

Kaynak Kısıtlı İnşaat Projeleri Süre-Gider Eniyilemesinde Karşılaştırmalı
Melez-Metasezgisel Yöntem Seçimi

Gülçağ Albayrak

DOKTORA TEZİ

İnşaat Mühendisliği Anabilim Dalı

Aralık 2016



A Comparative Selection of Hybrid-Metaheuristic Methods on Time-Cost Trade-Off
in Resource Constrained Construction Projects

Gülçağ Albayrak

DOCTORAL DISSERTATION

Department of Civil Engineering

December 2016

Kaynak Kısıtlı İnşaat Projeleri Süre-Gider Eniyilemesinde Karşılaştırmalı
Melez-Metasezgisel Yöntem Seçimi

Gülçağ Albayrak

Eskişehir Osmangazi Üniversitesi

Fen Bilimleri Enstitüsü

Lisansüstü Yönetmeliği Uyarınca

İnşaat Mühendisliği Anabilim Dalı

Yapı Bilim Dalında

DOKTORA TEZİ

Olarak Hazırlanmıştır

Danışman: Prof.Dr. İlker Özdemir

Aralık 2016

ONAY

İnşaat Mühendisliği Anabilim Dalı Doktora öğrencisi Gülçağ ALBAYRAK'ın DOKTORA tezi olarak hazırladığı “Kaynak Kısıtlı İnşaat Projeleri Süre-Gider Eniyilemesinde Karşılaştırmalı Melez-Metasezgisel Yöntem Seçimi” başlıklı bu çalışma, jürimizce lisansüstü yönetmeliğin ilgili maddeleri uyarınca değerlendirilerek oybirliği ile kabul edilmiştir.

Danışman : Prof.Dr. İlker ÖZDEMİR

İkinci Danışman : -

Doktora Tez Savunma Jürisi:

Üye : Prof.Dr. İlker ÖZDEMİR

Üye : Doç.Dr. Rifat AKBIYIKLI

Üye : Yrd.Doç.Dr. Osman AYTEKİN

Üye : Yrd.Doç.Dr. Serkan KIVRAK

Üye : Yrd.Doç.Dr. Hakan KUŞAN

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

Prof. Dr. Hürriyet ERŞAHAN
Enstitü Müdürü

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Prof.Dr. İlker Özdemir danışmanlığında hazırlamış olduğum “Kaynak Kısıtlı İnşaat Projeleri Süre-Gider Eniyilemesinde Karşılaştırmalı Melez-Metasezgisel Yöntem Seçimi” başlıklı DOKTORA tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 14/12/2016

Gülçağ Albayrak

İmza

ÖZET

Bir inşaat projesinin başarısını belirleyen en önemli unsurların başında, öngörülen zaman ve bütçe dahilinde tamamlanabilmesi gelmektedir. Projelerin hayata geçirilmesinde gerekli olan tüm kaynakların (işgücü, malzeme, makine, ekipman vs.) kısıtlı olarak bulunduğu bilinmektedir. Yalnızca maddi kaynakların değil, aynı zamanda sürenin de kısıtlı oluşu süre-gider ilişkisi olarak tanımlanan kavramın temelini oluşturur. Süre ve gider arasında bir denge kurulabilmesi projenin planlama aşamasındaki kritik noktalardan biridir. Bu iki proje ögesinin kısıtlar nedeniyle birbiriyle çelişmesi, eniyileme olarak adlandırılan tekniklerin geliştirilmesine öncülük etmiştir. Eniyileme, farklı çözümler içerisinde amaca en uygun olanı elde etmeye yönelik çalışmalardır.

Geçmişte süre-gider eniyilemesi yalnızca geleneksel olarak nitelendirilen bir takım deterministik yöntemler ile yapılmaktayken, günümüzde daha güncel stokastik yöntemler de kullanılmaya başlanmıştır. Geleneksel yöntemlerin karmaşık problemlerde çözüme ulaşmaktaki yetersizliği nedeniyle geliştirilen bu yeni teknikler toplu halde metasezgisel algoritmalar olarak isimlendirilmektedir. Metasezgisel algoritmalar sunduğu çözümler bakımından her zaman en iyiyi garanti altına almamakla birlikte en iyiye yakınsama özelliğine sahiptir. Çoğunlukla doğadan esinlenerek geliştirilen metasezgisel algoritmaların pek çok türü bulunmaktadır.

Bu tez çalışması kapsamında süre-gider eniyilemesinin teorik altyapısı ortaya konularak probleme yönelik çeşitli çözüm teknikleri sunulmuştur. Öncelikle geleneksel yöntemlerden serim akış algoritması ve analitik yöntem ile yaklaşım geliştirilmiştir. Ardından metasezgisel algoritmalarından genetik algoritma ve parçacık sürü optimizasyonu probleme uygun şekilde modellenerek bilgisayar programında kodlanmıştır. Tezin amacı, süre-gider eniyilemesi probleminin çözümüne yönelik tekniklerin geliştirilmesi ve karşılaştırmalar yaparak farklı özellikteki uygulamalarda uygun yöntemin seçimini göstermektir.

Anahtar Kelimeler: süre-gider eniyilemesi, kaynak kısıtlı proje, metasezgisel algoritmalar, proje planlama.

SUMMARY

The completion of a project within the estimated time and budget is one of the most important factors for determining the success of a construction project. It is known that, all the resources (labour, materials, machinery, equipment etc.) required by the projects are limited. Not only the limited material resources, but also limited duration of the project are the basis of the concept defined as time-cost relationship. Balancing of duration and expenditure is one of the critical points in the planning steps of the project. The contradiction between these two elements leads to the development of techniques called optimization. Optimization is aimed to achieve the most appropriate one in different solutions.

In the past, time-cost trade-off was only executed with a number of deterministic methods defined as traditional; however more current stochastic methods are being used nowadays. These new techniques, which are developed due to the inadequacy of the conventional methods to solve the complex problems, are called metaheuristic algorithms. Even though metaheuristic algorithms do not assure the best through the solutions for every time, they have the feature of convergence to the best. There are many types of metaheuristic algorithms which are developed with mostly inspiration from the nature.

Within the scope of this thesis, the theoretical background of time-cost trade-off is presented and various solution techniques for the problem are presented. First of all, the approach has been developed by network flow algorithm and analytical method from traditional methods. Then, genetic algorithms and particle swarm optimization are modelled properly according to the problem and coded in the computer program. The aim of the thesis is to develop and compare techniques for solving time-cost trade-off problem and show the choice of the appropriate method in different applications.

Keywords: time-cost trade-off, resource limited project, metaheuristic algorithms, project planning.

TEŐEKKÖR

Doktora tez alıŐması sűresince, bilgi, tecrűbe ve yardımlarıyla her aŐamada beni yűnlendiren saygıdeđer hocam, tez danıŐmanım Prof.Dr. İlker ÖZDEMİR'e ve tez savunma jűrisinde bulunan deđerli hocalarıma katkılarından dolayı teŐekkűrlerimi sunarım.

Bununla beraber, her zaman yanımda olan ve desteđini daima hissettiđim sevgili eŐim Dr. Uđur ALBAYRAK'a ve biricik ođlumuz Kayra'ya teŐekkűr ederim.



İÇİNDEKİLER

| | <u>Sayfa</u> |
|---|--------------|
| ÖZET | vi |
| SUMMARY | vii |
| TEŞEKKÜR | viii |
| İÇİNDEKİLER | ix |
| ŞEKİLLER DİZİNİ | xi |
| ÇİZELGELER DİZİNİ | xiii |
| SİMGELER VE KISALTMALAR DİZİNİ | xiv |
| 1. GİRİŞ | 1 |
| 2. LİTERATÜR ARAŞTIRMASI | 4 |
| 2.1. Matematiksel Programlama, Doğrusal Modelleme ve Deterministik Yaklaşım.... | 4 |
| 2.2. Metasezgisel Yöntemler..... | 5 |
| 3. MATERYAL VE YÖNTEM | 10 |
| 3.1. Projelerde Kaynakların Sınıflandırılması..... | 10 |
| 3.1.1. Kısıtlı kaynak kullanımı..... | 11 |
| 3.2. İşlem Süresi-Kaynak Kullanımı İlişkisi..... | 12 |
| 3.3. Süre-Gider Eniyilemesi..... | 14 |
| 3.3.1. Süre-gider eniyileme probleminin matematiksel modeli..... | 17 |
| 3.3.2. Süre-gider eniyileme problemine serim akış algoritması ile yaklaşım..... | 18 |
| 4. METASEZGİSEL ALGORİTMALAR | 28 |
| 4.1. Genetik Algoritmalar..... | 31 |
| 4.1.1. Genetik Algoritma Operatörleri..... | 34 |
| 4.1.2. Genetik Algoritma Prosedürleri..... | 36 |
| 4.2. Parçacık Sürü Optimizasyonu..... | 39 |
| 5. ANALİTİK VE METASEZGİSEL YÖNTEMLER KULLANARAK SÜRE-GİDER ENİYİLEME UYGULAMASI | 42 |
| 5.1. Analitik Yöntemle Süre-Gider Eniyileme Uygulaması..... | 42 |
| 5.2. Genetik Algoritmalar (GA) ile Yaklaşım..... | 50 |
| 5.2.1. Genetik algoritmalar (GA) ile süre-gider eniyileme uygulaması..... | 53 |

İÇİNDEKİLER (devam)

| | <u>Sayfa</u> |
|---|---------------------|
| 5.3. Parçacık Sürü Optimizasyonu (PSO) ile Yaklaşım..... | 57 |
| 5.3.1. Parçacık sürü optimizasyonu (PSO) ile süre-gider eniyileme uygulaması... | 57 |
| 6. BULGULAR VE TARTIŞMA..... | 65 |
| 7. SONUÇLAR VE ÖNERİLER..... | 69 |
| KAYNAKLAR DİZİNİ..... | 71 |
| ÖZGEÇMİŞ | |

ŞEKİLLER DİZİNİ

| <u>Şekil</u> | <u>Sayfa</u> |
|---|--------------|
| 3.1.a. İşlemlerde süre-gider ilişkisi..... | 12 |
| 3.1.b. Projelerde süre-gider eğrisi..... | 12 |
| 3.2.a. Sürekli süre-gider fonksiyon..... | 13 |
| 3.2.b. Kesikli süre-gider fonksiyonu..... | 13 |
| 3.3.a. İşlemlerde süre-gider fonksiyon türleri: doğrusal (parçalı)..... | 14 |
| 3.3.b. İşlemlerde süre-gider fonksiyon türleri: doğrusal olmayan (iç ve dışbükey melez) | 14 |
| 3.3.c. İşlemlerde süre-gider fonksiyon türleri: kesikli..... | 14 |
| 3.4. Akış algoritmasıyla süre-gider eniyilemesi örnek serimi..... | 20 |
| 3.5. Serim akış algoritması ile süre-gider eniyilemesi (a)..... | 21 |
| 3.5. Serim akış algoritması ile süre-gider eniyilemesi (b), (c), (d)..... | 22 |
| 3.5. Serim akış algoritması ile süre-gider eniyilemesi (e), (f), (g)..... | 23 |
| 3.5. Serim akış algoritması ile süre-gider eniyilemesi (h), (i), (j)..... | 24 |
| 3.5. Serim akış algoritması ile süre-gider eniyilemesi (k), (l), (m)..... | 25 |
| 3.6. Serim akış algoritmasıyla süre-gider eniyileme sonuç grafiği..... | 27 |
| 4.1. Metasezgisel algoritmaların akış şeması..... | 31 |
| 4.2. Tek noktadan çaprazlama işlemi örneği..... | 35 |
| 4.3. Mutasyon işlemi örneği..... | 35 |
| 4.4. GA akış şeması..... | 38 |
| 4.5. PSO akış şeması..... | 41 |
| 5.1. Proje serimi, başlangıç çözümü ve kritik yol gösterimi..... | 43 |
| 5.2. İlk hızlandırmadan sonraki durum ve kritik yolların gösterimi..... | 45 |
| 5.3. İkinci hızlandırmadan sonraki durum ve kritik yolların gösterimi..... | 46 |
| 5.4. Üçüncü hızlandırmadan sonraki durum ve kritik yolların gösterimi..... | 47 |
| 5.5. Dördüncü hızlandırmadan sonraki durum ve kritik yolların gösterimi..... | 48 |
| 5.6. Beşinci hızlandırmadan sonraki durum ve kritik yolların gösterimi..... | 49 |
| 5.7. Analitik süre-gider eniyileme çözüm grafiği..... | 50 |
| 5.8. GA'nın süre-gider eniyileme problemine uyarlanması..... | 51 |
| 5.9. GA akış şemasına süre-gider eniyileme probleminin dahil edilmesi..... | 52 |

