arasından optimum proje süresi belirlenmiştir.

3.6 Örnek problem 7

Şebeke diyagramı Şekil 3.30’da sunulan 10 aktiviteli, 8 düğüm noktalı şebekenin sonsuz kaynak tanımı ile çözümü gerçekleştirilmiştir. Hesap cetvelinde proje verileri girilerek en kısa proje süresi bulunması amaçlanmıştır.



Şekil 3.30: Problem 7'nin şebeke diyagramı

Örnek problem 7'nin çözümünde Çizelge 3.12'de sunulan bilgiler hesap cetvelinde tanımlanarak öncelikler denenmiştir.

Çizelge 3.12: Problem 7'nin aktivite süreleri ve kaynak ihtiyaçları ile kaynak kısıtları kaldırıldığında elde edilen olay zamanları

Akt	Süre	Kaynak	Eozi	Gozi	Eozj	Gozej
A	7	5	0	0	7	7
B	6	4	0	0	6	12
C	4	6	0	0	6	12
D	8	7	7	7	15	15
E	5	3	6	12	11	17
F	6	5	6	12	12	18
G	6	4	15	15	21	21
H	4	3	11	17	21	21
I	3	2	12	18	21	21
K1	0	0	6	12	6	12

Problem 7, A-D-G, B-E-H, B-F-I ve C-F-I hatlarından oluşmaktadır. Belirtilen hatlardan kaç farklı aktivite önceliklerinin tayin edilebileceği hesaplanacaktır. İlk önce B-F-I ve C-F-I hatları arasında aktivite önceliği kombinasyonları üretilecektir. Belirtilen iki hattın öncelik tayini B-C-F-I ve C-B-F-I olarak belirlenir. Belirtilen öncelik sıralamaları B-E-H hattı ile eşleştirilerek aktivite öncelik kombinasyonları tanımlanır.

C-B-F-I öncelik sıralaması ile B-E-H hattı kombinasyona tabi tutulursa aşağıda belirtilen öncelik sıralamaları elde edilir.

C-B-E-H-F-I, C-B-E-F-H-I, C-B-E-F-I-H

C-B-F-E-H-I, C-B-F-E-I-H

C-B-F-I-E-H

$(3 + 2 + 1) = 6$ adet seçenekleri elde edilir.

B-C-F-I öncelik sıralaması ile B-E-H hattı kombinasyona tabi tutulursa aşağıda belirtilen öncelik sıralamaları elde edilir.

B-E-H-C-F-I, B-E-C-H-F-I, B-E-C-F-H-I, B-E-C-F-I-H

B-C-E-H-F-I, B-C-E-F-H-I, B-C-E-F-I-H

B-C-F-E-H-I, B-C-F-E-I-H

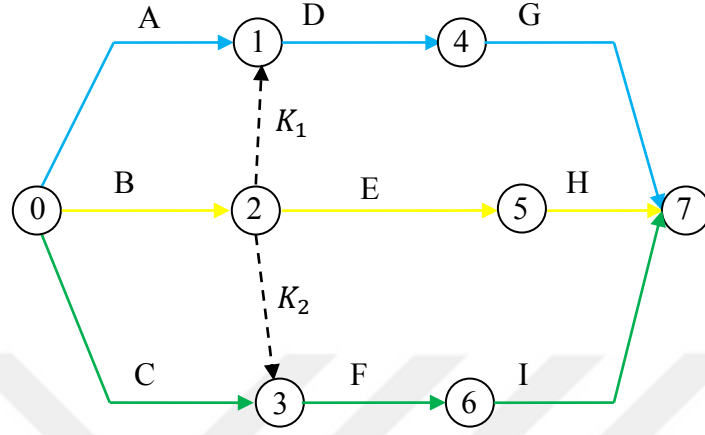
B-C-F-I-E-H

$(4 + 3 + 2 + 1) = 10$ adet seçenekleri elde edilir

İki hattan elde edilen öncelik sıralamaları A aktivitesi 7, B aktivitesi 1, C aktivitesi 4, D aktivitesi 8, E aktivitesi 2, F aktivitesi 5, G aktivitesi 9, H aktivitesi 3, I aktivitesi 6 olarak eklendiğinde toplam 16 adet aktivite öncelik kombinasyonu üretilebilir. Üretilen öncelik sıralamaları A-D-G hattı ile ayrıca kombinasyona tutulurlar. Bu şekilde Şekil 3.30'da verilen şebekede her bir aktivitenin değişen öncelik sırasına göre 1344 adet kombinasyon üretilmiştir. İlk kombinasyon BEHCFIADG aktivitelerinden oluşmaktadır. Aktivitelerin öncelik sırasına göre oluşturulmuş bu kombinasyona göre A aktivitesi 12 gün geciktirilerek proje süresi 33 gün olarak hesaplanmıştır. Geri kalan 1343 alternatif için öncelik sırasına göre kaynak çakışmasının olduğu aktiviteler öncelik sıralamasına göre geciktirilerek 276 alternatifte 24, 368 alternatifte 25, 97 alternatifte 26, 10 alternatifte 27, 311 alternatifte 28, 111 alternatifte 29, 53 alternatifte 30, 75 alternatifte 32, 43 alternatifte 33 gün olarak hesaplanmıştır. Optimum proje süresi 24 gün olarak belirlenmiştir. En iyi sonucu veren bu problemde 24 gün kombinasyonlarından biri ABCEDFHGI 'dir. Bu kombinasyonda ortaya çıkan aktivite erteleme süreleri A aktivitesi 0 , B aktivitesi 0, C aktivitesi 6 , D aktivitesi 3, E aktivitesi 1 , F aktivitesi 2 , G aktivitesi 0, H aktivitesi 6 , I aktivitesi 0 ve K1 aktivitesi 0 gün ertelenmiştir.

3.7 Örnek problem 8

Şebeke diyagramı Şekil 3.31’de sunulan 11 aktiviteli, 8 düğüm noktalı şebekenin sonsuz kaynak tanımı ile çözümü gerçekleştirilmiş ve proje süresi 21 gün bulunmuştur.



Şekil 3.31: Problem 8’in şebeke diyagramı

Örnek problem 8 ile ilgili aktivite süreleri ve kaynak ihtiyaçları Çizelge 3.13’te sunulmuştur. Ayrıca kaynak kısıtlarının gevşetilmiş haliyle elde edilen olay zamanları listelenmiştir.

Çizelge 3.13: Problem 8’in aktivite süreleri ve kaynak ihtiyaçları ile kaynak kısıtları kaldırıldığında elde edilen olay zamanları

Akt	Süre	Kaynak	EOZ _i	GOZ _i	EOZ _j	GOZ _j
A	4	5	0	0	5	11
B	5	7	0	0	5	7
C	7	9	0	0	7	7
D	6	7	5	11	11	17
E	3	7	5	7	8	16
F	8	9	7	7	15	15
G	4	6	11	17	21	21
H	5	8	8	16	21	21
I	6	9	15	15	21	21
K1	0	0	5	7	5	11
K2	0	0	5	7	7	7

Problem 8’de yer alan şebeke, A-D-G, B-D-G, B-E-H, B-F-I ve C-F-I hatlarından oluşmaktadır. İlk önce A-D-G ve B-D-G hatlarının aktivite önceliği kombinasyonları

üretilcektir. Belirtilen iki hattın kombinasyonu ile A-B-D-G ve B-A-D-G öncelik sıralamaları elde edilir. Belirtilen öncelik sıralamaları B-E-H hattı ile kombinasyona tutularak aktivite öncelik kombinasyonları tanımlanır.

A-B-D-G öncelik sıralaması ile B-E-H hattı kombinasyona tabi tutulursa aşağıdaki tabloda belirtilen öncelik sıralamaları elde edilir.

Çizelge 3.14: A-B-D-G öncelik sıralaması ile B-E-H hattı kombinasyonun öncelik sıralaması

A-B-D-G-E-H	A-B-D-E-H-G	A-B-E-D-H-G
A-B-D-E-G-H	A-B-E-H-D-G	
A-B-E-D-G-H		

Tabloda belirtildiği üzere toplamda $(3 + 2 + 1) = 6$ adet aktivite öncelik seçenekleri elde edilir. B-A-D-G öncelik sıralaması ile B-E-H hattı kombinasyona tabi tutulursa aşağıdaki tabloda belirtilen öncelik sıralamaları elde edilir.

Çizelge 3.15: B-A-D-G öncelik sıralaması ile B-E-H hattı kombinasyonunun öncelik sıralaması

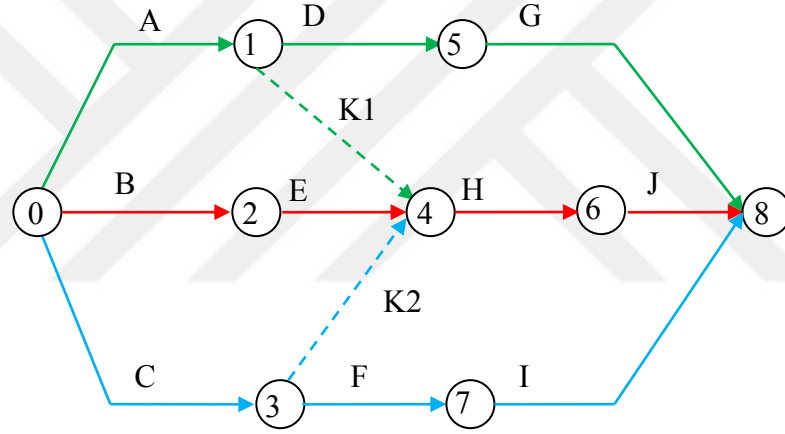
B-A-D-G-E-H	B-A-D-E-H-G	B-A-E-H-D-G	B-E-H-A-D-G
B-A-D-E-G-H	B-A-E-D-H-G	B-E-A-H-D-G	
B-A-E-D-G-H	B-E-A-D-H-G		
B-E-A-D-G-H			

Tabloda belirtildiği üzere toplamda $(4 + 3 + 2 + 1) = 10$ adet aktivite öncelik seçenekleri elde edilir. A-B-D-G ve B-A-D-G öncelik sıralamaları B-E-H hattı ile kombinasyona tabi tutulduğunda toplam 16 adet aktivite öncelik sıralaması üretilir. Üretilen kombinasyonlar C-F-I hattı ile ayrıca kombinasyona tutulurlar. Bu şekilde Şekil 3.31’de verilen şebeke de her bir aktivitenin değişen öncelik sırasına göre A aktivitesi 4, B aktivitesi 1, C aktivitesi 7, D aktivitesi 5, E aktivitesi 2, F aktivitesi 8, G aktivitesi 6, H aktivitesi 3, I aktivitesi 9 olarak 1160 adet kombinasyon üretilmiştir. İlk kombinasyon CABFIDGEH aktivitelerinden oluşmaktadır. Aktivitelerin öncelik sırasına göre oluşturulmuş bu kombinasyona göre C aktivitesi 23 gün, D aktivitesi 8 gün geciktirilerek proje süresi 44 gün olarak hesaplanmıştır. Geri kalan 1159 alternatif için öncelik sırasına göre kaynak çakışmasının olduğu aktiviteler öncelik sıralamasına göre geciktirilerek 1 alternatifte 33, 16 alternatifte 34, 8 alternatifte 35, 27 alternatifte 36, 60 alternatifte 40, 39

alternatifte 41, 33 alternatifte 42, 56 alternatifte 43, 577 alternatifte 44, 67 alternatifte 45,21 alternatifte 46, 23 alternatifte 47,140 alternatifte 48, 2 alternatifte 49, 2 alternatifte 51 gün olarak hesaplanmıştır. Optimum proje süresi 33 gün olarak belirlenmiştir. En iyi sonucu veren bu problemde 33 gün kombinasyonu BCAFDIEHG'dir. Bu kombinasyonda ortaya çıkan aktivite erteleme süreleri A aktivitesi 12, B aktivitesi 0 ,C aktivitesi 5, D aktivitesi 8, E aktivitesi 31, F aktivitesi 4, G aktivitesi 14 ve I aktivitesi 6, K1 aktivitesi 0 ve K2 aktivitesi 0 gün ertelenmiştir.

3.8 Örnek problem 9

Şebeke diyagramı Şekil 3.32'de sunulan 12 aktiviteli, 8 düğüm noktalı şebekenin sonsuz kaynak tanımı ile çözümü gerçekleştirilmiş ve proje süresi 23 gün bulunmuştur.



Şekil 3.32: Problem 9'un şebeke diyagramı

Örnek problem 9'un çözümünde Çizelge 3.16'da sunulan bilgiler hesap cetvelinde tanımlanarak öncelikler denenmiştir.

Çizelge 3.16: Problem 9'un aktivite süreleri ve kaynak ihtiyaçları

Akt	Süre	Kaynak	EOZ _i	GOZ _i	EOZ _j	GOZ _j
A	5	7	0	0	5	13
B	6	8	0	0	6	6
C	4	5	0	0	4	8
D	3	5	5	13	8	19
E	7	8	6	6	13	13
F	7	9	4	8	11	15
G	4	6	8	19	23	23
H	5	6	13	13	18	18
J	5	8	18	18	23	23
I	8	9	11	15	23	23
K1	0	0	5	13	13	13
K2	0	0	4	8	13	13

Problem 9, A-D-G, A-K1-H-J, B-E-H-J, C-K2-H-J ve C-F-I hatlarından oluşmaktadır. Belirtilen hatlardan kaç farklı aktivite önceliklerinin tayin edilebileceği hesaplanacaktır. A-D-G, A-K1-H-J ve B-E-H-J hatları kombinasyona tabi tutulacaktır. A aktivitesi B-E aktiviteleri ile A-B-E-H-J, B-A-E-H-J ve B-E-A-H-J öncelik sıraları oluşturulur. Elde edilen öncelik sıralamaları A-D-G hattının kalan kısmı olan D-G hattı ile kombinasyona tabi tutularak 3 hattan elde edilebilecek öncelik sıraları belirlenir.

A-B-E-H-J öncelik sıralaması ile D-G hattı kombinasyona tabi tutulursa aşağıdaki tabloda belirtilen öncelik sıralamaları elde edilir.

Çizelge 3.17: A-B-E-H-J öncelik sıralaması ile D-G hattı kombinasyonunun tabi öncelik sıralaması

A-B-E-H-J-D-G	A-B-E-H-D-J-G	A-B-E-D-H-J-G	A-B-D-E-H-J-G
A-B-E-H-D-G-J	A-B-E-D-H-G-J	A-B-D-E-H-G-J	
A-B-E-D-G-H-J	A-B-D-E-G-H-J		
A-B-D-G-E-H-J			

Yukarıdaki tabloda belirtildiği üzere toplamda $(4 + 3 + 2 + 1) = 10$ adet aktivite öncelik seçenekleri elde edilir.

B-A-E-H-J öncelik sıralaması ile D-G hattı kombinasyona tabi tutulursa aşağıdaki tabloda belirtilen öncelik sıralamaları elde edilir.

Çizelge 3.18: B-A-E-H-J öncelik sıralaması ile D-G hattı kombinasyonunun öncelik sıralaması

B-A-E-H-J-D-G	B-A-E-H-D-J-G	B-A-E-D-H-J-G	B-A-D-E-H-J-G
B-A-E-H-D-G-J	B-A-E-D-H-G-J	B-A-D-E-H-G-J	
B-A-E-D-G-H-J	B-A-D-E-G-H-J		
B-A-D-G-E-H-J			

Yukarıdaki tabloda belirtildiği üzere toplamda $(4 + 3 + 2 + 1) = 10$ adet aktivite öncelik seçenekleri elde edilir. B-E-A-H-J öncelik sıralaması ile D-G hattı kombinasyona tabi tutulursa aşağıdaki tabloda belirtilen öncelik sıralamaları elde edilir.

Çizelge 3.19: B-E-A-H-J öncelik sıralaması ile D-G hattı kombinasyonunun öncelik sıralaması

B-E-A-H-J-D-G	B-E-A-H-D-J-G	B-E-A-D-H-J-G
B-E-A-H-D-G-J	B-E-A-D-H-G-J	
B-E-A-D-G-H-J		

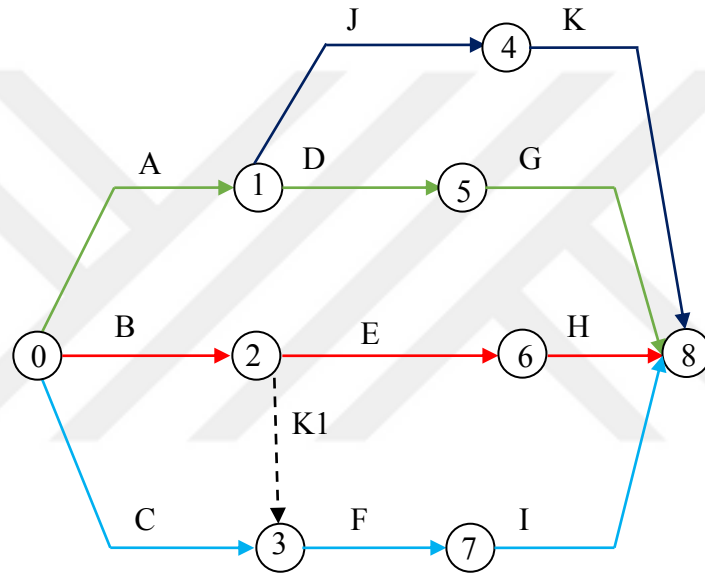
Yukarıdaki tabloda belirtildiği üzere toplamda $(3 + 2 + 1) = 6$ adet aktivite öncelik seçenekleri elde edilir.

A aktivitesi 3, B aktivitesi 1, C aktivitesi 6 , D aktivitesi 4, E aktivitesi 2, F aktivitesi 9, G aktivitesi 5, H aktivitesi 7, J aktivitesi 8, I aktivitesi 10 olarak öncelikler elde edilmiştir.Üç kombinasyondan toplam 26 adet aktivite öncelik sıralaması üretilebilir. Üretilen kombinasyonlar C-F-I hattı ile ayrıca kombinasyona tutulurlar. Bu şekilde Şekil 3.32’de verilen şebeke de her bir aktivitenin değişen öncelik sırasına göre 3380 adet kombinasyon üretilmiştir. İlk kombinasyon CFIABEHJDG aktivitelerinden oluşmaktadır. Aktivitelerin öncelik sırasına göre oluşturulmuş bu kombinasyona göre A aktivitesi 13 gün, C aktivitesi 13 gün, F aktivitesi 12 gün, J aktivitesi 2 gün geciktirilerek proje süresi 44 gün olarak hesaplanmıştır. Geri kalan 3379 alternatif için öncelik sırasına göre kaynak çakışmasının olduğu aktiviteler öncelik sıralamasına göre geciktirilerek 61 alternatifte 36, 57 alternatifte 37, 165 alternatifte 38, 163 alternatifte 39, 117 alternatifte 40, 113 alternatifte 41, 243 alternatifte 42, 185 alternatifte 43, 105 alternatifte 44, 229 alternatifte 45, 447 alternatifte 46, 296 alternatifte 47,38 alternatifte 48, 166 alternatifte 49, 713 alternatifte 50, 151 alternatifte 51, 3 alternatifte 52, 3 alternatifte 53, 125 alternatifte 54 gün olarak hesaplanmıştır. Optimum proje süresi 36 gün olarak belirlenmiştir. En iyi sonucu veren bu problemde 36 gün kombinasyonlarından biri CABFEIHJDG’dır. Bu

kombinasyonda ortaya çıkan aktivite erteleme süreleri A aktivitesi 0, B aktivitesi 5 , C aktivitesi 0, D aktivitesi 28, E aktivitesi 7 , F aktivitesi 7 , G aktivitesi 7, H aktivitesi 8, J aktivitesi 0, I aktivitesi 7, K1 aktivitesi 0 ve K2 aktivitesi 0 gün olarak gerçekleşmiştir.