ŞEKİLLER DİZİNİ (devam)

| <u>Şekil</u> | <u>Sayfa</u> |
|---|---------------------|
| 5.10. GA uygulaması için örnek proje serimi..... | 54 |
| 5.11. GA ile süre-gider eniyileme grafiđi..... | 57 |
| 5.12. PSO uygulaması için örnek proje serimi..... | 60 |
| 5.13. PSO algoritmasına ait MATLAB kodlamaları..... | 62 |
| 5.14. PSO ile süre-gider eniyileme sonuç grafiđi..... | 64 |
| 5.15. Uyum değeri-iterasyon sayısı grafiđi..... | 64 |

ÇİZELGELER DİZİNİ

| <u>Çizelge</u> | <u>Sayfa</u> |
|---|--------------|
| 2.1. Süre-gider eniyileme probleminde son on yıla ait literatür özeti..... | 6 |
| 3.1. Serim akış yöntemiyle süre-gider eniyilemesinde zorlama giderleri..... | 26 |
| 3.2. Serim akış yöntemiyle süre-gider eniyilemesi sonuçları..... | 26 |
| 4.1. Doğadan esinlenen metasezgisel algoritmalar..... | 29 |
| 4.2. Sürü zekasına dayalı metasezgisel algoritmalar..... | 29 |
| 4.3. Diğer bazı metasezgisel algoritmalar..... | 30 |
| 4.4. Araştırmacılar tarafından önerilmiş kontrol değişken değerleri..... | 36 |
| 5.1. Analitik uygulamaya ilişkin işlem bilgileri..... | 43 |
| 5.2. Analitik süre-gider eniyilemesi sonuçları..... | 49 |
| 5.3. GA uygulaması için örnek proje verileri..... | 53 |
| 5.4. GA ile süre-gider eniyileme sonuçları..... | 56 |
| 5.5. PSO uygulaması için örnek proje bilgileri..... | 58 |
| 5.6. PSO ile süre-gider eniyileme sonuçları..... | 63 |

SİMGELER VE KISALTMALAR DİZİNİ

| <u>Simgeler</u> | <u>Açıklama</u> |
|-----------------|---|
| $a(j)$ | Düğüme gelen akım miktarı |
| a_{ik} | i. durum ve k. durum için süre-gider fonksiyonunun eğimi |
| B_{ijk} | İşleme ait bolluk |
| b_{ik} | i. durum ve k. durum için süre-gider fonksiyonunun kesişimi |
| c | Hızlandırma katsayısı |
| $C_{(aday)}$ | Aday karar değişkeni |
| $C_{(ilk)}$ | Başlangıç karar değişkeni |
| EB_i | i işlemine ait erken başlama zamanı |
| E_{ij} | ij işlemine ait süre-gider fonksiyon eğimi |
| E_{ijk} | i'den j'ye k. yoldan akım gönderme bedeli |
| f_{ijk} | i'den j'ye giden k. yoldaki akım miktarı |
| GB | Geç başlama zamanı |
| $g_{best}(s)$ | s zamanında sürü içinde bulunan en iyi konum |
| g_d | Günlük dolaylı gider |
| g_i | i işlemine ait gider |
| g_{ij} | i. işleme ait j. mod gideri |
| GT | Geç tamamlanma zamanı |
| $h_i(s)$ | s. zamanda çözüm uzayında yer alan i parçacığının hızı |
| k^\pm | Akımın geldiği yön |
| $l_{best}(s)$ | s zamanında bireysel en iyi konum |
| mn | Mod seçenekleri |
| p | Atalet katsayısı |
| $P(i)$ | Çözüm uzayı |
| $P(i+1)$ | Yeni çözüm uzayı |
| $r_j(s)$ | Stokastik rasgele sayı |
| s_i | i işlemine ait süre |
| S_{ij} | ij işlemine ait normal süre |

SİMGELER VE KISALTMALAR DİZİNİ (devam)

| <u>Simgeler</u> | <u>Açıklama</u> |
|------------------------|---|
| s_{ij} | ij işlemine ait zorlanmış süre |
| S_n | n. işlemin başlama zamanı |
| T | Proje tamamlanma süresi |
| T_s | Aşılmaması gereken proje bitiş süresi |
| $x_i(s)$ | s. zamanda çözüm uzayında yer alan i parçacığının pozisyonu |
| x_{ij} | i işleminin j. moduna ait karar değişkeni |
| y_g | s+1 zamanında küresel en iyi konum |
| y_i | s+1 zamanında bireysel en iyi konum |
| ZG | Zorlanabilecek gün sayısı |

| <u>Kısaltmalar</u> | <u>Açıklama</u> |
|---------------------------|-----------------------------|
| EA | Evrimsel algoritmalar |
| GA | Genetik algoritmalar |
| PSO | Parçacık sürü optimizasyonu |

1. GİRİŞ

Süre ve gider her inşaat projesi için en önemli iki problemdir (Ng ve Zhang, 2008). Sürenin ve giderin belirlenmesi bir projenin başarılı ya da başarısız sonuçlanmasına etki eden en kritik aşamasını teşkil etmektedir.

1950'lerden itibaren ortaya çıkışı ve uygulaması ile beraber kritik yol yöntemi proje sürelerinin hesaplanmasında, proje planlama ve çizelgelemesinde bir temel teşkil etmiştir. Bununla beraber kritik yol yöntemi kritik yol üzerindeki işlemlerin sürelerini toplayarak toplam proje süresini hesaplamakta, ancak açık bir şekilde proje bitiş süresi kısıtlamasına riayet etmemektedir. Ayrıca bu yöntem proje giderleri hesabına da katkı sunmamaktadır. Bundan dolayı süre ve gider arasında bir ödünleşme ihtiyacı doğmaktadır. Burada hedeflenen, mümkün olan en düşük maliyet artışı ile, başlangıçta hesaplanan kritik yol süresini kısaltarak spesifik olarak belirlenmiş proje tamamlanma süresini elde edebilmektir. Bu amaç projenin içerdiği işlemlerden bazılarının daha hızlı ve genellikle daha pahalı alternatiflerinin seçilmesi ile mümkün olabilmektedir (Hegazy ve Ayed, 1999).

Bir projeye ait işlemler yürütülürken proje yöneticisi tarafından farklı mod kombinasyonları seçilebilir. Bu modlar, malzeme ya da inşaat teknolojisi değişikliği, işgücü ve ekipmanda düzenleme, fazla mesai ya da vardiya planlamaları ile oluşturulmaktadır (El-Rayes ve Kandil, 2005; Hegazy ve Menesi, 2010).

Alternatif modlara sahip her bir işlem için toplamdaki değerleri de etkileyen ayrı bir gider ve süre hesabı gerekmektedir. İşlemlerin olası modlarının kombinasyonları birbirinden farklı çok sayıda proje planı ortaya koymaktadır. Bu nedenle büyük projelerde bu alternatiflerin ayrı ayrı hesaplanabilmesi zordur, çünkü işlem sayısındaki çoğalma hesaplama yükünde eksponansiyel artışa neden olmaktadır. Örnek olarak 180 işlem içeren bir projede her bir işlemin yalnızca 3 modu olması durumunda, olası proje plan alternatifi sayısı 3^{180} şeklinde hesaplanmakta, bir başka ifade ile $7,6 \times 10^{85}$ adet olmaktadır. Projede işlem sayısının 200 olması durumunda da alternatiflerin sayısı $9,45 \times 10^{104}$ e yükselmektedir (Kandil ve El-Rayes, 2006). Bu durum eniyileme problemlerinin önündeki en büyük engel olarak planlayıcıların ve araştırmacıların karşısına çıkmaktadır.

Günümüzün oldukça rekabetçi ve değişken piyasa ortamında, inşaat firmalarının proje planlarını yapabilme ve performansı gözlemleyebilme kabiliyetlerinin kısıtlı imkanlar dahilinde olması optimizasyonu giderek öne çıkarmaktadır (Chen ve Tsai, 2011). Bu bağlamda gelişen teknoloji ve hesaplama yöntemlerindeki yeni yaklaşımlar optimizasyon problemlerine büyük katkı sağlamıştır. Literatürde bu kombinatorial optimizasyon probleminin çözümü için matematiksel yöntemlerden sezgisel, metasezgisel ve evrimsel yöntemlere kadar değişik pek çok model önerilmiştir.

Bu tez kapsamında, karmaşık problemleri çözmeye kullanılan metasezgisel algoritmaların inşaat projelerinde karşılaşılan süre-gider eniyileme problemi üzerindeki sonuçlarının değerlendirilmesi hedeflenmektedir.

Proje temelinde yürütülen bütün işlerde olduğu gibi özellikle inşaat projelerinde de işgücü, malzeme ve makine gibi kaynaklar uygulamada sınırlı olarak bulunur. Toplam proje giderine esas teşkil eden bu kaynakların kullanımı süre bakımından da ayrı bir değerlendirme ve plana ihtiyaç duymaktadır. Bu noktada proje yöneticilerinin proje bütçesi ile proje süresi arasında bir denge kurabilmesi proje planlamasının en önemli aşamalarındandır (Albayrak ve Özdemir, 2016).

Optimizasyon olarak da adlandırılan eniyileme alanında yeni tekniklerin birbiri ardına ortaya konulduğu ve varolan tekniklerin durmaksızın geliştirildiği bilinmektedir. Bu nedenle günümüzde eniyilemeye yönelik çalışmalarda karşılaşılan en önemli güçlük hangi problem türü sözkonusu olduğunda hangi algoritmadan daha iyi sonuç alınacağına önceden bilinmemesidir. Oysa eniyileme problemlerinin çözümünde doğru algoritmanın seçilmesi güvenilir sonuç çıktılarını için son derece önemlidir. Bu bağlamda yapılan tez çalışmasında süre-gider eniyilemesinde kullanışlı algoritmaları araştırmak ve kıyaslama yaparak proje türlerine göre öneriler getirmek amaçlanmıştır.

Çalışmanın ilk bölümü olan “Giriş” bölümünde tez konusuna ilişkin kısa bir bilgi verilmiş, ikinci bölüm olan “Literatür Araştırmasında” da bu alanda daha önce yapılmış çalışmalar sunulmuştur. Üçüncü bölüm “Materyal ve Yöntem” ise çalışmaya konu olan süre-gider eniyilemesi hakkında teorik ve uygulamalı bilgi vermektedir. Dördüncü bölümde ise “Metasezgisel Yöntemler” başlığı altında öncelikle genel bilgiler ve ardından

seçilen iki yöntem olan genetik algoritmalar (GA) ve parçacık sürü optimizasyonu (PSO)'ya ilişkin detaylı incelemeler verilmektedir. Beşinci bölüm olan “Analitik ve Metasezgisel Yöntemler Kullanarak Süre-Gider Eniyileme Uygulaması” seçilen yöntemlerin örnek problemler üzerindeki kullanımını göstermektedir. Altıncı bölüm “Bulgular ve Tartışma” ile yedinci bölüm “Sonuçlar ve Öneriler” çalışmadan elde edilenlerin sunulması, yorum ve değerlendirmelerin aktarıldığı bölümlerdir. Çalışmanın sonunda yararlanılan ve referans alınan kaynaklara yer verilmiştir.



2. LİTERATÜR ARAŞTIRMASI

Bu bölümde süre-gider eniyilemesine ait bir literatür taraması sunulmaktadır. Konuyla ilgili literatürde mevcut olan çalışmalar matematiksel programlama gibi geleneksel yöntemlerden evrimsel algoritmalara kadar değişik kategoriler altında araştırılmış ve incelenmiştir.

Proje serimlerindeki süre-gider ilişkisi başlangıçta işlem süreleri ile işlem giderleri arasındaki doğrusal bir fonksiyon olarak varsayılmaktaydı ve esas amaç işlem sürelerini belirlemek ve proje giderlerini minimize etmek için işlemleri çizelgelemektir (Vanhoucke vd., 2007). İşlemlere ait gider fonksiyonlarının içbükey, dışbükey ve kesikli gibi değişik tipleri yıllar içerisinde ortaya atılmış ve incelenmiştir. Gerçek hayattaki projeleri en iyi yansıttığı düşünülen ve en pratik uygulamaya sahip yaklaşım ise işlem süreleri ile giderler arasındaki bağı kesikli süre-gider ilişkisi olarak modellenmesidir (Golzarpoor, 2012).

2.1. Matematiksel Programlama, Doğrusal Modelleme ve Deterministik Yaklaşım

Kelly (1961) işlemlere ait süre-gider ilişkisini doğrusal varsayarak formüle etmiştir. Hendrickson ve Au (1989) ile Pagnoni (1990), doğrusal programlamayı süre-gider problemini çözüme bir araç olarak kullanmışlardır. Doğrusal süre-gider ilişkisinde matematiksel programlama yaklaşımları uygun olmakla beraber kesikli süre-gider ilişkilerinde çözüme yardımcı olamamaktadır. Meyer ve Shaffer (1963) ile Patterson ve Huber (1974) hem doğrusal hem de kesikli bağ içeren süre-gider problemini karışık tamsayılı programlama kullanarak çözmüşlerdir. Bununla beraber serimdeki faaliyet sayısı arttıkça problemin kompleks hale gelmesinden ötürü tamsayılı programlama çok miktarda hesaplama adımı gerektirmektedir.

Süre-gider probleminin deterministik olarak ele alındığı diğer bazı çalışmalara Phillips ve Dessouky (1977), Talbot (1982), Liang vd. (1995) ile Hegazy (1999) örnek olarak verilebilmektedir. Deterministik yaklaşımlara dair genel bir derleme çalışması Chen ve Tsai (2011) tarafından yapılmıştır. Belirsiz işlem sürelerinin ele alındığı süre-gider eniyileme problemleri için bazı stokastik modeller geliştirilmiştir. Charnes vd. (1964), Mon

vd. (1995), Golenko-Ginzburg ve Gonik (1997), Gutjahr vd. (2000), Feng vd. (2000) ile Ke vd. (2009) bunlara örnek olarak verilebilir.

Burns vd. (1996) eniyileme eğrisinde alt sınırı bulabilmek için doğrusal programlama, istenilen süredeki kesin çözümü bulabilmek için de tamsayı programlama kullanan melez bir yaklaşım ele almışlardır. Robinson (1975), Elmaghraby (1993) ve De vd. (1995) serilere veya paralel alt serimlere ayrılabilen faaliyet şebekelerinde süre-gider problemi çözümü için dinamik programlama kullanmışlardır. De vd. (1997) süre-gider probleminin teorik altyapısını ortaya koyarak matematiksel karmaşıklığına ve NP-zor (polinom zamanda çözülemeyen) sınıfta yer aldığına vurgu yapmışlardır.

Süre-gider eniyilemesine yönelik en düşük maliyeti veren proje teslim süresinin tespiti için Yavuz (1992) tarafından Basic dilinde bir bilgisayar programı önerilmiştir. Çalışmadan elde edilen bulgularla gerçek durumun paralellik gösterdiği vurgulanmaktadır.

Literatür incelendiğinde erken dönem çalışmalarındaki amacın süre ya da giderden birini minimize etmek olduğu bilinmektedir. Daha sonraki yıllarda ise çok amaçlı optimizasyon yaklaşımlarının uygulandığı görülmektedir. Bu alandaki çalışmalardan bazıları şunlardır: Feng vd. (1997, 2000), Zheng vd. (2004, 2005), Kandil ve El-Rayes (2006), Azaron ve Tavakkoli-Moghaddam (2007), Ng ve Zhang (2008), Xiong ve Kuang (2008), Afshar vd. (2009) ve Castro-Lacouture vd. (2009).

2.2. Metasezgisel Yöntemler

Metasezgisel algoritmalar özellikle son 20 yılda ortaya çıkmış ve araştırmacılar tarafından sürekli geliştirilmiştir. Proje planlama alanında da önemli bir yere sahip metasezgisel algoritmaların literatürdeki kapsamı dikkate alındığında, tezin bu bölümünde son on yıla yoğunlaşarak güncel çalışmaların biraraya getirilmesi hedeflenmiştir.

Vanhoucke vd. (2007) problemi kesikli süre-gider eniyilemesi olarak ele almış ve sezgisel yöntem prosedürlerini tanıtmışlardır. Tareghian ve Taheri (2007) kesikli süre-gider problemi için elektromanyetik dağılım aramasını tanıtmıştır.

Huang vd. (2008)'nin çalışmasında probleme modifiye edilmiş karınca kolonisi algoritması önerisi getirilmiştir. Klerides ve Hadjiconstantinou (2010) problemdeki belirsizliklere işaret ederek stokastik programlama kullanmışlardır. Salmasnia vd. (2011) problem için gürbüz modeli öneri olarak sunmuşlardır.

Shrivastava vd. (2012) süre-gider eniyileme problemine karınca kolonisi yöntemini uygulayarak çok amaçlı optimizasyon çözümleri geliştirmişlerdir.

Golpira ve Hejazi (2014) senaryo tabanlı çok amaçlı stokastik modelleme ile probleme çözüm aramışlardır. Tavana vd. (2014) süre-gider eniyilemesine üçüncü bir kriter olan kaliteyi de ilave etmişlerdir. Böylelikle ortaya koydukları yeni yaklaşımda çok modlu çok amaçlı bir problemi incelemişlerdir. Monghasemi vd. (2015) inşaat projeleri için üç kriterli karar verici eniyileme modeli geliştirmişlerdir.

Yukarıda değinilenler dışında 2007 yılından bu yana yapılmış çalışmalar Çizelge 2.1. ile özetlenmiştir.

Çizelge 2.1. Süre-gider eniyileme probleminde son on yıla ait literatür özeti

| Yıl | Yazar(lar) | Yöntem | Notlar |
|------|----------------------|----------------------------------|---|
| 2007 | Afshar vd. | Çok kolonili karınca algoritması | Süre-gider probleminin inşaat projelerine uygulamasında süre ve giderden ödün vererek kaliteyi maksimize etmeye yönelik bir model geliştirilmiştir. |
| 2007 | Kasaeian vd. | Geliştirilmiş çok amaçlı GA | Süre-gider problemi için oluşturulan model GA ve adaptif ağırlık algoritması ile çözülerek karşılaştırma yapılmıştır. |
| 2007 | Pathak ve Srivastava | Bulanık-GA | Gerçek hayat belirsizlikleri altında çok amaçlı süre-gider eniyileme problemi önerilen algoritma ile çözülmüştür. |
| 2007 | Yang | Yeni seçkinci PSO | Klasik PSO algoritmasında yerel ve global en iyilerin seçimi sırasında seçkinlerin arşivlenmesi değişikliğini yaparak probleme uygulamıştır. |
| 2008 | Eshtehardian vd. | GA ve bulanık kümeler | Çok amaçlı bulanık süre-gider eniyileme modeline uygun GA geliştirilmiştir. |
| 2008 | Hooshyar vd. | GA | Problemin çözümü için zeki mutasyon operatörlü bir algoritma sunulmuştur. |

Çizelge 2.1. Süre-gider eniyileme probleminde son on yıla ait literatür özeti (devam)

| Yıl | Yazar(lar) | Yöntem | Notlar |
|------|-----------------------------|---|---|
| 2008 | Iranmanesh vd. | Hızlı-PGA | GA'nın geliştirilmesine dayalı bir metasezgisel algoritma oluşturarak çok amaçlı problemlere çözüm aranmıştır. |
| 2008 | Ng ve Zhang | Karınca koloni algoritması | Süre-gider eniyilemesine karınca koloni algoritması ile yaklaşımda bulunulmuştur. |
| 2008 | Pathak vd. | Yapay sinir ağları ile modifiye edilmiş çok amaçlı GA | GA'yı yapay sinir ağları ile güçlendirerek proje faaliyetlerine ait süre-gider çiftlerinin eniyilenmesi hedeflenmiştir. |
| 2008 | Rahimi ve Iranmanesh | Çok amaçlı PSO | Çok modlu, kesikli, süre, gider ve kalite eniyilemesinde PSO yöntemi önerilmiştir. |
| 2008 | Xiong ve Kuang | Karınca koloni algoritması | Adaptif ağırlık algoritması, karınca koloni algoritması ve GA çözümleri karşılaştırılmıştır. |
| 2009 | Afshar vd. | Karınca koloni algoritması | Bastıralamayan ve arşivleyen yeni bir karınca koloni algoritması geliştirilmiştir. |
| 2009 | Eshtehardian vd. | Bulanık tabanlı çok amaçlı GA | Belirsiz ortam koşullarında süre-gider eniyilemesine yeni bir yaklaşım sunulmuştur. |
| 2009 | Ezeldin ve Soliman | Dinamik programlamalı GA | Stokastik optimizasyon problemlerinin kombinatoriyal ve np-zor karakteristiğinin üstesinden gelmeyi amaçlayan bir teknik önermişlerdir. |
| 2009 | Yang | Geliştirilmiş PSO | Yerel optimum noktalarından kaçınarak süre-gider problemini çözen bir algortima geliştirilmiştir. |
| 2010 | Abd El Razek vd. | Basitleştirilmiş GA | Proje gider ve süresini kısaltırken aynı zamanda kaliteyi maksimize edebilmek için AMTCROS adında bir yazılım geliştirilmiştir. |
| 2010 | Zhang ve Xing | Bulanık çok amaçlı PSO | Çeşitli süre-gider kombinasyonlarını değerlendirmede bulanık çoklu-nitelikli metodoloji adapte edilmiştir. |
| 2011 | Abdel-Raheem ve Khalafallah | Evrimsel algorima | Elektronların elektrik devrelerinde en düşük dirençle hareket etme davranışlarını simule eden Electimize isimli bir algoritma geliştirilmiştir. |
| 2011 | Aladini vd. | Çok amaçlı karınca koloni optimizasyonu | Nakit akışlarını gözönüne alarak proje doğrudan giderlerini minimize etmeyi amaçlayan bir algoritma sunulmuştur. |

Çizelge 2.1. Süre-gider eniyileme probleminde son on yıla ait literatür özeti (devam)

| Yıl | Yazar(lar) | Yöntem | Notlar |
|------|---------------------|--|--|
| 2011 | Mohammadi | Çok amaçlı GA | Süre-gider optimizasyonu ile eşzamanlı olarak yerel Pareto optimalite veya yerel bastırılmayan sınırı arayan bir yaklaşım önerilmiştir. |
| 2011 | Mokhtari vd. | Karınca koloni sistemi | İşlem sürelerinin normal dağılımı ve kesikli gider fonksiyonu bulunan yeni çok modlu kesikli süre-gider problemine karınca koloni sistemi uygulanmıştır. |
| 2011 | Salmasnia vd. | Gürbüz yöntemi | Problem süre-gider-kalite ölçütlerine göre ele alınmıştır. |
| 2011 | Li vd. | Güdüleyen GA | Optimizasyon, duyarlılık analizi ve geliştirilmiş GA içeren bir metodoloji önerilmiştir. |
| 2012 | Haque ve Hasin | Bulanık ortamda GA | Bulanık zaman periyotlu belirsiz koşullar altındaki süre-gider probleminin çözümünde daha gerçekçi yaklaşım getirilmiştir. |
| 2012 | Shahsavari Pour vd. | Yeni melez GA | Dilbilgisel değişkenler kullanarak yeni bir melez GA üretilmiş ANOVA metotlu GA ile karşılaştırılmıştır. |
| 2012 | Sönmez ve Bettemir | Melez algoritma | Tavlama benzetimi, kuantum tavlama benzetimi ve GA kullanarak yeni bir melez algoritma önerilmiştir. |
| 2012 | Zhang ve Ng | Karınca koloni sistemi | Adaptif ağırlık algoritması kullanarak oluşturulan, süre ve gider amaçlarının ağırlık değerlerinden oluşan tek bir birleşik amaç fonksiyonuna sahip karınca koloni sistemi temelli bir teknik sunulmuştur. |
| 2013 | Issa ve Eid | GA | İnşaat projelerinde belirli bir süre limiti dahilinde minimum giderle farklı kalite seviyelerini elde edebilen bir model ortaya konulmuştur. |
| 2013 | Mungle vd. | Bulanık küme tabanlı GA | Ödünleşim problemi çözümünde yeni bir yaklaşım tanıtılmıştır. |
| 2014 | Afruzi vd. | Çok amaçlı emperyalist rekabetçi algoritma | Kesikli süre, gider ve kalite eniyilemesine, mod tanımlamaları ve kısıtlı kaynak koşulları ile bir yaklaşım geliştirilmiştir. |
| 2014 | Zhang | Çok modlu karar verme modeli | Kısıtlı kaynak ortamında bulanık süre-gider-kalite-çevre eniyilemesi yapılmıştır. |

Çizelge 2.1. Süre-gider eniyileme probleminde son on yıla ait literatür özeti (devam)

| Yıl | Yazar(lar) | Yöntem | Notlar |
|------------|-----------------------|-------------------------|---|
| 2015 | Kaveh vd. | CSS ve CBO | Süre-gider problem çözümü için yeni metasezgiseller tanıtılmıştır. |
| 2015 | Pathak ve Srivastava | Bulanık-ANN-HMH | Bulanık ortamda süre ve gider optimizasyonu için proje planlayıcılarına yeni bir teknik geliştirilmiştir. |
| 2015 | Zareei ve Hassan-Pour | NSGA-II, MOSA and MOPSO | Üç evrimsel algoritma kullanılarak geliştirilen bir matematiksel model, çok amaçlı çizelgeleme probleminde Pareto çözüm setini bulmada uygulanmıştır. |

3. MATERYAL VE YÖNTEM

3.1. Projelerde Kaynakların Sınıflandırılması

Projelerde her bir adımın yani işlemin gerçekleştirilmesi için bir araya getirilip kullanılan unsurlar kaynaklar olarak adlandırılmakta ve özelliklerine göre sınıflandırılmaktadırlar.