3.9 Örnek problem 10

Şebeke diyagramı Şekil 3.33'te sunulan 12 aktiviteli, 8 düğüm noktalı şebekenin sonsuz kaynak tanımı ile çözümü gerçekleştirilmiş ve proje süresi 16 gün bulunmuştur.



Şekil 3.33: Problem 10'un şebeke diyagramı

Örnek problem 10'un çözümünde Çizelge 3.20'de sunulan aktivite süreleri ve kaynak ihtiyaçları hesap cetvelinde tanımlanarak öncelikler denenmiştir.

Çizelge 3.20: Problem 10'un aktivite süreleri ve kaynak ihtiyaçları ile kaynak kısıtları kaldırıldığında elde edilen olay zamanları

Akt	Süre	Kaynak	EOZ _i	GOZ _i	EOZ _j	GOZ _j
A	4	7	0	0	4	4
B	6	8	0	0	6	7
C	5	6	0	0	6	7
J	7	8	4	4	11	12
D	6	7	4	4	10	10
E	5	8	6	7	11	13
F	4	6	6	7	10	11
K	4	6	11	12	16	16
G	6	9	10	10	16	16
H	3	6	11	13	16	16
I	5	8	10	11	16	16
K1	0	0	6	7	6	7

Problem 10, A-D-G, A-J-K, B-E-H, B-F-I ve C-F-I hatlarından oluşmaktadır. Belirtilen hatlardan kaç farklı biçimde aktivitelere önceliklerin tayin edilebileceği hesaplanacaktır. İlk önce B-F-I ve C-F-I hatlarının aktivite önceliği kombinasyonları üretilmiştir. Belirtilen iki hatta F ve I aktiviteleri ortak olduğu için sadece B ve C aktiviteleri arasında kombinasyon yapılmıştır. Bu hatlardan oluşturulan öncelik sıralamaları B-C-F-I ve C-B-F-I olarak belirlenir. Belirtilen öncelik sıralamaları önce E-H hattı ile sonrasında A-D-G ve A-J-K hatları ile kombinasyona tabi tutularak aktivite öncelik kombinasyonları oluşturulur.

B-C-F-I öncelik sıralaması ile E-H hattı kombinasyona tabi tutulursa aşağıdaki tabloda belirtilen öncelik sıralamaları elde edilir.

Çizelge 3.21: B-C-F-I öncelik sıralaması ile E-H hattı kombinasyonunun öncelik sıralaması

B-C-F-I-E-H	B-C-F-E-I-H	B-C-E-F-I-H	B-E-C-F-I-H
B-C-F-E-H-I	B-C-E-F-H-I	B-E-C-F-H-I	
B-C-E-H-F-I	B-E-C-H-F-I		
B-E-H-C-F-I			

B-C-F-I öncelik sıralaması ile E-H hattı kombinasyona tabi tutulduğunda E aktivitesi B aktivitesinden önce yapılamayacağı için 4 farklı konumda yer alabilecektir. H aktivitesi ise E aktivitesinden önce başlayamayacağı için H aktivitesinin konumuna göre

farklı biçimde konumlandırılabilir. E aktivitesi B aktivitesi ile C aktivitesi arasında bulunduğunda H aktivitesi yukarıdaki tabloda görüldüğü üzere 4 farklı biçimde yer alabilmektedir. E aktivitesinin önceliği bir azaltıldığında H aktivitesine 3 farklı öncelik tanımlanabilmektedir. Bu durumda oluşturulabilecek toplam öncelik sıralama sayısı toplamda $(4+3 + 2 + 1) = 10$ adet olarak hesaplanır.

C-B-F-I öncelik sıralaması ile E-H hattı kombinasyona tabi tutulursa aşağıdaki tabloda belirtilen öncelik sıralamaları elde edilir.

Çizelge 3.22: C-B-F-I öncelik sıralaması ile E-H hattı kombinasyonunun öncelik sıralaması

C-B-F-I-E-H	C-B-F-E-I-H	C-B-E-F-I-H
C-B-F-E-H-I	C-B-E-F-H-I	
C-B-E-H-F-I		

E aktivitesi B aktivitesi tamamlandıktan sonra başlayabileceği için E aktivitesine 3 farklı öncelik tayin edilebilmektedir. H aktivitesi ise E aktivitesi tamamlandıktan sonra başlayabileceği için H aktivitesine atanabilecek öncelik E aktivitesinin konumuna bağlıdır. Bu durumda toplamda $(3 + 2 + 1) = 6$ adet öncelik sıralaması elde edilir.

Şebekede yer alan diğer 2 hat olan A-D-G ve J-K hatları kombinasyonlaştırılırsa aşağıdaki tabloda sunulan öncelik sıraları elde edilir. J aktivitesi A aktivitesi tamamlandıktan sonra başlayabileceği için 3 farklı konumda yer alabilecektir. K aktivitesi ise J aktivitesi tamamlandıktan sonra başlayabileceği için K aktivitesine atanabilecek öncelikler J aktivitesinin önceliğine bağlıdır.

Çizelge 3.23: A-D-G ve J-K hattı kombinasyonu

A-D-G-J-K	A-D-J-G-K	A-J-D-G-K
A-D-J-K-G	A-J-D-K-G	
A-J-K-D-G		

Tabloda gösterildiği üzere toplamda $(3 + 2 + 1) = 6$ adet öncelik sıralaması elde edilir.

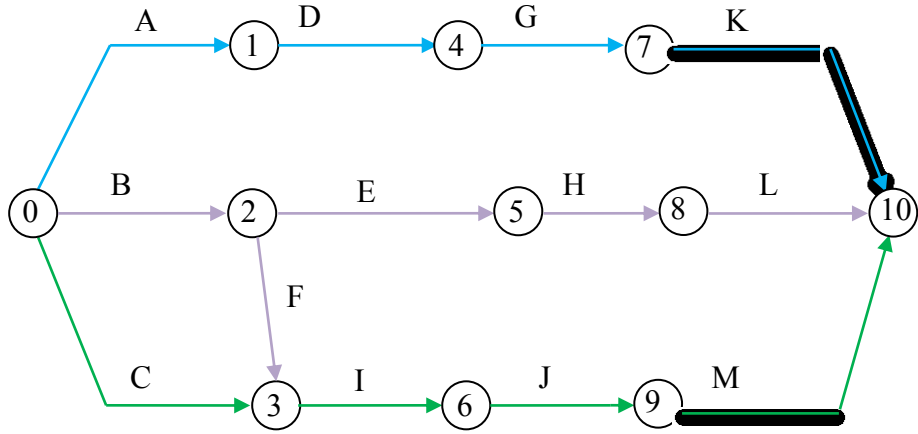
B-C-F-I ve E-H hatlarının kombinasyonu ile C-B-F-I ve E-H hatlarının kombinasyonlarından elde edilen farklı öncelik sıralamaları toplanabilir. Çünkü elde edilen öncelik sıralamaları E-H ortak hattı ile yapılan kombinasyon sonucu elde edilmiştir. A-D-G-J-K ile B-C-E-F-H-I hatları birbirine paraleldir. Bu nedenle bu hatlardan oluşturulan

öncelik sıralamaları kendi içinde tekrar kombinatöryel biçimde sıralanabilmektedir. Bu şekilde Şekil 3.33'te verilen şebeke de her bir aktivitenin değişen öncelik sırasına göre 44352 adet kombinasyon üretilmiştir.

İlk kombinasyon ADJGKBCFIEH aktivitelerinden oluşmaktadır. Aktivitelerin öncelik sırasına göre; A aktivitesi 1, B aktivitesi 6, C aktivitesi 7, D aktivitesi 2, E aktivitesi 10, F aktivitesi 8, K aktivitesi 5, G aktivitesi 4, H aktivitesi 11 , I aktivitesi 9 olarak oluşturulmuş bu kombinasyona göre A aktivitesi 5, B aktivitesi 32, C aktivitesi 38, J aktivitesi 6, E aktivitesi 11, K aktivitesi 6, G aktivitesi 7 gün geciktirilerek proje süresi 57 gün olarak hesaplanmıştır. Geri kalan 44351 alternatif için öncelik sırasına göre kaynak çakışmasının olduğu aktiviteler öncelik sıralamasına göre geciktirilerek 174 alternatifte 31, 58 alternatifte 32, 477 alternatifte 33, 1266 alternatifte 34, 2739 alternatifte 35, 4343 alternatifte 36, 5590 alternatifte 37, 4779 alternatifte 38, 4037 alternatifte 39, 3756 alternatifte 40, 3189 alternatifte 41, 2868 alternatifte 42, 2492 alternatifte 43, 1527 alternatifte 44, 1164 alternatifte 45, 1400 alternatifte 46, 875 alternatifte 47, 874 alternatifte 48, 420 alternatifte 49, 267 alternatifte 50, 736 alternatifte 51, 882 alternatifte 52, 84 alternatifte 53, 94 alternatifte 54, 243 alternatifte 55, 9 alternatifte 56, 8 alternatifte 57, 1 alternatifte 60 gün olarak hesaplanmıştır. Optimum proje süresi 48 gün olarak belirlenmiştir. En iyi sonucu veren bu problemde 48 gün kombinasyonlarından biri ADJGKCBEFIH'dir. Bu kombinasyonda ortaya çıkan aktivite erteleme süreleri A aktivitesi 0, B aktivitesi 28, C aktivitesi 23, J aktivitesi 6, D aktivitesi 0, E aktivitesi 0, F aktivitesi 5, K aktivitesi 6, G aktivitesi 7, H aktivitesi 0, I aktivitesi 0, K1 aktivitesi 0 gün ertelenmiştir.

3.10 Örnek problem 11

Şebeke diyagramı Şekil 3.34'te sunulan 13 aktiviteli, 10 düğüm noktalı şebekenin sonsuz kaynak tanımı ile çözümü gerçekleştirilmiş ve proje süresi 27 gün bulunmuştur.



Şekil 3.34: Problem 11'in şebeke diyagramı

Çizelge 3.24: Problem 11'in aktivite süreleri ve kaynak ihtiyaçları.