Zaman bakımından sınıflandırma kaynak sınıflandırmalarının bir türüdür ve yenilenebilir yani kullanımla tükenmeyen kaynaklar, kısmi yenilenebilir kaynaklar, yenilenemez yani kullanım sonucu tükenen kaynaklar ve hem zaman hem de miktar bakımından yani çift yönden kısıtlı kaynaklar olarak dört gruba ayrılmaktadır.

Yenilenebilir kaynakların bir zaman birimi içindeki toplam kullanım miktarı kısıtlı olmakla birlikte kullanım ile tüketilmezler. Bu tip kaynaklar iş makineleri ya da bilgisayarlarda olduğu gibi belirli bir faaliyet çerçevesinde kullanıldıktan sonra faaliyet sona erince tekrar kullanıma hazır duruma gelmektedirler.

Genellikle sarf malzemeler olarak adlandırılan yenilenemez kaynaklar ise kullanımları esnasında tüketilmektedirler. Proje süresi boyunca yenilenemez kaynakların kullanılabilir toplam miktarı sınırlıdır. Bazı kaynaklar ise hem bir zaman dilimi içinde kullanım miktarı bakımından hem de proje süresi boyunca toplam tüketimi bakımından sınırlıdır. Kaynakların her iki bakımdan da kısıtlı olması çift yönden kısıtlı kaynak olarak nitelendirilmektedir. Örneğin proje finansmanları belli süreler dahilinde kısıtlanabilir olmasının yanı sıra proje süresince de toplam harcama açısından kısıtlanabilir olduğundan çift yönden kısıtlı bir kaynak tanımına uygundur.

Bir takım kaynaklar ise kısmi yenilenebilir kaynaklar türündedir. Bu durumda belirli zaman aralıklarında kaynak kullanımlarının planlanması öne çıkmaktadır. İş gücünün planlanması esnasında vardiyalara göre kaynak planlaması bu tip kaynak kullanımına bir örnek teşkil etmektedir.

Kaynakları bölünebilen ve bölünemeyen kaynaklar olarak sınıflandırmaya tabi tutmak da mümkündür. Belirli bir sayısı olan veya miktar cinsinden ifade edilebilen, yani bölünebilen kaynaklara, ayrık kaynaklar denilmektedir. İş makineleri ve teçhizatlar bu tür kaynaklardandır. Enerji gibi bölünemeyen türden kaynaklara ise sürekli kaynaklar adı verilmektedir.

3.1.1. Kısıtlı kaynak kullanımı

İşgücü, makine gibi kaynaklar uygulamada sınırlı olarak bulunmaktadır. Genelde belirli sayıda, değişik konularda uzmanlaşmış işgücü ve makineden oluşan bu gruplar ekip olarak adlandırılır. Ekiplerin boyutu ve sayısı işin büyüklüğüne bağlı olabileceği gibi piyasanın kaynak arzına da bağlı olabilir. Kaynakların kullanımı yapılan işin ya da üretimin tipine göre çeşitlilik göstermektedir. Özellikle süreklilik beklenen akım tipi üretimlerde kaynakların boş bekletilmemesi temel amaçtır.

Proje tipi üretimlerde ise işgücü ve makine arzı genelde taşeronluk biçiminde olduğundan boş bekleme sorunu olmaz ve böyle durumlar kaynağı arz eden tarafından değerlendirilir. Bu tipteki üretimlerde amaç üretimin mevcut kaynaklarla minimum sürede bitirilmesidir.

Proje tipi üretim planlamalarında gelir ve hammadde kaynakları, işgücü ve makine gücünden ayrı tutulmaktadır. Bunun sebebi periyodik olarak temin edilip belli periyot içinde tüketilmesidir. Hammaddenin yapılacak işlemlere bağlı olarak önceden belirlenip alınması mümkün olmasına karşın gelir kaynağının planlaması her zaman net olarak yapılamaz. Bu nedenle periyot boyunca mevcut geliri aşmayacak şekilde işler planlanır. Bu durum bir bakıma, gelirin de değişken kısıtlı kaynak olarak tanımlanmasını gerektirir.

Kaynaklar projelerde 3 şekilde kullanılmaktadır:

- Tüm projede ve her işlemde birden fazla kaynak kullanılabilir.
- Tüm projede birden fazla fakat her işlemde yalnızca bir kaynak kullanılabilir.
- Tüm projede ve her işlemde yalnızca bir tek kaynak kullanılabilir.

Kaynak kullanımının eniyilenmesine (optimizasyonuna) ilişkin modeller 2 grupta toplanan yöntemlerle çözümler.

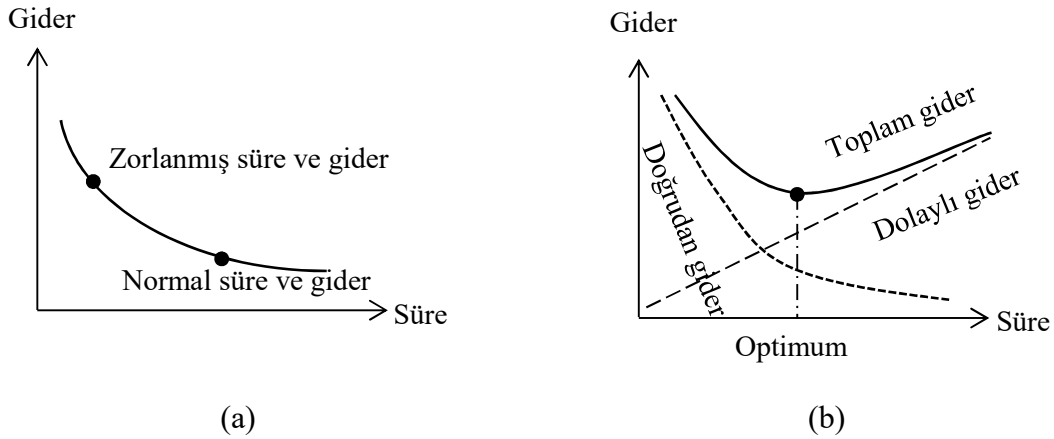
- Analitik teknikler (Doğrusal programlama, Doğrusal olmayan programlama, Tam sayılı programlama, Dinamik programlama vb.)

- Metasezgisel teknikler (Genetik algoritmalar, Parçacık sürü optimizasyonu, Karınca kolonisi, Yapay arı algoritması, Ateş böceği optimizasyonu vb.)

Çözüm tekniklerine ilişkin detaylı bilgi sonraki bölümlerde yer almaktadır.

3.2. İşlem Süresi-Kaynak Kullanımı İlişkisi

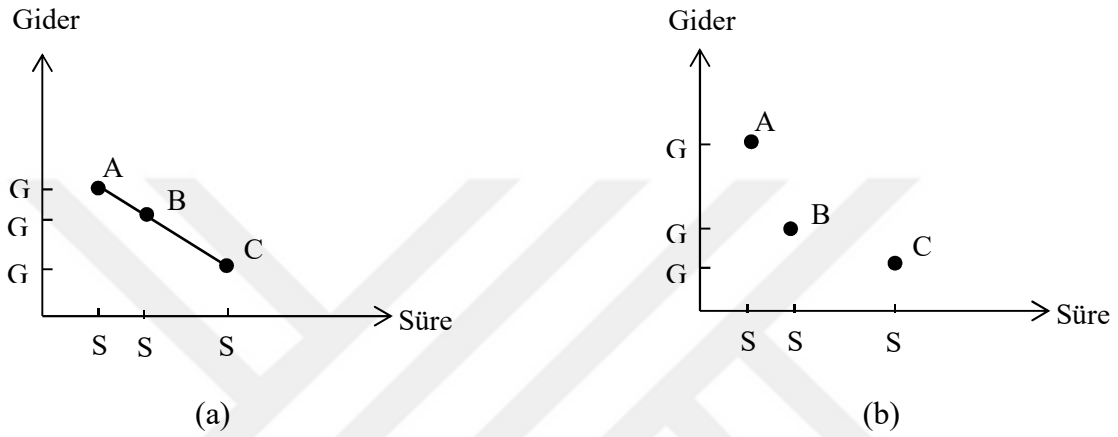
Bir işlemin gerçekleştirilme süresi ile bu işlem için kullanılan kaynaklar dolayısıyla gider arasında sıkı bir ilişki vardır. Birim zamanda kullanılan kaynak miktarının artışının doğrudan maliyeti yükseltmesi, işlem süresini ise azaltması işlem süresi-kaynak kullanımı ilişkileri bakımından çoğunlukla öngörülebilir bir durumdur. Bu öngörü dikkate alındığında iki unsur arasındaki ilişkinin en geleneksel hali şekildeki gibi modellenebilir (Şekil 3.1.a.).



Şekil 3.1. Süre-gider ilişkisi grafiği (a) İşlemlerdeki durum (b) Projelerdeki durum

Genelde bir işlemin tamamlanabilmesi için geçen süre ile meydana gelen maliyet arasında ters orantılı bir bağıntı vardır. Bu sebeple daha az maliyetli kaynak kullanımı tercih edildiğinde işlemin bitiş süresi gecikmektedir (Feng vd., 1997). Bir projede toplam gider, doğrudan ve dolaylı giderlerin toplanması ile elde edilmekte ve en düşük toplam maliyete karşılık gelen optimum süre değeri bulunabilmektedir (Şekil 3.1.b.).

Süre-gider ilişkisi kullanılan kaynağın özelliğine göre sürekli ya da kesikli bir fonksiyon şeklinde ifade edilmektedir (Şekil 3.2.a ve Şekil 3.2.b.). Kesikli fonksiyon durumunda, fonksiyonun her bir süre-gider çiftine karşı gelen noktası bir mod olarak nitelendirilir ve ayırık noktalardan oluşan bir küme oluşturulabilir. Bir veya daha fazla sayıda işlemi birden fazla kaynak kullanım alternatifine sahip yani birden çok moda sahip proje planlama problemleri çok modlu problemler olarak nitelendirilirler.



Şekil 3.2. Süre-gider fonksiyonu grafiği (a) Sürekli fonksiyon (b) Kesikli fonksiyon

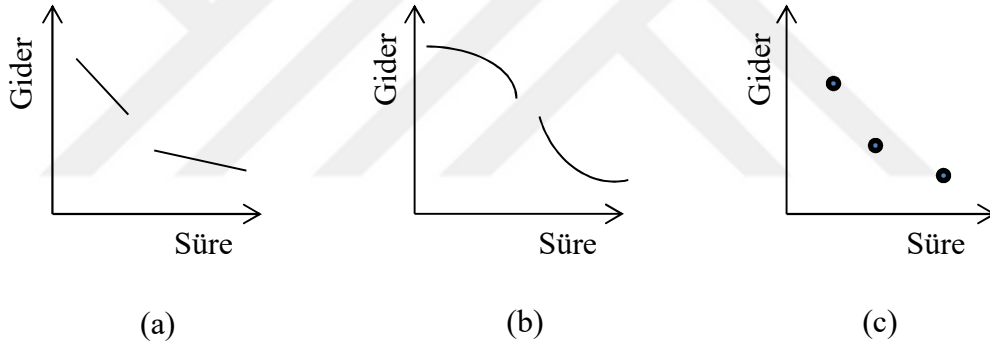
Süre gider analizi inşaat projelerinde kritik yol üzerindeki bazı işlemlerin daha hızlı ancak daha maliyetli modlarının da gözönüne alınarak proje planlayıcıları tarafından en uygun seçimin yapılabilmesini sağlamaktadır. Bir işlemin hesaplanan en az giderle (normal gider) tamamlandığı süre “normal süre”, bir işlemin tamamlanabileceği en az süre “hızlandırılmış süre”, hızlandırılmış süreye karşılık gelen gider de “hızlandırılmış gider” olarak tanımlanmaktadır. Normal ve hızlandırılmış noktalar arasındaki herhangi bir ara noktadaki değer süre-gider ilişkisinden hesaplanabilmektedir (Liao vd., 2011). Literatürde “crashing” olarak adlandırılan uygulama Türkçe kaynaklarda “hızlandırma”, “zorlama”, “kısaltma” ve “indirgeme” karşılıklarını almakta ve tamamı süre-gider eniyilemesi problemi bakımından aynı anlamda kullanılmaktadır.

Kritik yol yöntemi bitiş süresi kısıtlanmamış projelerin çizelgelendirilmesinde kullanılan bir yöntemdir (Liao vd. 2011). Kritik yol yöntemini sabitlenmiş bir bitiş tarihi ile ya da planlamanın gerisinde kalmış bir proje için kullanmak istenildiğinde projeyi zamanında bitirebilmek için süre gider analizine ihtiyaç duyulur. Süre-gider analizinde kritik yol üzerindeki bazı işlemlerin daha kısa sürelerle sahip alternatifleri getirilir. Böylece

zamandan kazanç elde edilmektedir. Buna ilave olarak Siemens (1971)'e göre maliyetten tasarruf etmek adına da kritik olmayan işlemlerde gevşetmeler yapılabilir.