Akt	Süre	Kaynak	Eozi	Gozi	Eozj	Gozej	Öncelik
A	5	9	0	0	5	18	1
B	10	6	0	0	10	10	5
C	6	7	0	0	15	15	7
D	7	9	5	18	12	25	2
E	6	7	10	10	16	18	11
F	5	6	10	10	15	15	6
G	3	6	12	25	15	28	3
H	8	9	16	18	24	26	12
I	4	5	15	15	19	19	8
J	6	9	19	19	25	25	9
K	5	8	15	28	33	33	4
L	7	6	24	26	33	33	13
M	8	6	25	25	33	33	10

Problem 11, A-D-G-K, B-E-H-L, B-F-I-J-M ve B-C-I-J-M hatlarından oluşmaktadır. Belirtilen hatlardan kaç farklı aktivite önceliklerinin tayin edilebileceği hatların kombinleştirilmesi ile hesaplanacaktır.

Öncelik tayinine B-F-I-J-M ve C-F-I-J-M hatlarının aktivite önceliği kombinasyonları üretilerek başlanmıştır. Hatlarda F-I-J-M aktiviteleri ortak olduğu için sadece B ve C aktiviteleri arasında farklı öncelik verilebilmektedir. Belirtilen iki hattın öncelik tayini B-F-C-I-J-M ve B-C-F-I-J-M öncelik sıralamaları olarak belirlenir. Belirtilen

öncelik sıralamaları E-H-L hattı ile kombinasyona tabi tutularak aktivite öncelik kombinasyonları tanımlanır.

B-F-C-I-J-M ile E-H-L hatlarının kombinasyonundan elde edilen öncelik sıralamaları aşağıdaki tabloda sunulmuştur.

Çizelge 3.25: B-F-C-I-J-M ile E-H-L hatlarının kombinasyonundan elde edilen öncelik sıralamaları

B-F-C-I-J-	B-F-C-I-J-E-	B-F-C-I-E-J-	B-F-C-E-I-J-	B-F-E-C-I-J-	B-E-F-C-I-J-
M-E-H-L	M-H-L	M-H-L	M-H-L	M-H-L	M-H-L
B-F-C-I-J-E-	B-F-C-I-E-J-	B-F-C-E-I-J-	B-F-E-C-I-J-	B-E-F-C-I-J-	
H-M-L	H-M-L	H-M-L	H-M-L	H-M-L	
B-F-C-I-E-	B-F-C-E-I-	B-F-E-C-I-	B-E-F-C-I-		
H-J-M-L	H-J-M-L	H-J-M-L	H-J-M-L		
B-F-C-E-H-	B-F-E-H-C-	B-E-F-H-C-			
I-J-M-L	I-J-M-L	I-J-M-L			
B-F-E-H-C-	B-E-F-H-C-				
I-J-M-L	I-J-M-L				
B-E-H-F-C-					
I-J-M-L					

Tablonun birinci satırında E aktivitesinin önceliği sağa doğru ilerlendikçe arttırılmaktadır. E aktivitesi B aktivitesi tamamlandıktan sonra başlayabildiği için 6 farklı öncelik tanımlanabilmektedir. İlk satırda H ve L aktivitelerine olabilecek en düşük öncelik atanmıştır. İkinci satırda ise L aktivitesinin önceliği değiştirilmemiş, fakat H aktivitesinin önceliği 1 arttırılmıştır. E aktivitesi H aktivitesine göre daha yüksek öncelik değerine sahip olması gerektiği için bu aşamada E aktivitesine 5 farklı öncelik değeri atanabilmektedir. İkinci satırda sağa doğru ilerlendikçe E aktivitesinin önceliği artmaktadır. Üçüncü satırda ise L aktivitesi sabit tutulmuş ve H aktivitesinin önceliği 1 arttırılmıştır. Bu durumda E aktivitesine 4 farklı öncelik değeri atanabilmektedir. L aktivitesinin önceliği sabit tutularak toplamda $(6+5+4+3+2+1) = 21$ adet öncelik kombinasyonu elde edilir.

B-C-F-I-J-M ve E-H-L hatları kombinasyona tabi tutulursa aşağıdaki tabloda gösterilen öncelikler elde edilir.

Çizelge 3.26: B-C-F-I-J-M ve E-H-L hatları kombinasyonu öncelik sıralaması

B-C-F-I-J-	B-C-F-I-J-E-	B-C-F-I-E-J-	B-C-F-E-I-J-	B-C-E-F-I-J-	B-E-C-F-I-J-
M-E-H-L	M-H-L	M-H-L	M-H-L	M-H-L	M-H-L
B-C-F-I-J-E-	B-C-F-I-E-J-	B-C-F-E-I-J-	B-C-E-F-I-J-	B-E-C-F-I-J-	
H-M-L	H-M-L	H-M-L	H-M-L	H-M-L	
B-C-F-I-E-	B-C-F-E-I-	B-C-E-F-I-	B-E-C-F-I-		
H-J-M-L	H-J-M-L	H-J-M-L	H-J-M-L		
B-C-F-E-H-	B-C-E-H-F-	B-E-C-H-F-			
I-J-M-L	I-J-M-L	I-J-M-L			
B-C-E-H-F-	B-E-C-H-F-				
I-J-M-L	I-J-M-L				
B-E-H-C-F-					
I-J-M-L					

Bu kombinasyonda sadece C ve F aktivitelerinin yeri değişmiştir ve bu durum E-H-L hattı ile oluşacak kombinasyonları etkilememektedir. Bu seçenekte de L aktivitesine en düşük öncelik atandığında $(6+5+ 4 + 3 + 2 + 1) = 21$ adet farklı öncelik sıralaması elde edilir. L aktivitesinin önceliği arttırıldığı durumda ise $15 + 10 + 6 +3 + 1 = 35$ adet farklı kombinasyon oluşturulur. Böylece bu hattan da toplamda 56 adet farklı kombinasyon oluşturulur.

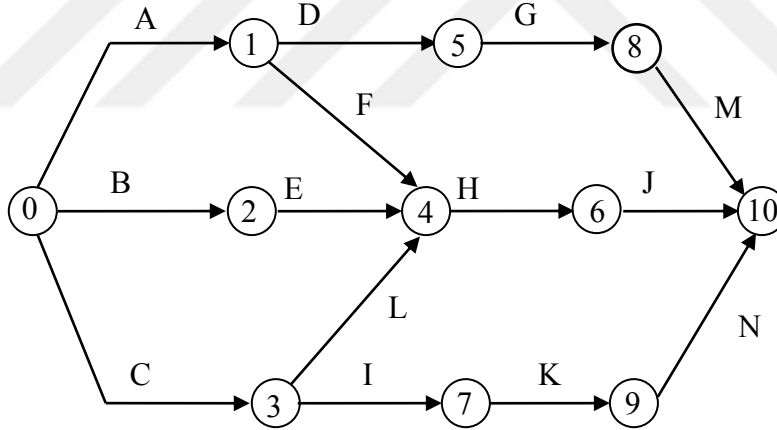
B-F-C-I-J-M ile E-H-L ve B-C-F-I-J-M ile E-H-L hatlarından toplamda 112 adet farklı aktivite önceliği oluşturulabilir.

Aktivitelerin öncelik sırasına göre; A aktivitesi 1, B aktivitesi 5, C aktivitesi 7, D aktivitesi 2, E aktivitesi 11, F aktivitesi 6, G aktivitesi 3, H aktivitesi 12, I aktivitesi 8 , J aktivitesi 9,K aktivitesi 4, L aktivitesi 13, M aktivitesi 10 olarak oluşturulan aktivite öncelik kombinasyonları A-D-G-K hattı ile ayrıca kombinasyona tutulurlar. Bu şekilde Şekil 3.34’te verilen şebeke de her bir aktivitenin değişen öncelik sırasına göre 105105 adet kombinasyon üretilmiştir. İlk kombinasyon ADGKBFCIJMEHL aktivitelerinden oluşmaktadır. Aktivitelerin öncelik sırasına göre oluşturulmuş bu kombinasyona göre B aktivitesi 20, C aktivitesi 29 ve E aktivitesi 29 gün geciktirilerek proje süresi 74 gün olarak hesaplanmıştır. Geri kalan 105104 alternatif için öncelik sırasına göre kaynak çakışmasının olduğu aktiviteler öncelik sıralamasına göre geciktirilerek 336 alternatifte 43, 1080 alternatifte 45, 2465 alternatifte 46, 1088 alternatifte 47, 7709 alternatifte 48, 5028 alternatifte 49, 2853 alternatifte 50, 15581 alternatifte 51, 9612 alternatifte 52,5250 alternatifte 53, 10034 alternatifte 54, 6448 alternatifte 55, 4773 alternatifte 56, 5637

alternatifte 57, 5529 alternatifte 58, 7403 alternatifte 59, 2932 alternatifte 60, 1680 alternatifte 61, 2748 alternatifte 62, 1433 alternatifte 63, 1461 alternatifte 64,412 alternatifte 65,1272 alternatifte 66, 1338 alternatifte 67,80 alternatifte 68, 178 alternatifte 69,304 alternatifte 70, 290 alternatifte 71, 16 alternatifte 72,17 alternatifte 73, 118 alternatifte 74 gün olarak hesaplanmıştır. Optimum proje süresi 43 gün olarak belirlenmiştir.En iyi sonucu veren bu problemde 43 gün kombinasyonlarından biri ABDCEFGKIJMHL'dir. Bu kombinasyonda ortaya çıkan aktivite erteleme süreleri A aktivitesi 0, B aktivitesi 5, C aktivitesi 16, D aktivitesi 4, E aktivitesi 13, F aktivitesi 10, G aktivitesi 6, H aktivitesi 0, I aktivitesi 0, J aktivitesi 0, K aktivitesi 0, L aktivitesi 0 ve M aktivitesi 0 gün ertelenmiştir.

3.11 Örnek problem 12

Şebeke diyagramı Şekil 3.35'te sunulan 14 aktiviteli, 10 düğüm noktalı şebekenin sonsuz kaynak tanımı ile çözümü gerçekleştirilmiş ve proje süresi 23 gün bulunmuştur.



Şekil 3.35: Problem 12'nin şebeke diyagramı

Örnek problem 12 ile ilgili aktivite süreleri ve kaynak ihtiyaçları Çizelge 3.27 'de sunulmuştur. Ayrıca kaynak kısıtlarının gevşetilmiş haliyle elde edilen olay zamanları listelenmiştir.

Çizelge 3.27: Problem 12'nin aktivite süreleri ile kaynak ihtiyaçları

Akt	Süre	Kaynak	Eozi	Gozi	Eozj	Goj
A	4	7	0	0	4	7
B	5	8	0	0	5	10
C	6	9	0	0	6	6
D	5	7	4	7	9	12
E	4	6	5	10	13	14
I	4	8	6	6	10	10
F	4	5	4	7	13	14
L	7	9	6	6	13	14
G	6	7	9	12	15	18
H	3	5	13	14	16	17
K	7	8	10	10	17	17
J	6	8	16	17	23	23
M	5	9	15	18	23	23
N	6	5	17	17	23	23

Problem 12'de yer alan şebeke, A-D-G-M, A-F-H-J, B-E-H-J, C-L-H-J ve C-I-K-N hatlarından oluşmaktadır. Belirtilen hatlardan kaç farklı aktivite önceliklerinin tayin edilebileceği hesaplanacaktır. Belirtilen hatlar kedi arasında ortak aktiviteler içermektedir. Bu nedenle bu hatlardaki ortak aktiviteleri içerecek biçimde öncelik sıralaması oluşturulmuştur. İlk önce A-D-G-M hattı ile F aktivitesi arasında aktivite önceliği kombinasyonları üretilmiştir. F aktivitesi A aktivitesi tamamlandıktan sonra başlayabileceği için A-F-D-G-M, A-D-F-G-M, A-D-G-F-M ve A-D-G-M-F öncelik sıralamaları oluşturulmuştur. Belirtilen öncelik sıralamaları B-E-H-J hattı ile kombinasyona tutularak aktivite öncelik kombinasyonları tanımlanır. Bu kombinasyona tabi tutma işleminde yer alan aktiviteler A-F-H-J hattındaki aktivitelerin tamamını kapsadığı için A-F-H-J hattının ayrıca kombinasyona tabi tutulmasına gerek kalmamaktadır. Örnek olarak A-F-D-G-M öncelik sıralaması ile B-E-H-J hattının kombinasyonu aşağıdaki tabloda gösterilmiştir.

Çizelge 3.28: Örnek olarak A-F-D-G-M ile B-E-H-J hattının öncelik sıralaması

B-E-A-F-H- J-D-G-M	B-E-A-F-H- D-J-G-M	B-E-A-F-H- D-G-J-M	B-E-A-F-H- D-G-M-J	B-E-A-F-D- H-J-G-M	B-E-A-F-D- H-G-J-M	B-E-A-F-D- H-G-M-J
B-E-A-F-D- G-H-J-M	B-E-A-F-D- G-H-M-J	B-E-A-F-D- G-M-H-J	B-A-E-F-H- J-D-G-M	B-A-E-F-H- D-J-G-M	B-A-E-F-H- D-G-J-M	B-A-E-F-H- D-G-M-J
B-A-E-F-D- H-J-G-M	B-A-E-F-D- H-G-J-M	B-A-E-F-D- H-G-M-J	B-A-E-F-D- G-H-J-M	B-A-E-F-D- G-H-M-J	B-A-E-F-D- G-M-H-J	B-A-F-E-H- J-D-G-M
B-A-F-E-H- D-J-G-M	B-A-F-E-H- D-G-J-M	B-A-F-E-H- D-G-M-J	B-A-F-E-D- H-J-G-M	B-A-F-E-D- H-G-J-M	B-A-F-E-D- H-G-M-J	
B-A-F-E-D- G-H-J-M	B-A-F-E-D- G-H-M-J	B-A-F-E-D- G-M-H-J	B-A-F-D-E- H-J-G-M	B-A-F-D-E- H-G-J-M	B-A-F-D-E- H-G-M-J	B-A-F-D-E- G-H-J-M
B-A-F-D-E- G-H-M-J	B-A-F-D-E- G-M-H-J	B-A-F-D-G- E-H-J-M	B-A-F-D-G- E-H-M-J	B-A-F-D-G- E-M-H-J		
B-A-F-D-G- M-E-H-J	A-B-E-F-H- J-D-G-M	A-B-E-F-H- D-J-G-M	A-B-E-F-H- D-G-J-M	A-B-E-F-H- D-G-M-J	A-B-E-F-D- H-J-G-M	A-B-E-F-D- H-G-J-M
A-B-E-F-D- H-G-M-J	A-B-E-F-D- G-H-J-M	A-B-E-F-D- G-H-M-J	A-B-E-F-D- G-M-H-J			
A-B-F-E-H- J-D-G-M	A-B-F-E-H- D-J-G-M	A-B-F-E-H- D-G-J-M	A-B-F-E-H- D-G-M-J	A-B-F-E-D- H-J-G-M	A-B-F-E-D- H-G-J-M	A-B-F-E-D- H-G-M-J
A-B-F-E-D- G-H-J-M	A-B-F-E-D- G-H-M-J	A-B-F-E-D- G-M-H-J				
A-B-F-D-E- H-J-G-M	A-B-F-D-E- H-G-J-M	A-B-F-D-E- H-G-M-J	A-B-F-D-E- G-H-J-M	A-B-F-D-E- G-H-M-J	A-B-F-D-E- G-M-H-J	A-B-F-D-G- E-H-J-M
A-B-F-D-G- E-H-M-J	A-B-F-D-G- E-M-H-J					
A-B-F-D-G- M-E-H-J	A-F-B-E-H- J--D-G-M	A-F-B-E-H- D-J-G-M	A-F-B-E-H- D-G-J-M	A-F-B-E-H- D-G-M-J	A-F-B-E-D- H-J-G-M	A-F-B-E-D- H-G-J-M
A-F-B-E-D- H-G-M-J						
A-F-B-E-D- G-H-J-M	A-F-B-E-D- G-H-M-J	A-F-B-E-D- G-M-H-J	A-F-B-D-E- H-J-G-M	A-F-B-D-E- H-G-J-M	A-F-B-D-E- H-G-M-J	A-F-B-D-E- G-H-J-M
A-F-B-D-E- G-H-M-J	A-F-B-D-E- G-M-H-J	A-F-B-D-G- E-H-J-M	A-F-B-D-G- E-H-M-J	A-F-B-D-G- E-M-H-J	A-F-B-D-G- M-E-H-J	
A-F-D-B-E- H-J-G-M	A-F-D-B-E- H-G-J-M	A-F-D-B-E- H-G-M-J	A-F-D-B-E- G-H-J-M	A-F-D-B-E- G-H-M-J		
A-F-D-B-E- G-M-H-J	A-F-D-B-G- E-H-J-M	A-F-D-B-G- E-H-M-J	A-F-D-B-G- E-M-H-J			
A-F-D-B-G- M-E-H-J	A-F-D-G-B- E-H-J-M	A-F-D-G-B- E-H-M-J				
A-F-D-G-B- E-M-H-J	A-F-D-G-B- M-E-H-J					
A-F-D-G-M- B-E-H-J						

Tabloda belirtildiği üzere toplamda $(14+13+12+11+10+9+8+7+6+5+4+3+2+1) = 105$ adet aktivite öncelik seçenekleri elde edilir. A-D-F-G-M öncelik sıralaması ile B-E-H-

J hattı kombinasyona tabi tutulursa aşağıdaki tabloda gösterilen öncelik sıralamaları elde edilir.

Çizelge 3.29: A-D-F-G-M ile B-E-H-J hattı kombinasyonunun öncelik sıralaması

B-E-A-D-F-H-J-G-M	B-E-A-D-F-H-G-J-M	B-E-A-D-F-H-G-M-J	B-E-A-D-F-G-H-J-M	B-E-A-D-F-G-H-M-J	B-E-A-D-F-G-M-H-J	B-A-E-D-F-H-J-G-M
B-A-E-D-F-H-G-J-M	B-A-E-D-F-H-G-M-J	B-A-E-D-F-G-H-J-M	B-A-E-D-F-G-H-M-J	B-A-E-D-F-G-M-H-J	B-A-D-E-F-H-J-G-M	
B-A-D-E-F-H-G-J-M	B-A-D-E-F-H-G-M-J	B-A-D-E-F-G-H-J-M	B-A-D-E-F-G-H-M-J	B-A-D-E-F-G-M-H-J	B-A-D-F-E-H-J-G-M	B-A-D-F-E-H-G-J-M
B-A-D-F-E-H-G-M-J	B-A-D-F-E-G-H-J-M	B-A-D-F-E-G-H-M-J	B-A-D-F-E-G-M-H-J	B-A-D-F-G-E-H-J-M		
B-A-D-F-G-E-H-M-J	B-A-D-F-G-E-M-H-J	B-A-D-F-G-M-E-H-J	A-B-E-D-F-H-J-G-M	A-B-E-D-F-H-G-J-M	A-B-E-D-F-H-G-M-J	A-B-E-D-F-G-H-J-M
A-B-E-D-F-G-H-M-J	A-B-E-D-F-G-M-H-J	A-B-D-E-F-H-J-G-M	A-B-D-E-F-H-G-J-M			
A-B-D-E-F-H-G-M-J	A-B-D-E-F-G-H-J-M	A-B-D-E-F-G-H-M-J	A-B-D-E-F-G-M-H-J	A-B-D-F-E-H-J-G-M	A-B-D-F-E-H-G-J-M	A-B-D-F-E-H-G-M-J
A-B-D-F-E-G-H-J-M	A-B-D-F-E-G-H-M-J	A-B-D-F-E-G-M-H-J				
A-B-D-F-G-E-H-J-M	A-B-D-F-G-E-M-H-J	A-B-D-F-G-E-M-H-J	A-B-D-F-G-M-E-H-J	A-D-B-E-F-H-J-G-M	A-D-B-E-F-H-G-J-M	A-D-B-E-F-H-G-M-J
A-D-B-E-F-G-H-J-M	A-D-B-E-F-G-H-M-J					
A-D-B-E-F-G-M-H-J	A-D-B-F-E-H-J-G-M	A-D-B-F-E-H-G-J-M	A-D-B-F-E-H-G-M-J	A-D-B-F-E-G-H-J-M	A-D-B-F-E-G-H-M-J	A-D-B-F-E-G-M-H-J
A-D-B-F-G-E-H-J-M						
A-D-B-F-G-E-H-M-J	A-D-B-F-G-E-M-H-J	A-D-B-F-G-M-E-H-J	A-D-F-B-E-H-J-G-M	A-D-F-B-E-H-G-J-M	A-D-F-B-E-H-G-M-J	A-D-F-B-E-G-H-J-M
A-D-F-B-E-G-H-M-J	A-D-F-B-E-G-M-H-J	A-D-F-B-G-E-H-J-M	A-D-F-B-G-E-H-M-J	A-D-F-B-G-E-M-H-J	A-D-F-B-G-M-E-H-J	
A-D-F-G-B-E-H-J-M	A-D-F-G-B-E-H-M-J	A-D-F-G-B-E-M-H-J	A-D-F-G-B-M-E-H-J	A-D-F-G-M-B-E-H-J		