3.3. Süre-Gider Eniyilemesi

Proje hızlandırma analizlerinin amacı belirli bir bitiş tarihini sağlayacak maliyetin en küçük değerini bulmaktır (Yang, 2007). Eğer işlem sayısı fazla ve işlemlere ait süre ile gider değişimleri doğrusallık göstermiyor ise uygulamada bu problem oldukça karmaşıklık içermektedir. Bunun sebebi, projeye ait işlemlerin süre-gider ilişkisini gösteren fonksiyonların sıklıkla doğrusal olmayan, kesikli hatta bazı durumlarda parçalı olmasından kaynaklanmaktadır. Aşağıdaki şekilde değişik türlere ait süre-gider ilişkisini gösteren grafikler verilmektedir (Şekil 3.3.).



Şekil.3.3. İşlemlerde süre-gider fonksiyon türleri:

(a) doğrusal (parçalı) (b) doğrusal olmayan (iç ve dışbükey melez) ve (c) kesikli.

Projelerde özel bir bitiş noktasını karşılamak adına çeşitli yaklaşımlar denenmektedir. Bu yaklaşımlardan en genelgeçer olanı, ilave bir maliyeti üstlenmek suretiyle, kullanılan kaynakların geliştirilmesidir. Örneğin işçi veya ekipmanlarda nicelik ya da nitelik yönünden değişiklikler yapmak, projeye ilave maliyet yüklese de, toplam proje süresinin kısılmasına olumlu etki yapmaktadır. Proje hızlandırma olarak ifade edilen bu yaklaşımların çeşitli türleri bulunmaktadır.

Proje hızlandırma probleminde bir işlemin maliyeti genellikle proje süresi bazında değişiklik göstermeyen, yalnız işlem süresinin kısalıp uzamasına bağlı değişen şekilde sınırlanmış bir fonksiyon ile ifade edilmektedir.

Kelly (1961) ve Fulkerson (1961) gibi bu alanda yapılan ilk çalışmalarda işlemlere dair süre-gider fonksiyonları doğrusaldır. Bu durum çözüm prosedürleri bakımından doğrusal programlama tekniklerinde etkin bir gelişmeye olanak tanımıştır. Takip eden çalışmalarda da bu doğrusal varsayım süre-gider fonksiyonlarının değişik formlarda ortaya konulması ile ileri bir adıma taşınmıştır. Örneğin Falk (1972) içbükey, Foldes (1993) dışbükey, Moder vd. (1995) melez iç ve dışbükey, Deckro vd. (1995) kuadratik formda süre-gider fonksiyonları tanımlamışlardır. Geçmiş çalışmalar göstermektedir ki gerçekte doğrusal olmayan fonksiyonlar içermesine rağmen doğrusal programlama çözümlerinin etkin hesap kabiliyeti dolayısıyla doğrusal şekilde modellenen problemler kullanılmaktadır. Bu durum problemi daha anlaşılır hale getirirse de çözüm bakımından karmaşıklığını arttırmakta ve geleneksel optimizasyon tekniklerini, optimum sonucu arama bakımından yetersiz kılmaktadır. Bu noktada proje hızlandırma gibi kompleks problemlere optimal çözümler getirebilme yeteneği nedeniyle genel ismiyle “Evrimsel Algoritmalar” (EA) denilen modern teknikler ortaya çıkmıştır.

Bütün EA teknikleri arasında “Genetik Algoritmalar” (GA) proje planlama alanında geniş yer bulan bir yöntemdir. Rastgele bir başlangıç popülasyonu üzerinden mutasyon, çaprazlama ve seçim operatörleri aracılığıyla yeni nesiller üreterek daha iyi sonuçlara varmayı hedefleyen GA'nın özü Darwin'in evrim teorisine dayanmaktadır (Goldberg, 1989). Feng vd. (1997), Leu ve Yang (1999), Li ve Love (1997), Li vd. (1999) gibi çalışmalar geçmişte GA'nın proje planlama alanındaki örnekleri olarak gösterilebilir.

Uygulamada bir projeye ait işlemler için farklı süre-gider fonksiyon alternatifleri bulunabilir (Moussourakis ve Haksever, 2004). Gerçek problemler doğrusal, doğrusal olmayan (içbükey veya dışbükey), kesikli ya da parçalı bağıntıların karışımını içerdiğinden doğrusal programlama ya da tam sayılı programlama gibi konvansiyonel teknikler yetersiz kalmaktadır. Diğer bir deyişle bir işleme alternatif yapım metotları uygulanabilir, ancak bu durumda her birinin olası süre-gider değeri farklı aralıkta bulunabilir. Örneğin bir projede herhangi bir işlemin el ile 8-10 günde tamamlanabilecek olmasına karşın, makine kullanımı ile bu süre 2-4 güne düşebilmektedir. Bu durum planlamada aynı işleme ait farklı süre-gider fonksiyonları içeren modları işaret etmektedir.

Eniyileme (optimizasyon), muhtemel pek çok çözüme sahip bir problemdeki en iyi çözümü bulmayı deneyen işlem adımlarıdır. Bütün diğer uygun çözümler arasında optimum olanı bulabilmek özellikle büyük arama uzaylarının tercih edildiği durumlarda oldukça güçtür.

Yüzlerce yada binlerce işlemi barındıran projelerde yapı işlerine ait optimum süre-gider modlarını bulmak pek çok zaman alıcı permütasyonu içerdiğinden yoğun ve karmaşık bir prosedürdür (Liang vd. 1995). Kritik yol yöntemi kullanarak çizelgelemede her bir mod alternatifi için tekrar tekrar hesaplamaya ihtiyaç duyulmaktadır. En hızlı bilgisayarlarda dahi sözkonusu bu işlem yükü çözümü pratiklikten uzak kılmaktadır (Hegazy, 1999).

Gerçekte optimum süre-gider dengesini üretebilmek için mevcut inşaat optimizasyon modelleri kullanılabilmesine rağmen büyük ölçekli projelerde bunların uygulanması yoğun süre ve emek gerektirmesi bakımından kullanışsız kabul edilmektedir (Kandil ve El-Rayes, 2005).

Konvansiyonel yöntemlerden doğrusal programlamanın yanı sıra daha geniş uygulama alanı bulunması ve daha esnek çözümler üretmesi bakımından bu tez kapsamında metasezgisel yöntemler incelenmiştir. Metasezgisel yöntemler içerisinde evrimsel algoritmalar sınıfında genetik algoritma, sürü zekası sınıfında parçacık sürü optimizasyonu ele alınmış ve süre-gider problemine uygulanmıştır.

PSO'nun da içerisinde bulunduğu sürü zekası temelli algoritmalar kuş, balık, karınca ya da arı gibi biyolojik öğelerin kollektif hareketlerinden şaşırtıcı derecede anlamlı davranışlar gözlenmesinden yola çıkılarak üretilmiştir. Bu anlamlı davranışlar canlı gruplarının; barınma, korunma, beslenme vb. hayati ihtiyaçlarını karşılama konusunda hep optimum olanı arama çabalarının göstergesi olarak yorumlanmaktadır.

3.3.1. Süre-gider eniyileme probleminin matematiksel modeli

Bu tez çalışması kapsamında projelerde süre-gider eniyileme problemine ilişkin matematiksel model aşağıda verildiği şekilde oluşturulmuştur.

$$EB_i + s_i - Eb_j \leq 0, \forall j \in S \quad (3.1)$$

$$T = \max_{\forall_i} \{EB_i + s_i\} \leq T_s \quad (3.2)$$

$$EB_i \geq 0, \forall_i \quad (3.3)$$

$$S_i \geq 0, \forall_i \quad (3.4)$$

$$g_i = f(s_i), \forall_i \quad (3.5)$$

Koşulları altında

$$\text{Min} \sum_{\forall} g_i \quad (3.6)$$

Yukarıdaki modele ait ifadelerde yer alan EB_i , i işlemine ait erken başlama zamanını, s_i , i işlemine ait süreyi, T , projenin tamamlanma süresini, T_s , projenin aşılmaması gereken bitiş süresini göstermektedir. (3.3) ve (3.4) nolu bağıntılar erken başlama ile işlem sürelerinin negatif değer alamayacaklarını ifade etmektedir. (3.5) nolu bağıntıda ise i işleminin gideri i işleminin süresine bağlı bir fonksiyon şeklinde gösterilmektedir.

Amaç fonksiyonuna karşılık gelen (3.5) nolu ifade ise bütün işlemlere ait giderlerin toplamının minimize edileceğini göstermektedir.

Yukarıda verilen model için öncelikle (3.3) nolu bağıntıda proje süresine etki eden doğrudan gider eğrisinin elde edilmesine ihtiyaç duyulmaktadır. Bu noktada “Pareto Eğrisi” olarak tanımlanan bir kavram ortaya çıkmaktadır. Süre-gider eniyileme probleminde Pareto eğrisi işlemlere ait süre ya da gider değerlerinden birinin sabit tutulup ötekini bu sabit değer karşısındaki optimal değerine karşılık gelen noktalarının birleştirilmesi ile oluşturulmaktadır.

Süre-gider ilişkisini gösteren fonksiyonlar doğrusal, doğrusal olmayan ve kesikli türde oluşabilir. Doğrusal ve doğrusal olmayan fonksiyonlar, parçalı fonksiyon şeklinde

olabileceği gibi doğrusal olmayan süre-gider fonksiyonları içbükey-dışbükey eğrilerin kombinasyonları şeklinde de olabilmektedir.

Süre-gider fonksiyonunun doğrusal parçalı türde olması halinde aşağıda verilen (3.7) nolu denklem geçerli olmaktadır. Burada a_{ik} ve b_{ik} i. durum ve k. durum için süre-gider fonksiyonunun sırasıyla eğimini ve kesişim noktalarını vermektedir.

$$g_i = a_{ik}s_i + b_{ik} \quad (3.7)$$

Süre-gider fonksiyonu doğrusal olmayan formda ise kuadratik fonksiyon ile tanımlamak mümkündür. Buna göre, aşağıda görülen (3.8) nolu denklemde gider, m_{ik} ve n_{ik} katsayıları ile q_{ik} sabitine bağlı olarak süreyle ilişkilendirilmiştir.

$$g_i = m_{ik}s_i^2 + n_{ik}s_i + q_{ik} \quad (3.8)$$

Üçüncü tür süre-gider fonksiyonları ise belli sayıda seçenekten meydana gelen setlerdir ve aşağıda (3.9) nolu ifade ile gösterilmektedir.

$$\text{Tüm } s_i = S_{ih} \text{ için,} \quad g_i = G_{ih} \quad (3.9)$$

3.3.2. Süre-gider eniyileme problemine serim akış algoritması ile yaklaşım

Ford ve Fulkerson (1956)'nın geliştirdiği “serim akış algoritması” (network flow algorithm), bir serim üzerinde, başlangıç düğümünden bitiş düğümüne kadar olan akışı sınır koşulları altında maksimuma çıkararak eniyilemeye çalışan bir yaklaşımdır (Özdemir, 1987). Serim akış algoritması, Yüksel (2012) tarafından süre-gider eniyileme problemine uygulanmış, bu amaçla öncelikle detayları aşağıda görülen algoritma işleyişi hakkında bilgi verilmiştir.

S_{ij} , ij işlemine ait normal süreyi, s_{ij} , ij işlemine ait zorlanmış süreyi, E_{ij} ise ij işlemine ait süre-gider fonksiyon eğimini göstermek suretiyle serim akış algoritmasının süre-gider problemine uyarlanmasına yönelik algoritma adımları aşağıdaki gibidir:

1. Düğümlerin erken tamamlanma zamanları (ET_i), işlemlerin normal sürelerine (S_{ij}) göre hesaplanmakta, son düğümde elde edilen değer ise toplam proje süresini (T_s) vermektedir.

2. İşlemlere ait bollukları ifade eden B_{ijk} değerleri aşağıda verilen (3.10) ve (3.11) bağıntılarına göre hesaplanmaktadır.

$$B_{ijk} = ET_i + s_{ijk} - ET_j \quad (3.10)$$

$$s_{ij1} = S_{ij} \quad s_{ij2} = s_{ij} \quad (3.11)$$

Serimde göz önüne alınması gereken önemli bir unsur, S_{ij} normal süreli ilk yol ve s_{ij} zorlanmış süreli ikinci yol olmak üzere iki düğüm arasında iki yol olmasıdır. Buna göre ilk yolun kapasitesi E_{ij} , ikincisinin ise (∞) olmaktadır. B_{ijk} , ilk ve ikinci yollardaki bolluk değerlerini göstermekte, her iki yolda da bolluklar sıfır olmadan akım gönderilememektedir.

3. $B_{ijk} = 0$ bulunan yollar kullanılarak i , akımın geldiği düğüm, k^\pm , akımın geldiği yol ve yön, $a(j)$ düğüme gelen akım miktarını göstermek suretiyle düğümler $[i, k^\pm, a(j)]$ şeklinde işaretlenir. Başlangıç düğümünün işareti mutlaka $(-, -, \infty)$ olmalıdır.

İşaretleme i 'den j yönüne doğru yapılmaktaysa aşağıda verilen (3.12), tersi durumda ise (3.13) bağıntısı kullanılmalıdır. Bitiş düğümüne (∞) akım ulaşmışsa algoritma sona ermektedir.

$$[i, k^+, a(j)] \quad a(j) = \min[a(i) \vee E_{ijk} - f_{ijk}] \quad (3.12)$$

$$[i, k^-, a(j)] \quad a(i) = \min[a(j) \vee f_{ijk}] \quad (3.13)$$

(3.12) ve (3.13)'te görülen f_{ij1} ile f_{ij2} sırasıyla 1. ve 2. yolda bulunan akım miktarını gösterirken, E_{ij1} ile E_{ij2} ise sırasıyla 1. ve 2. yolda akım gönderme bedelini ifade etmektedir ve değerleri (3.14) ve (3.15)'te verilmektedir.

$$0 \leq f_{ij1} \leq E_{ij} \quad 0 \leq f_{ij2} \leq \infty \quad (3.14)$$

$$E_{ij1} = E_{ij} \quad E_{ij2} = \infty \quad (3.15)$$

4. Bitiş düğümü de işaretlendikten sonra akımın geldiği yol buraya gelen akım miktarı kadar artırılır veya azaltılır. Bu işlemin ardından 3. adıma dönülür.

5. İşaretleme yapılmamışsa işlemler K_1 ve K_2 şeklinde kümelenir. Bu kümeler yardımıyla (3.16), (3.17) ve (3.18)'de verilen Δ_1 ve Δ_2 değerleri ve sonrasında Δ değeri elde edilmektedir.

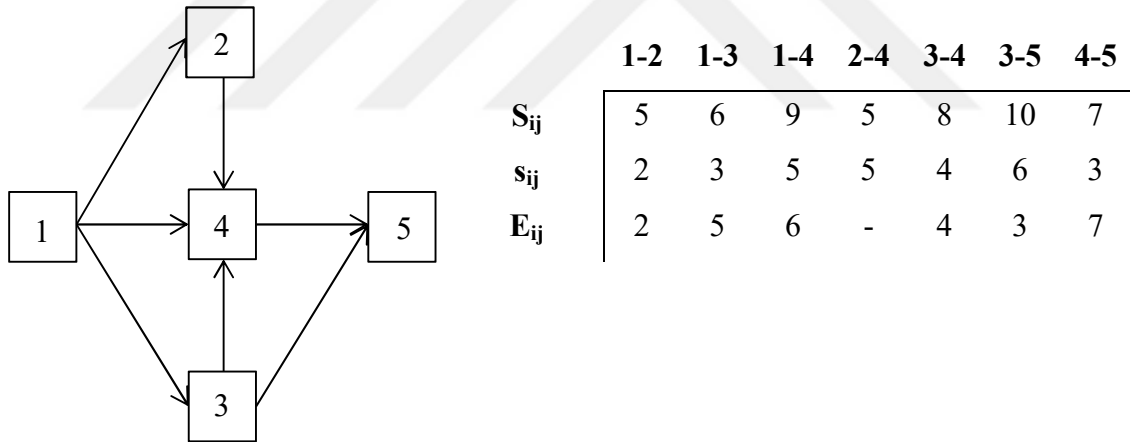
$$K_1 = [(ijk)/i \text{ işaretli}, j \text{ işaretsiz}, B_{ijk} < 0] \quad \Delta_1 = \min_{K_1} [\infty \vee | - B_{ijk} |] \quad (3.16)$$

$$K_2 = [(ijk)/i \text{ işaretsiz}, j \text{ işaretli}, B_{ijk} > 0] \quad \Delta_2 = \min_{K_2} [\infty \vee | - B_{ijk} |] \quad (3.17)$$

$$\Delta = \min [\Delta_1 \vee \Delta_2] \quad (3.18)$$

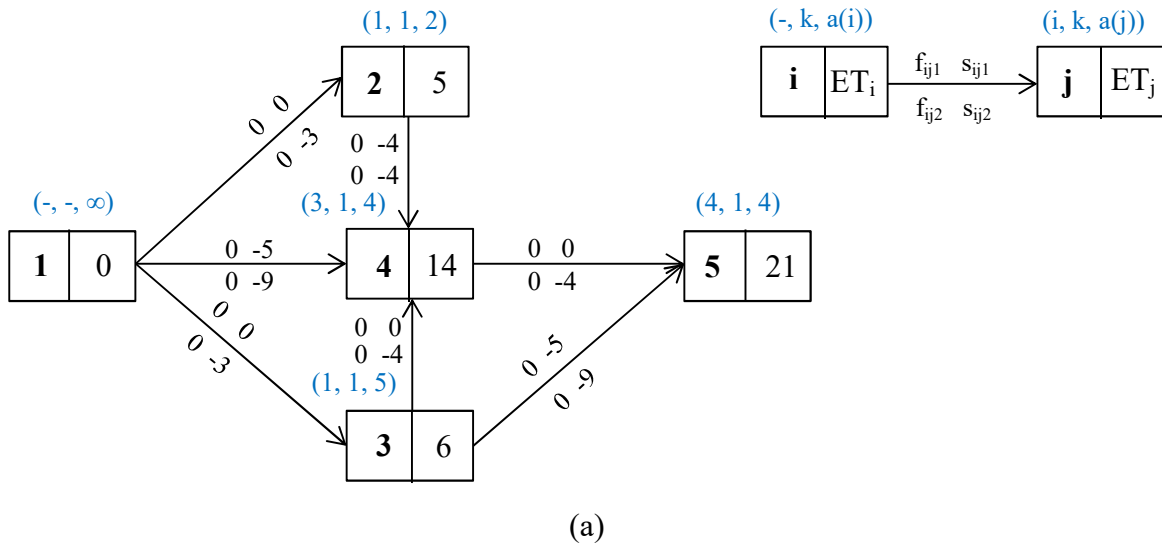
6. İşaretlenmemiş düğümlerin ET_i değerlerinden Δ çıkarılarak 2. adıma dönülmektedir.

Yukarıda işleyişi anlatılan serim akış algoritması 5 işlemlilik bir serim üzerinde uygulanarak örneklendirilmektedir. Şekil 3.4.'te uygulamada kullanılan serim ve işlemlere ait süre-eğim bilgileri görülmektedir.



Şekil 3.4. Akış algoritmasıyla süre-gider eniyilemesi örnek serimi (Yüksel, 2012)

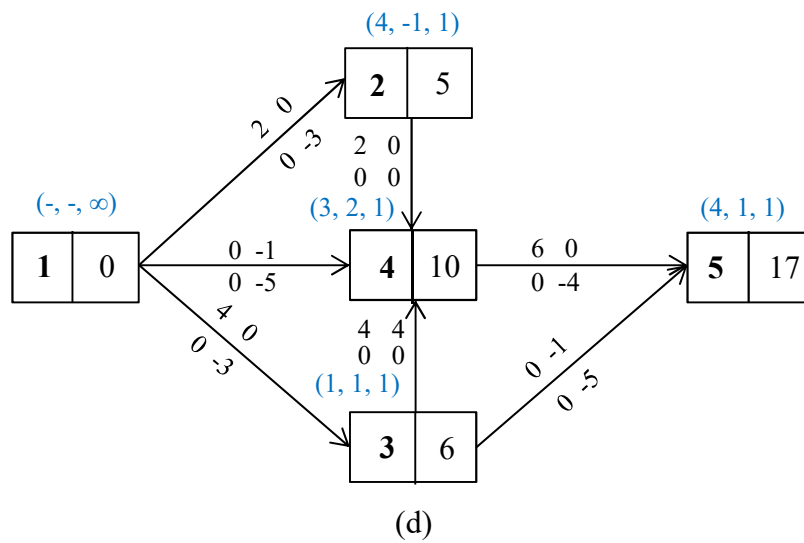
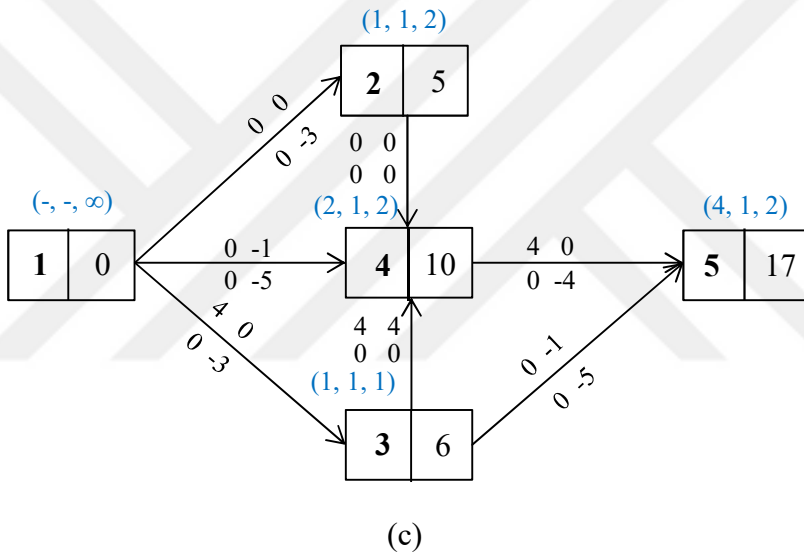
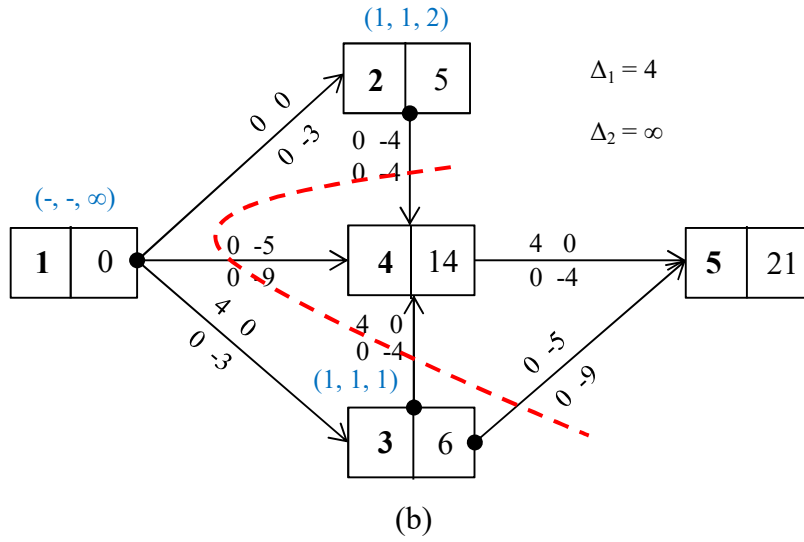
Serim akış algoritmasına başlayabilmek için öncelikle ET_i zamanları Şekil 3.5.a.'da görüldüğü üzere bulunmakta ve B_{ijk} değerleri hesaplanmaktadır (1. ve 2. adım). Ardından düğümler sırasıyla $(-, -, \infty)$, $(1, 1, 2)$, $(1, 1, 5)$, $(3, 1, 4)$ ve $(4, 1, 4)$ şeklinde işaretlenmektedir (3. adım). Bu işaretleme, örneğin 5 nolu düğüme 4 nolu düğümden 1. yol üzerinden 4 birim akım gönderilir anlamı taşımaktadır.