Tüm seçeneklerin toplamı (13+12+11+10+9+8+7+6+5+4 + 3 + 2 + 1) = 81 adet aktivite öncelik seçeneği sunar. A-D-G-F-M öncelik sıralaması ile B-E-H-J hattı kombinasyona tabi tutulursa aşağıdaki tabloda gösterilen öncelik sıralamaları elde edilir.

Çizelge 3.30: A-D-G-F-M ile B-E-H-J hattı kombinasyonunun öncelik sıralaması

B-E-A-D- G-F-H-J-M	B-E-A-D- G-F-H-M-J	B-E-A-D- G-F-M-H-J	B-A-E-D- G-F-H-J-M	B-A-E-D- G-F-H-M-J	B-A-E-D- G-F-M-H-J	B-A-D-E- G-F-H-J-M
B-A-D-E- G-F-H-M-J	B-A-D-E- G-F-M-H-J	B-A-D-G- E-F-H-J-M	B-A-D-G- E-F-H-M-J	B-A-D-G- E-F-M-H-J		
B-A-D-G- F-E-H-J-M	B-A-D-G- F-E-H-M-J	B-A-D-G- F-E-M-H-J	B-A-D-G- F-M-E-H-J	A-B-E-D- G-F-H-J-M	A-B-E-D- G-F-H-M-J	A-B-E-D- G-F-M-H-J
A-B-D-E- G-F-H-J-M	A-B-D-E- G-F-H-M-J	A-B-D-E- G-F-M-H-J	A-B-D-G- E-F-H-J-M			
A-B-D-G- E-F-H-M-J	A-B-D-G- E-F-M-H-J	A-B-D-G- F-E-H-J-M	A-B-D-G- F-E-H-M-J	A-B-D-G- F-E-M-H-J	A-B-D-G- F-M-E-H-J	A-D-B-E- G-F-H-J-M
A-D-B-E- G-F-H-M-J	A-D-B-E- G-F-M-H-J	A-D-B-G- E-F-H-J-M				
A-D-B-G- E-F-H-M-J	A-D-B-G- E-F-M-H-J	A-D-B-G- F-E-H-J-M	A-D-B-G- F-E-H-M-J	A-D-B-G- F-E-M-H-J	A-D-B-G- F-M-E-H-J	A-D-G-B- E-F-H-J-M
A-D-G-B- E-F-H-M-J	A-D-G-B- E-F-M-H-J					
A-D-G-B- F-E-H-J-M	A-D-G-B- F-E-H-M-J	A-D-G-B- F-E-M-H-J	A-D-G-B- F-M-E-H-J	A-D-G-F- B-E-H-J-M	A-D-G-F- B-E-H-M-J	A-D-G-F- B-E-M-H-J
A-D-G-F- B-M-E-H-J	A-D-G-F- M-B-E-H-J					

Belirtilen kombinasyonda toplam (12+11+10+9+8+7+6+5+4 + 3 + 2 + 1) = 51 adet aktivite öncelik seçenekleri elde edilir. A-D-G-M-F öncelik sıralaması ile B-E-H-J hattı kombinasyona tabi tutulursa aşağıdaki tabloda gösterilen öncelik sıralamaları elde edilir.

Çizelge 3.31: A-D-G-M-F ile B-E-H-J hattı kombinasyonunun öncelik sıralaması

B-E-A-D-G-M- F-H-J	B-A-E-D-G-M- F-H-J	B-A-D-E-G-M- F-H-J	B-A-D-G-E-M- F-H-J	B-A-D-G-M-E- F-H-J	B-A-D-G- M-F-E-H- J
A-B-E-D-G-M- F-H-J	A-B-D-E-G-M- F-H-J	A-B-D-G-E-M- F-H-J	A-B-D-G-M-E- F-H-J	A-B-D-G-M-F- E-H-J	
A-D-B-E-G-M- F-H-J	A-D-B-G-E-M- F-H-J	A-D-B-G-M-E- F-H-J	A-D-B-G-M-F- E-H-J		
A-D-G-B-E-M- F-H-J	A-D-G-B-M-E- F-H-J	A-D-G-B-M-F- E-H-J			
A-D-G-M-B-E- F-H-J	A-D-G-M-B-F- E-H-J				
A-D-G-M-F-B- E-H-J					

Belirtilen kombinasyonda toplam (6+5+4 + 3 + 2 + 1) = 51 adet aktivite öncelik seçenekleri elde edilir.

Aktivitelerin öncelik sırasına göre; A aktivitesi 6, B aktivitesi 11, C aktivitesi 1, D aktivitesi 7, E aktivitesi 12, I aktivitesi 10, F aktivitesi 8, L aktivitesi 13, G aktivitesi 2 , H aktivitesi 14, K aktivitesi 3, J aktivitesi 5, M aktivitesi 9 ve N aktivitesi 14 olarak öncelik sıralamaları elde edilmiştir. A-F-D-G-M, A-D-F-G-M, A-D-G-F-M ve A-D-G-M-F öncelik sıralamaları B-E-H-J hattı ile kombinasyona tabi tutulduğunda toplam 258 adet aktivite öncelik sıralaması üretilebilir. Üretilen öncelik sıralamaları C-L-H-J ve C-I-K-N hatlarının kombinasyonu ile elde edilen C-L-I-K-N, C-I-L-K-N, C-I-K-L-N ve C-I-K-N-L öncelik sıralamaları ile ayrıca kombinasyona tutulurlar. Bu şekilde Şekil 3.35'te verilen şebeke de her bir aktivitenin değişen öncelik sırasına göre 1276044 adet kombinasyon üretilmiştir. İlk kombinasyon BEAFCLHJDGMKN aktivitelerinden oluşmaktadır. Aktivitelerin öncelik sırasına göre oluşturulmuş bu kombinasyona göre A aktivitesi 9, C aktivitesi 17, D aktivitesi 26 ve I aktivitesi 32 gün geciktirilerek proje süresi 72 gün olarak hesaplanmıştır. Geri kalan 1276043 alternatif için öncelik sırasına göre kaynak çakışmasının olduğu aktiviteler öncelik sıralamasına göre geciktirilerek şebeke çözülür.

Çizelge 3.32: 1276043 alternatif için öncelik sırasına göre kaynak çakışmasının olduğu aktivitelerin öncelik sıralaması

9	32	995	62	23347	86	1172	110
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
6762	33	5879	63	7018	87	12221	112
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
1	39	4056	64	211634	88	8686	113
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
1216	41	5998	65	441	89	146	114
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
3636	42	440	66	111330	90	62	115
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
1	43	799	67	24291	91	8	116
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
24255	44	199	68	1194	92	63	117
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
56795	45	22	69	222	93	1	118
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
10922	46	1023	70	89	94	3249	119
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
1576	47	3640	71	60715	95	6	120
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
3	48	266	72	11830	96	56	121
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
2965	49	2400	73	8335	97	44	122
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün

327	50	5116	74	945	98	3529	123
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
2708	51	2551	75	255166	99	6 alternatifte	124
alternatifte	gün	alternatifte	gün	alternatifte	gün		gün
18646	52	23632	76	83	100	2 alternatifte	126
alternatifte	gün	alternatifte	gün	alternatifte	gün		gün
2093	53	2101	77	6 alternatifte	101	9725	129
alternatifte	gün	alternatifte	gün		gün	alternatifte	gün
4311	54	430	78	83	102	101	130
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
16187	55	20422	79	61	103	128	131
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
1023	56	30077	80	16775	104	16	135
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
17393	57	212	81	433	105	54761	136
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
64421	58	5677	82	5 alternatifte	106	100	137
alternatifte	gün	alternatifte	gün		gün	alternatifte	gün
566	59	8354	83	1160	107	97	138
alternatifte	gün	alternatifte	gün	alternatifte	gün	alternatifte	gün
22847	60	54174	84	1 alternatifte	108	31	140
alternatifte	gün	alternatifte	gün		gün	alternatifte	gün
7	61	7202	85	2335	109		
alternatifte	gün	alternatifte	gün	alternatifte	gün		

Optimum proje süresi 59 gün olarak belirlenmiştir. En iyi sonucu veren bu problemde 59 gün kombinasyonlarından biri CILKABDFENGHMJ'dir. Bu kombinasyonda ortaya çıkan aktivite erteleme süreleri A aktivitesi 8 , B aktivitesi 8, C aktivitesi 0, D aktivitesi 0 , E aktivitesi 0 , I aktivitesi 0, F aktivitesi 0, L aktivitesi 0, G aktivitesi 0, H aktivitesi 33 ,K aktivitesi 0, J aktivitesi 0, M aktivitesi 0 ve N aktivitesi 0 gün ertelenmiştir.

4. TARTIŞMA VE SONUÇ

İnşaat sektöründe projelerin programlanmasında yaygın olarak kullanılan paket programlarda yer alan yöntemler KKPPP'yi yeterli bir şekilde çözememenin yanı sıra birçok program ücretlidir. Küçük ve orta ölçekli müteahhitler projelerde kaynak kullanımı için genellikle programların ücretinden dolayı ilave bir bütçe ayırmamaktadır. Kaynak kullanımı planlanmayan projelerde olası maliyet artışının yanı sıra proje süresi de uzayabilmektedir.

KKPPP özellikle imalat ve inşaat sektöründe yaygın olarak karşılaşılan bir problemdir. Proje tabanlı faaliyet gösteren inşaat sektöründe kaynak tahsisi her proje için baştan yapıldığı için planlamacılar kaynak kısıtlarının ihlalini düzenlerken en iyi çözümü bulmak için yeterli vakte sahip olmayabilirler.

Çalışmada geliştirilen uygulama sayesinde küçük ve orta ölçekli müteahhitlerin uygulaması imkansız olan optimizasyon algoritmaları yerine; tüm seçenekleri deneyerek çözüm yapan, küçük ölçekli projelerde KKPPP'nin tam sonucunu makul hesaplama süresinde elde eden ve müteahhitler tarafından uygulanabilen bir hesap cetveli uygulaması geliştirilmiştir. Böylece yükleniciler için inşaatın her aşamasında kaynakların mümkün olan en üst verimde kullanılması mümkün olup, inşaat maliyetlerinde de önemli ölçüde düşüşler sağlanabilecektir.

Bu çalışmada karmaşık optimizasyon yöntemlerini kullanmadan tüm proje çizelgeleme olasılıklarını deneyerek KKPPP'ni çözen bir algoritma geliştirilmiştir. Algoritma hesap tablosu üzerinde programlanması ile birlikte küçük ve orta ölçekli projelerde denenerek çalıştığı gösterilmiştir. KKPPP NP-Zor türü problemler olduğundan arama uzayı aktivite sayısının artması ile üstel biçimde arttığı için büyük problemlerde hesaplama süresi çok artmaktadır.

Çalışmada HCU ile yapılan analizlerin sonucunda incelenen projelerin mümkün olan en iyi çözümü elde edilmiş ve bazı projelerde büyük oranda iyileşme sağlanmıştır. Sonuç olarak geliştirilen algoritma, orta ve küçük ölçekli inşaat projelerinde etkili bir kaynak kullanımı sağlayarak, inşaatta planlama sürecinin verimliliğini arttırmaktadır.

5. KAYNAKLAR

- Araujo, J. A., Santos, H. G., Gendron, B., Jena, S. D., Brito, S. S., ve Souza, D. S.** (2020). Strong bounds for resource constrained project scheduling: Preprocessing and cutting planes. *Computers & Operations Research*, 113, 104782.
- A.A.B. Pritsker, L.J. Watters, P.M. Wolfe.** (1969), Multi project scheduling with limited resources: A zero-one programming approach, *Management Science* 16, 93–107.
- Araújo, José Alberto, Javier Pajares, and Adolfo Lopez-Paredes.** (2010) "Simulating the dynamic scheduling of project portfolios." *Simulation Modelling Practice and Theory* 18.10: 1428-1441.
- R. Alvarez-Valdes, J.M. Tamarit,** (1989) Heuristic algorithms for resource-constrained project scheduling: A review and an empirical analysis, in: R. Sowi_nski, J. Weßglarz (Eds.), *Advances in Project Scheduling*, Elsevier, Amsterdam, pp. 114±134.
- Banihashemi, Sayyid Ali, and Mohammad Khalilzadeh.**(2020). "Time-cost-quality-environmental impact trade-off resource-constrained project scheduling problem with DEA approach." *Engineering, Construction and Architectural Management*.
- Beşikci, Umut, Ümit Bilge, and Gündüz Ulusoy.**(2015). "Multi-mode resource constrained multi-project scheduling and resource portfolio problem." *European Journal of Operational Research* 240.1: 22-31.
- Bhaskar, Tarun, Manabendra N. Pal, and Asim K. Pal.** (2011). "A heuristic method for RCPSP with fuzzy activity times." *European Journal of Operational Research* 208.1 : 57-66.
- P. Brucker, B. Jurisch,** (1993). A new lower bound for the jobshop scheduling problem, *European Journal of Operational Research* 64. 156±167.
- P. Brucker, B. Jurisch, B. Sievers,** (1994). A branch and bound algorithm for the job-shop problem, *Discrete Applied Mathematics* 49.107±127.
- Brucker, P., ve Knust, S.** (1999). Solving Large-Sized Resource-Constrained Project Scheduling Problems. In *Project Scheduling* (pp. 27-51). Springer, Boston, MA.
- P. Brucker, S. Knust, A. Schoo, O. Thiele,** (1998). A branch and bound algorithm for the resource-constrained project scheduling problem, *European Journal of Operational Research*, 107. 272±288.
- T. Baar, P. Brucker, S. Knust,** (1997). Tabu-search algorithms for the resource-constrained project scheduling problem, Working Paper, Universiteit Osnabrock,
- Bukata, L., Šůcha, P., & Hanzálek, Z.** (2015). Solving the resource constrained project scheduling problem using the parallel tabu search designed for the CUDA platform. *Journal of Parallel and Distributed Computing*, 77, 58-68.

- M. Bartusch, R.H. Möhring, F.J. Radermacher**, (1988). Scheduling project networks with resource constraints and time windows, *Annals of Operations Research* 16. 201±240.
- J. Blazewicz, J.K. Lenstra, A.H.G. Rinnooy Kan**, (1983). Scheduling subject to resource constraints, *Discrete Applied Mathematics* 5 11±24.
- F.F. Bector**, (1990). Some efficient multi-heuristic procedures for resource-constrained project scheduling, *European Journal of Operational Research* 49. 3±13.
- Bold, Matthew, and Marc Goerigk**. (2021). "A compact reformulation of the two-stage robust resource-constrained project scheduling problem." *Computers & Operations Research* 130 : 105232.
- Chakraborty, Ripon K., Alireza Abbasi, and Michael J. Ryan**.(2020). "Multi-mode resource-constrained project scheduling using modified variable neighborhood search heuristic." *International Transactions in Operational Research* 27.1 : 138-167.
- Coelho, J., ve Vanhoucke, M.** (2020). Going to the core of hard resource-constrained project scheduling instances. *Computers & Operations Research*, 121, 104976.
- N. Christofides, R. Alvarez±Valdés, J.M. Tamarit**, (1987).Project scheduling with resource constraints: A branch and bound approach, *European Journal of Operational Research* 29.262-273.
- E.W. Davis, J.H. Patterson**, (1975). A comparison of heuristic and optimum solutions in resource-constrained project scheduling, *Management Science* 21 . 944±955.
- E. Demeulemeester, W. Herroelen**, (1992), A branch-and-bound procedure for the multiple resource-constrained project scheduling problem, *Management Science* 38.1803-1818.
- E. Demeulemeester, W. Herroelen**, (1997), New benchmark results for the resource-constrained project scheduling problem, *Management Science* 43 (1997) 1485-1492.
- S.E. Elmaghraby**, (1995), Activity nets: a guided tour through some recent developments, *European Journal of Operational Research* 82 . 383-408.
- F. Glover**, (1986), Future paths for integer programming and links to artificial intelligence, *Comput. Oper. Res.* 13 (5) .533–549.
- Hartmann, Sönke**.(1998). "A competitive genetic algorithm for resource-constrained project scheduling." *Naval Research Logistics (NRL)* 45.7 : 733-750.
- Hauder, Viktoria A., et al.**(2020). "Resource-constrained multi-project scheduling with activity and time flexibility." *Computers & Industrial Engineering* 150 : 106857.
- Issa, S., and Y. Tu**. (2020)."A survey in the resource-constrained project and multi-project scheduling problems." *Journal of Project Management* 5.2 : 117-138.

- G. Igelmund, F.J. Radermacher,**(1983), Preselective strategies for the optimization of stochastic project networks under resource constraints, *Networks* 13.1±28.
- G. Igelmund, F.J. Radermacher,**(1983), Algorithmic approaches to preselective strategies for stochastic scheduling problems, *Networks* 13.29±48.
- T.J.R. Johnson,** 1967, An algorithm for the resource-constrained project scheduling problem, Ph.D. Dissertation, MIT, Boston, USA.
- Kadam, Sachin U., and Narendra S. Kadam.** 2014, "Solving resource-constrained project scheduling problem by genetic algorithm." *2014 2nd International Conference on Business and Information Management (ICBIM)*. IEEE.
- Kadri, Roubila Lilia, and Fayez F. Bector,** (2018). "An efficient genetic algorithm to solve the resource-constrained project scheduling problem with transfer times: The single mode case." *European Journal of Operational Research* 265.2: 454-462.
- Kanıt, Recep, Umut Naci Baykan, and Mürsel ERDAL.** (2005) "Kısıtlı Kaynak Koşullarının Yapı Maliyetine Etkisinin İncelenmesi." *Politeknik Dergisi* 8.2: 209-221.
- Krüger, D., ve Scholl, A.** (2009). A heuristic solution framework for the resource constrained (multi-) project scheduling problem with sequence-dependent transfer times. *European Journal of Operational Research*, 197 , 492–508 .
- R. Klein, A. Scholl,** (1997), Computing lower bounds by destructive improvement ± An application to resource constrained project scheduling, Working Paper, Technische Universität Darmstadt, *European Journal of Operational Research*, to appear.
- Kurtulus, I.S. and E.W. Davis** (1982): Multi-project scheduling: Categorization of heuristic rules Performance, *Management Science*, Vol. 28, pp. 161-172.
- Kurtulus, I.S. and S.C. Narula** (1985): Multi-project scheduling: Analysis of project Performance, *IIE Transactions*, Vol. 17, pp.58-66.
- Kong, Feng, and Dong Dou.**(2021)."Resource-Constrained Project Scheduling Problem under Multiple Time Constraints." *Journal of Construction Engineering and Management* 147.2 : 04020170.
- Kosztván, Zsolt T., and István Szalkai.** (2020) "Multimode resource-constrained project scheduling in flexible projects." *Journal of Global Optimization* 76.1 (2020): 211-241.
- Kolisch, R.** (1994): Project scheduling under resource constraints - Efficient heuristics for several problem classes, PhD Dissertation, Kiel.
- Kolisch, R., Sprecher, A., & Drexel, A.** (1995). Characterization and generation of a general class of resource-constrained project scheduling problems. *Management Science*, 41 (10), 1693–1703 .
- Kolisch, R., & Sprecher, A.** (1996). PSPLIB –A project scheduling problem library. *European Journal of Operational Research*, 96 (1), 205–216 .