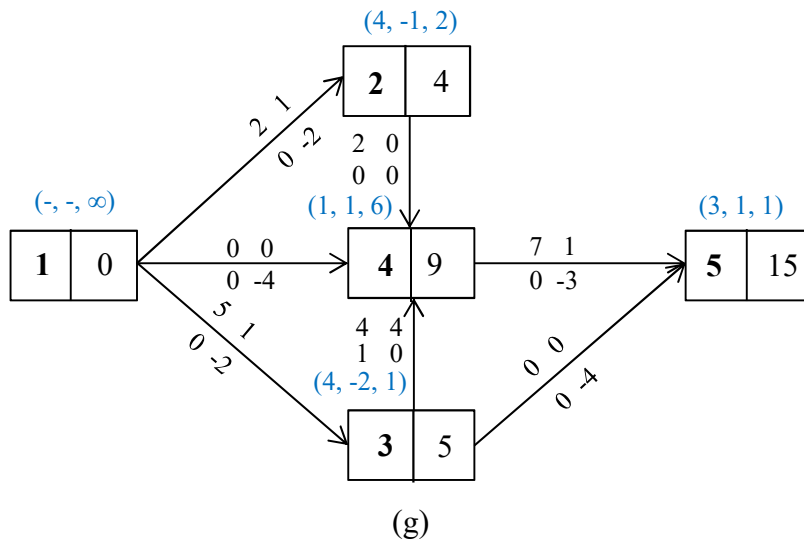
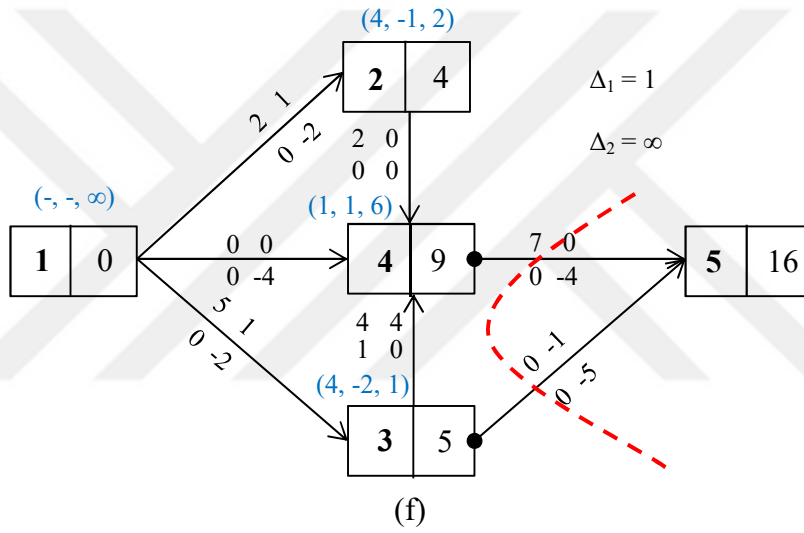
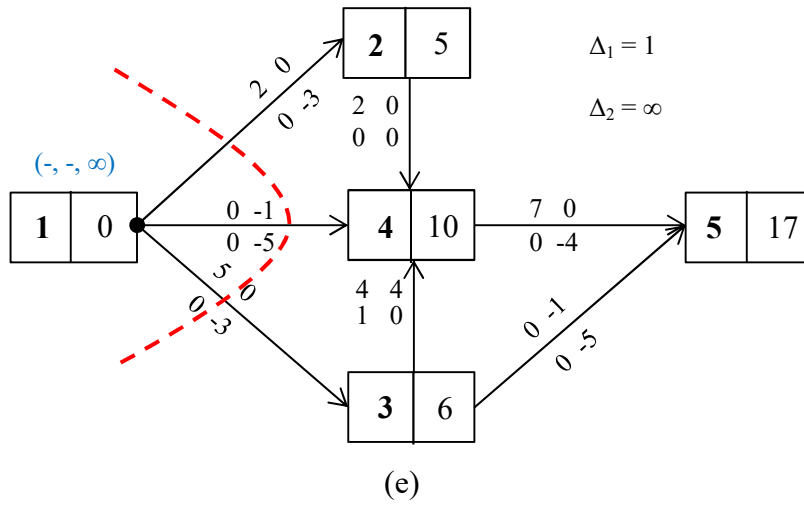
Şekil 3.5. Serim akış algoritması ile süre-gider eniyilemesi

Şekil 3.5.b.'den de görüleceği üzere son düğüme ulaşan akımın takip ettiği 1-3, 3-4, 4-5 yolundaki değerleri f_{131} , f_{341} ve f_{451} 4 birim arttırılmaktadır (4. adım). Yalnız 1, 2 ve 3 düğümleri işaretlenebilirken son düğüm işaretlenememektedir. 5. adıma geçilerek S_1 ve S_2 kümeleri bulunur. S_1 'de 1-4, 2-4, 3-4 ve 3-5 işlemleri mevcut iken S_2 boştur. $\Delta = 4$ olarak alınmaktadır. İşaretsiz düğüm ET_i 'lerinden Δ çıkarılarak 2. adıma dönülmelidir.

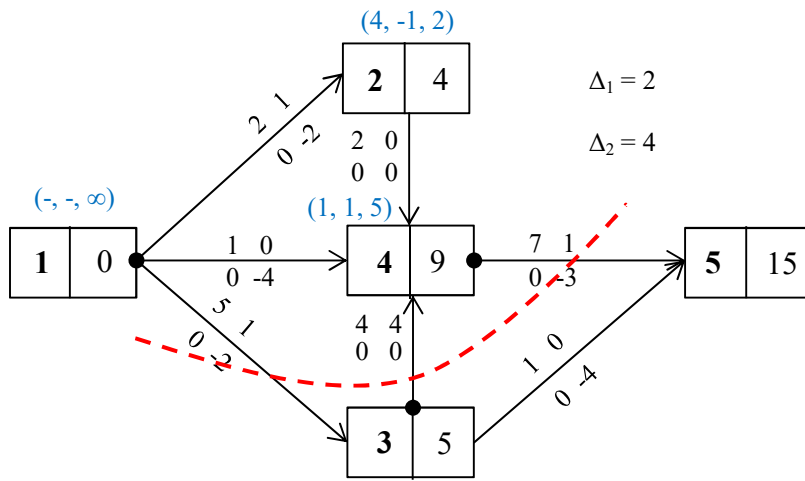
Sırasıyla algoritma adımları bitiş düğümüne akım ulaşmıncaya kadar devam ettirilmektedir. Bu aşamalar Şekil 3.5.(c, d, e, f, g, h, i, j, k, l, m)'de gösterilmektedir. Bu esnada Şekil 3.5.h.'ye dikkat edilirse 2 düğümüne s_{ijk} değerleri 0 olmadığından, 1'den akım gönderilemeyeceği görülmektedir. Benzer şekilde 3 düğümünde de aynı durum söz konusudur. 4 düğümüne, 1'den, 1. yolu kullanarak 5 birim akım gönderilebilmektedir. 2-4 arasında, 12. yoldan daha önce 2 birim akım gönderilmiş ($f_{241} = 2$) ve $S_{ij1} = 0$ olduğundan, 4 düğümünden 2'ye ters akım gönderilmektedir. Diğer bir ifade ile gönderilen akımın bir kısmı geri çekilmektedir. 2 düğümü (4, -1, 2) olarak işaretlenirken, 3 ve 5 düğümleri işaretlenemez. $K_1 = \{(1-3), (4-5)\}$ ve $K_2 = \{(3-4)\}$ olarak bulunmakta ve buradan da $\Delta_1 = 2$, $\Delta_2 = 4$ ve $\Delta = 2$ elde edilmektedir. Sonrasında algoritma 6. adım ile sürdürülmektedir. Algoritmada yalnızca bolluğu olan işlemler üzerinden akım gönderilmektedir. Bir yol üzerinde gönderilebilecek maksimum akım gönderildikten sonra, yolun en az giderli işleminin süresi, düğümlerin ET_i değerlerinin değiştirilmesi yardımıyla kısaltılmaktadır. Bu noktada 5. ve 6. adımlar uygulanmaktadır. Bu uygulama tüm işlemler kritik hale gelinceye kadar devam ettirilmektedir.



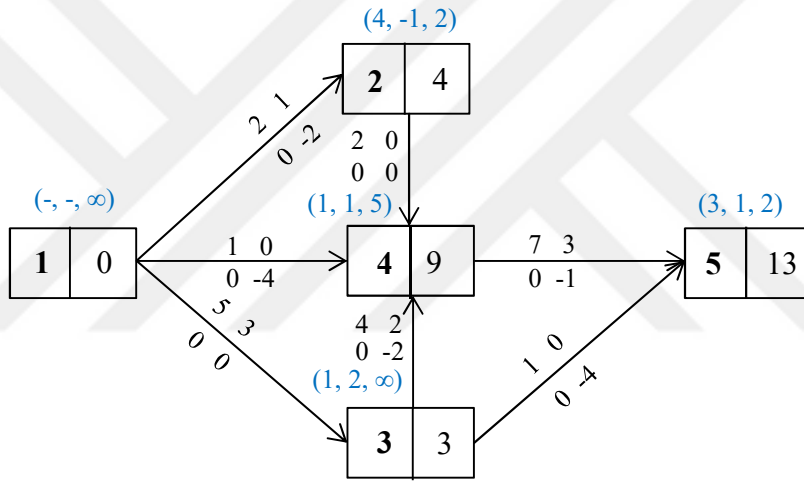
Şekil 3.5. Serim akış algoritması ile süre-gider eniyilemesi (devam)



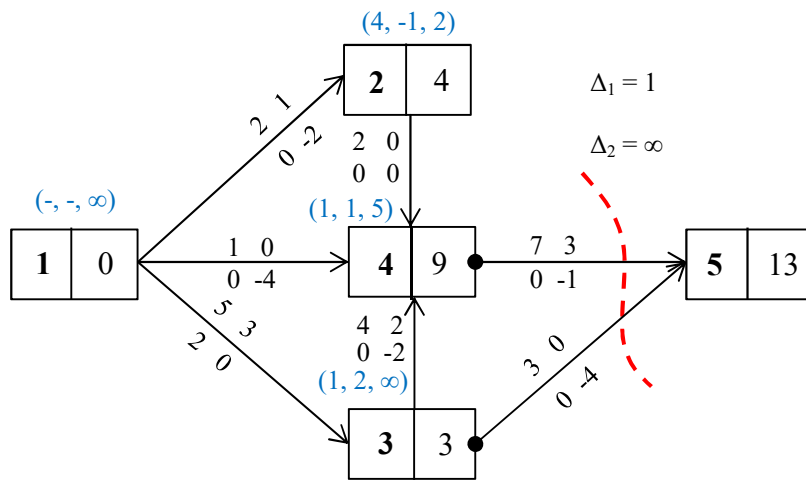
Şekil 3.5. Serim akış algoritması ile süre-gider eniyilemesi (devam)



(h)

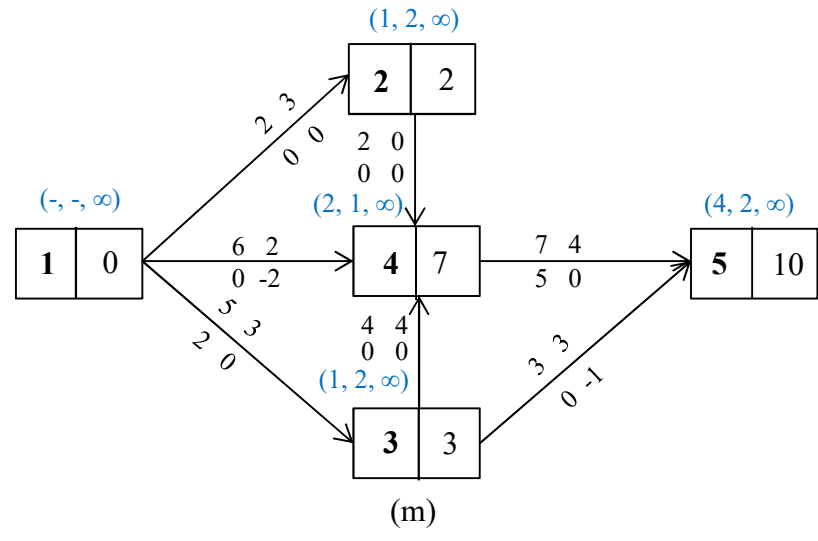
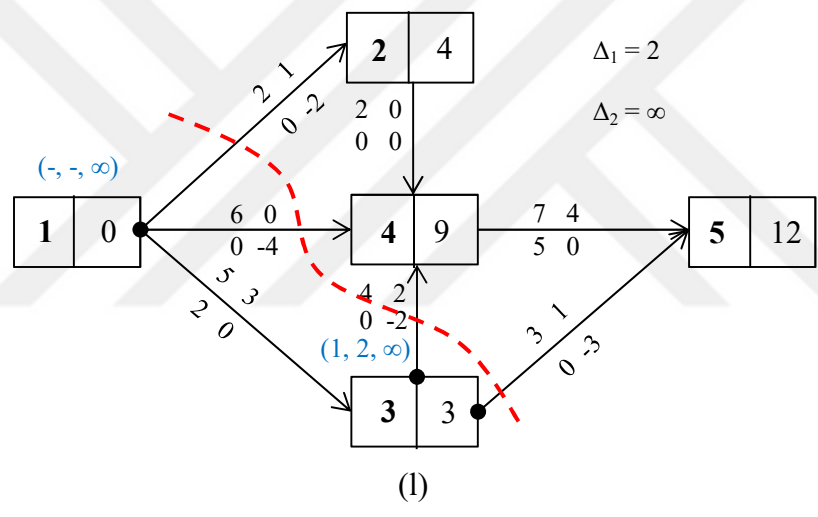
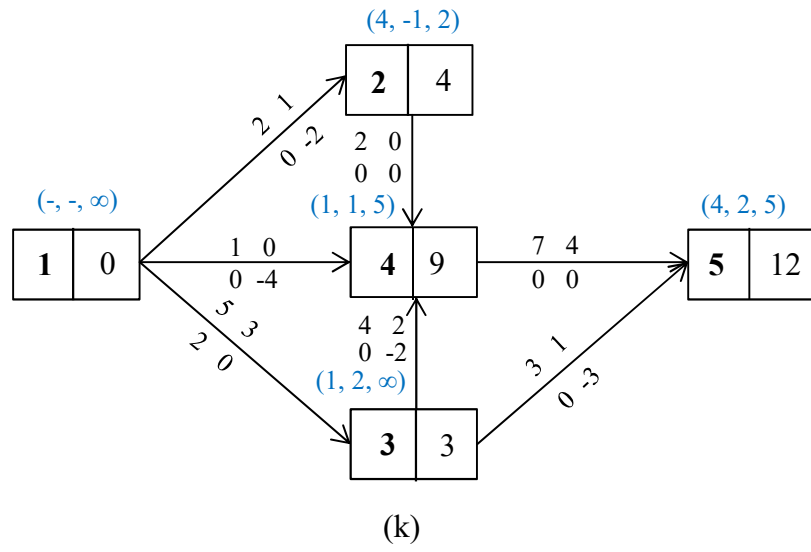


(i)



(j)

Şekil 3.5. Serim akış algoritması ile süre-gider eniyilemesi (devam)



Şekil 3.5. Serim akış algoritması ile süre-gider eniyilemesi (devam)

Şekil 3.5.'ten elde edilen değerler Çizelge 3.1. ile toplu halde ifade edilmektedir. Buna göre başlangıçta 21 gün olan proje süresi 10 güne kadar kısaltılabilmektedir. Her bir işleme ait zorlama süreleri ve bedelleri Çizelge 3.1.'den görülmektedir.