- Kolisch, Rainer.**(1996). "Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation." *European Journal of Operational Research* 90.2 : 320-333.
- Kolisch, R., and A. Sprecher.** (1997). "PSPLIB: A project scheduling problem library: OR software—ORSEP operations research software exchange program." *Eur. J. Oper. Res.* 96 (1): 205–216. [https://doi.org/10.1016/S0377-2217\(96\)00170-1](https://doi.org/10.1016/S0377-2217(96)00170-1).
- R. Kolisch, A. Drexl,**(1996),Adaptive search for solving hard project scheduling problems, *Naval Research Logistics* 43.23-40.
- R. Kolisch, R. Padman,** (1997). An integrated perspective of project scheduling, *Manuskripte aus den Instituten für Betriebswirtschaftslehre der Universität Kiel*, no. 463.
- R. Kolisch, A. Sprecher, A. Drexl,**(1995), Characterization and generation of a general class of resource-constrained project scheduling problems, *Management Science* 41. 1693-1703.
- Laborie, P., J. Rogerie, P. Shaw, and P. Vilím.** (2018). "IBM ILOG CP optimizer for scheduling." *Constraints* 23 (2): 210–250. <https://doi.org/10.1007/s10601-018-9281-x>.
- Laurent, A., et al.** (2017) "A new extension of the RCPSP in a multi-site context: Mathematical model and metaheuristics." *Computers & Industrial Engineering* 112: 634-644.
- Li, H., Womer, N.K.,** (2015). Solving stochastic resource-constrained Project scheduling problems by closed-loop approximate dynamic programming. *Eur. J. Oper. Res.* 246 (1), 20–33.
- Lin, Jian, Lei Zhu, and Kaizhou Gao.**(2020) "A genetic programming hyper-heuristic approach for the multi-skill resource constrained project scheduling problem." *Expert Systems with Applications* 140 : 112915.
- Liu, Jia, et al.**(2020) "Solving resource-constrained project scheduling problem via genetic algorithm." *Journal of Computing in Civil Engineering* 34.2 : 04019055.
- Merkle, Daniel, Martin Middendorf, and Hartmut Schmeck.**(2002) "Ant colony optimization for resource-constrained project scheduling." *IEEE transactions on evolutionary computation* 6.4 : 333-346.
- A. Mingozzi, V. Maniezzo, S. Ricciardelli, L. Bianco,**(1998), An exact algorithm for the resource-constrained Project scheduling based on a new mathematical formulation, *Management Science* 44 , 714±729.
- Myszkowski, P. B. , Skowronski, M. E. , & Sikora, K.** (2015). A new benchmark dataset for multi-skill resource-constrained project scheduling problem. In *M. Ganzha, L.*
- Möhring, R.H., Stork, F.,** 2000. Linear preselective policies for stochastic Project scheduling. *Math. Methods Oper. Res.* 52 (3), 501–515.

- Ozturk, Guler, and Adalet Oner.**(2020) "Continuous Time MILP Models for Multi-Mode Resource Constrained Project Scheduling Problems." *9th International Conference on Industrial Technology and Management (ICITM)*. IEEE, 2020.
- L. Ozdamar, G. Ulusoy,**(1995), A survey on the resource constrained project scheduling problem, *IIE Transactions* 27 . 574±586
- J.H. Patterson,**(1976), Project scheduling: The effect of problem structure on heuristic performance, *Naval Research Logistics Quarterly* 23.95±124.
- J.H. Patterson,**(1984), A comparison of exact approaches for solving the multiple constrained resource project scheduling problem, Research Report, Department of Industrial and Manufacturing Systems Engineering, Lehigh University.
- J.H. Patterson, R. Sowi_nski, F.B. Talbot, J. Weißglarz,** (1989). An algorithm for a general class of precedence and resource constrained scheduling problems, in: R. Sowi_nski, J. Weißglarz (Eds.), *Advances in Project Scheduling*, Elsevier, Amsterdam, pp. 3±28.
- Patterson, James H.**(1984). "A comparison of exact approaches for solving the multiple constrained resource, project scheduling problem." *Management science* 30.7 : 854-867.
- Pellerin, Robert, Nathalie Perrier, and François Berthaut.**(2020). "A survey of hybrid metaheuristics for the resource-constrained project scheduling problem." *European Journal of Operational Research* 280.2 : 395-416.
- Poppenborg, J. , ve Knust, S.** (2014). A flow-based tabu search algorithm for the RCPSP with transfer times. In *Proceedings of the 14th international conference on project management and scheduling* (pp. 181–184) .
- Poppenborg, J. , ve Knust, S.** (2016). A flow-based tabu search algorithm for the RCPSP with transfer times. *OR Spectrum*, 38 , 305–335 .
- Rahman, Humyun Fuad, Ripon K. Chakraborty, and Michael J. Ryan.** (2020). "Memetic algorithm for solving resource constrained project scheduling problems." *Automation in Construction* 111 : 103052.
- Ren, Yifei, Zhiqiang Lu, and Xinyi Liu.**(2020) "A branch-and-bound embedded genetic algorithm for resource-constrained project scheduling problem with resource transfer time of aircraft moving assembly line." *Optimization Letters* : 1-35.
- Riedler, Martin, et al.** (2020) "An iterative time-bucket refinement algorithm for a high-resolution resource-constrained project scheduling problem." *International Transactions in Operational Research* 27.1 : 573-613.
- Sallam, Karam M., Ripon K. Chakraborty, and Michael J. Ryan.**(2020) "A two-stage multi-operator differential evolution algorithm for solving resource constrained project scheduling problems." *Future Generation Computer Systems* 108 : 432-444.

- Snauwaert, Jakob, and Mario Vanhoucke.**(2021) "A new algorithm for resource-constrained project scheduling with breadth and depth of skills." *European Journal of Operational Research* 292.1 : 43-59.
- A. Schoo,**(1996),Untere Schranken und Immediate Selection four das Resource Constrained-Projekt Scheduling Problem, Diplomarbeit, Fachbereich Mathematik/Informatik, Unversiteat Osnabrouck.
- A. Sprecher,**(1996), Solving the RCPSP efficiently at modest memory requirements, Manuskripte aus den Instituten four Betriebswirtschaftslehre der Unversiteat Kiel, no. 426.
- J.P. Stinson,**(1978), E.W. Davis, B.M. Khumawala, Multiple resource-constrained scheduling using branch and bound, *AIIE Transactions* 10 (1978) 252±259.
- Stinson, J.P.** (1976), "A Branch and Bound Algorithm for a General Class of Resource-Constrained Scheduling Problems," Ph.D. Dissertation, University of North Carolina at Chapel Hill.
- Tao, S., ve Dong, Z. S.** (2017). Scheduling resource-constrained project problem with alternative activity chains. *Computers & Industrial Engineering*, 114, 288–296. <http://dx.doi.org/10.1016/j.cie.2017.10.027>.
- TALBOT, F.B.**(1976) "An Integer Programming Algorithm for the Resource-Constrained Project Scheduling Problem," Ph.D. Dissertation, Pennsylvania State University.
- Talbot, F. B.** (1982). Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case. *Management science*, 28(10), 1197-1210.
- Talbot, F. B., & Patterson, J. H.** (1978). An efficient integer programming algorithm with network cuts for solving resource-constrained scheduling problems. *Management Science*, 24 (11), 1163-1174.
- Tsang,** (1993) E. Foundations of Constraint Satisfaction, Academic Press, London.
- Ulusoy, G.** (2006), "Proje planlamada kaynak kısıtlı çizelgeleme", Sabancı Üniversitesi, Mühendislik ve Doğa Bilimleri Fakültesi.
- G. Ulusoy, L. Ozdamar,** (1994),A constrained-based perspective in resource constrained project scheduling, *International Journal of Production Research* 32.693-705.
- Van Peteghem, V., Vanhoucke, M. ,** 2014. An experimental investigation of meta-heuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. *Eur. J. Oper. Res.* 235 (1), 62–72.
- Vanhoucke, M., ve Coelho, J.,** 2018. A tool to test and validate algorithms for the resource-constrained project scheduling problem. *Comput. Industr. Eng.* 118, 251–265.
- Zhang, Hong, et al.** (2005) "Particle swarm optimization-based schemes for resource-constrained project scheduling." *Automation in construction* 14.3 : 393-404.

- Wang, X., Chen, Q., Mao, N., Chen, X., & Li, Z.** (2015). Proactive approach for stochastic RCMPSP based on multi-priority rule combinations. *International Journal of Production Research*, 53(4), 1098–1110.
- Wang, Yanting, et al.** (2017)"On the performance of priority rules for the stochastic resource constrained multi-project scheduling problem." *Computers & industrial engineering* 114 : 223-234.
- Wauters, T., Kinable, J., Smet, P., Vancroonenburg, W., Vanden Berghe, G., Verstichel, J.,** (2016). The multi-mode resource-constrained multi-project scheduling problem. *J. Scheduling* 19, 271–283 .



ÖZGEÇMİŞ

Ad Soyadı: Derya Çakmak

Lisans: İnönü Üniversitesi Mühendislik Fakültesi İnşaat Mühendisliği Bölümü

Mesleki Deneyim ve Ödüller:

Malatya İmar Yapı Denetim 2018-2020

Yayın Listesi:

Çakmak, D., ve Bettemir, Ö. H. "Yapı Denetim Sistemi Sorunlarının Belirlenmesi Üzerine Vaka Çalışması: Şanlıurfa il Merkezi A Case Study for the Detection of the Problems of Building Audit System: Şanlıurfa Province", Uluslararası Katılımlı 7. İnşaat Yönetimi Kongresi, 06 - 07 Ekim 2017, Samsun, Türkiye.