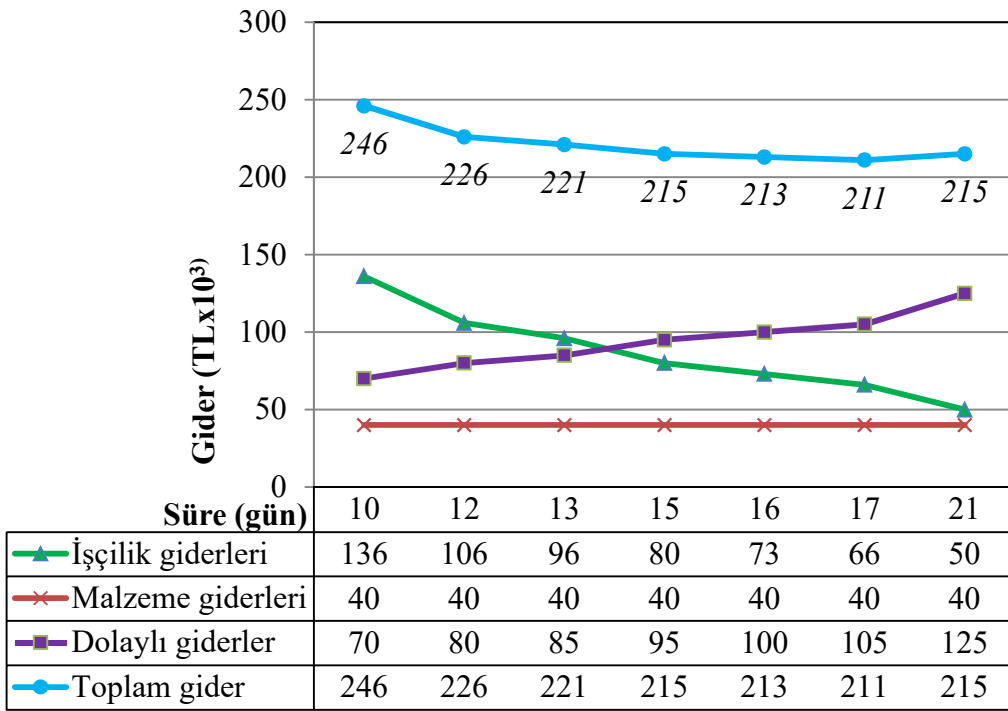
Çizelge 3.1. Serim akış yöntemiyle süre-gider eniyilemesinde zorlama giderleri

| Proje Süresi (gün) | İşlemler | | | | | | | Zorlama Gideri (TLx10 ³) | | |
|--------------------|----------|-----|-----|-----|-----|-----|-----|--------------------------------------|--------|----|
| | 1-2 | 1-3 | 1-4 | 2-4 | 3-4 | 3-5 | 4-5 | Hız | Toplam | Σ |
| 21 | 2 | 5 | 6 | ∞ | 4 | 3 | 7 | 0 | 0 | 0 |
| 17 | 5 | 6 | 9 | 5 | 4 | 10 | 7 | 4 | 16 | 16 |
| 16 | 4 | 5 | 9 | 5 | 4 | 10 | 7 | 7 | 7 | 23 |
| 15 | 4 | 5 | 9 | 5 | 4 | 10 | 6 | 7 | 7 | 30 |
| 13 | 4 | 3 | 9 | 5 | 6 | 10 | 4 | 8 | 16 | 46 |
| 12 | 4 | 3 | 9 | 5 | 6 | 9 | 3 | 10 | 10 | 56 |
| 10 | 2 | 3 | 7 | 5 | 4 | 7 | 3 | 15 | 30 | 86 |

3.1.'den elde edilen zorlama giderleri de kullanılarak Çizelge 3.2. oluşturulmuştur. Böylece her bir proje süresine karşılık gelen doğrudan ve dolaylı giderler ile toplam gider sunulmaktadır.

Çizelge 3.2. Serim akış yöntemiyle süre-gider eniyilemesi sonuçları

| Proje Süresi (gün) | Üretim Giderleri | | | | | Toplam Gider (TLx10 ³) |
|--------------------|-------------------|---------|----------------|------------------|-------|------------------------------------|
| | İşçilik giderleri | | Malzeme gideri | Dolaylı giderler | | |
| | Normal | Zorlama | | | Sabit | Değişken |
| 21 | 50 | 0 | 40 | 20 | 105 | 215 |
| 17 | 50 | 16 | 40 | 20 | 85 | 211 |
| 16 | 50 | 23 | 40 | 20 | 80 | 213 |
| 15 | 50 | 30 | 40 | 20 | 75 | 215 |
| 13 | 50 | 46 | 40 | 20 | 65 | 221 |
| 12 | 50 | 56 | 40 | 20 | 60 | 226 |
| 10 | 50 | 86 | 40 | 20 | 50 | 246 |



Şekil 3.6. Serim akış algoritmasıyla süre-gider eniyileme sonuç grafiği

Şekil 3.6. serim akış algoritması ile elde edilen sonuçları özetlemektedir. Buna göre proje 246000 TL toplam gider ile 10 günde tamamlanabilmektedir. Bu süre mümkün olan en az proje süresidir. Toplam giderin en küçük olduğu durum ise 211000TL ile 17 gün tamamlanma süresinin olduğu noktaya karşılık gelmektedir.

4. METASEZGİSEL ALGORİTMALAR

Özellikle son yıllarda hemen her alanda kullanılmaya başlanan eniyileme, mevcut sınırlayıcı şartları göz önüne alarak, çeşitli değişken parametrelerini bir amaç doğrultusunda en uygun değerlere getirme tekniklerinin bütünüdür. Günümüzde eniyileme ekonomide, üretimde, satış-dağıtımda ve mühendislik dallarının pek çoğunda sıklıkla kullanılmaktadır. Örneğin üretilen bir ürünün optimum boyutlarının bulunması, çalışma ve vardiya saatlerinin düzenlenmesi, araç rotalarının belirlenmesi ya da bir yapıda yükleri karşılayacak en uygun kesitlerin seçilmesi gibi konular eniyileme problemlerinin alanına girmektedir.

Eniyileme problemlerinin çözümünde geleneksel yöntemler giderek karmaşıklaşan problemleri çözüme yetersiz kalmaya başladığında, gerçek hayat problemlerine uygulanabilirliği yönünden daha başarılı olan sezgisel yöntemler 1970'lerden itibaren geliştirilmeye başlanmıştır. Bu yıllardan sonra araştırmacıların ilgisi doğada bulunan ve yiyecek bulma, eş seçme ya da göç etme gibi amaçlarını olabilecek en ideal şekilde elde edebilen sistemlere yönelmiştir. Son 20 yılda teknolojik gelişmelerin ve yapay zekanın da ışığında metasezgisel yöntemler doğmuştur.

Metasezgisel yöntemlerin pek çoğu doğadan esinlenen algoritmalarından oluşmaktadır. Doğadan esinlenen algoritmaların önemli bir kısmını da sürü zekasına dayalı algoritmalar oluşturmaktadır. Canlıların birer birey olarak elde etmekte güçlük yaşadıkları pek çok durum karşısında sürü halinde hareket ettiklerinde bireysel olarak kendilerinden beklenmeyecek düzeyde stratejiler geliştirdikleri ve başarılı sonuçlar ortaya koydukları araştırmacılar tarafında gözlenmiştir. Bunun sonucunda kuşlar, karıncalar, arılar, balıklar ve kurtlar gibi hayvanların avlanma, göç etme, barınak ve yiyecek bulma gibi konularda topluluk olarak hareket etmeleri durumunda olabilecek en iyi sonuçları alabilmeleri her birinin kendine ait özelliğinden yola çıkılarak geliştirilmiş bir algoritmaya ilham olmuştur. Doğadan esinlenen metasezgisel algoritmalar sürü zekasına dayalı olanlar haricinde Çizelge 4.1.'de, sürü zekasına dayalı metasezgisel algoritmalar Çizelge 4.2.'de, diğer metasezgisel algoritmalar da Çizelge 4.3.'te görülmektedir.

Çizelge 4.1. Doğadan esinlenen metasezgisel algoritmalar (Fister vd., 2013)

| Yıl | Yazar(lar) | Yöntem Adı |
|------|---------------------|--|
| 1997 | Storn ve Price | Diferansiyel gelişim algoritması |
| 2001 | Abbass | Balarısı evliliği |
| 2001 | Ferreira | Gen ekspresyonu |
| 2003 | Eusuff ve Lansey | Karışık kurbağa sıçrama algoritması |
| 2003 | Jung | Kraliçe arı evrimi |
| 2006 | Mehrabian ve Lucas | İstilacı ot optimizasyonu |
| 2008 | Lima vd. | Balık sürülerinde arama |
| 2008 | Simon | Biyocoğrafya tabanlı optimizasyon |
| 2008 | Havens vd. | Hamamböceği istilası algoritması |
| 2009 | Premaratne vd. | Çeltik tarlası algoritması |
| 2009 | He vd. | Grup arama algoritması |
| 2009 | Zhang vd. | İnsandan esinlenen algoritma |
| 2010 | Hedayatzadeh vd. | Termit koloni optimizasyonu |
| 2011 | Shi | Beyin fırtınası (parlak fikir) optimizasyonu |
| 2011 | Parpinelli ve Lopes | Doğadan esinlenen gelişimsel algoritma |
| 2012 | Yang | Çiçek tozlaşma algoritması |
| 2012 | Hernandez ve Blum | Japon ağaç kurbağası araması |
| 2012 | Maia | Optbees |
| 2012 | Mozaffari | Somon göçü algoritması |
| 2013 | Yan ve Hao | Atmosfer bulutları modeli |
| 2013 | Sur vd. | Mısır akbabası optimizasyonu |
| 2013 | Kaveh ve Farhoudi | Yunus ekolojasyonu |

Çizelge 4.2. Sürü zekasına dayalı metasezgisel algoritmalar (Fister vd., 2013)

| Yıl | Yazar(lar) | Yöntem Adı |
|------|-------------------------|---|
| 1992 | Dorigo | Karınca kolonisi optimizasyonu |
| 1995 | Kennedy ve Eberhart | Parçacık sürü optimizasyonu |
| 2001 | Lucic ve Teodorovic | Arı sistemi |
| 2002 | Passino | Bakteri yiyecek arama optimizasyonu |
| 2002 | Li vd. | Balık sürüsü algoritması |
| 2004 | Wedde vd. | Arı kovanı algoritması |
| 2005 | Teodorovic ve Dell'Orco | Arı kolonisi optimizasyonu |
| 2005 | Drias | Arı sürü optimizasyonu |
| 2005 | Yang | Sanal arılar |
| 2005 | Krishnanand ve Ghose | Parlak solucan sürü optimizasyonu |
| 2006 | Pham vd. | Arılar algoritması |
| 2006 | Chu vd. | Kedi sürü optimizasyonu |
| 2006 | Yang | Sanal karınca algoritması |
| 2007 | Chen vd. | Genetik algoritma ile bakteri yiyecek arama optimizasyonu |
| 2007 | Su vd. | Kafes sürü optimizasyonu |
| 2007 | Mucherino ve Seref | Maymun arama optimizasyonu |
| 2007 | Karaboğa ve Baştürk | Yapay arı kolonisi |
| 2008 | Yang | Hızlandırılmış parçacık sürü optimizasyonu |

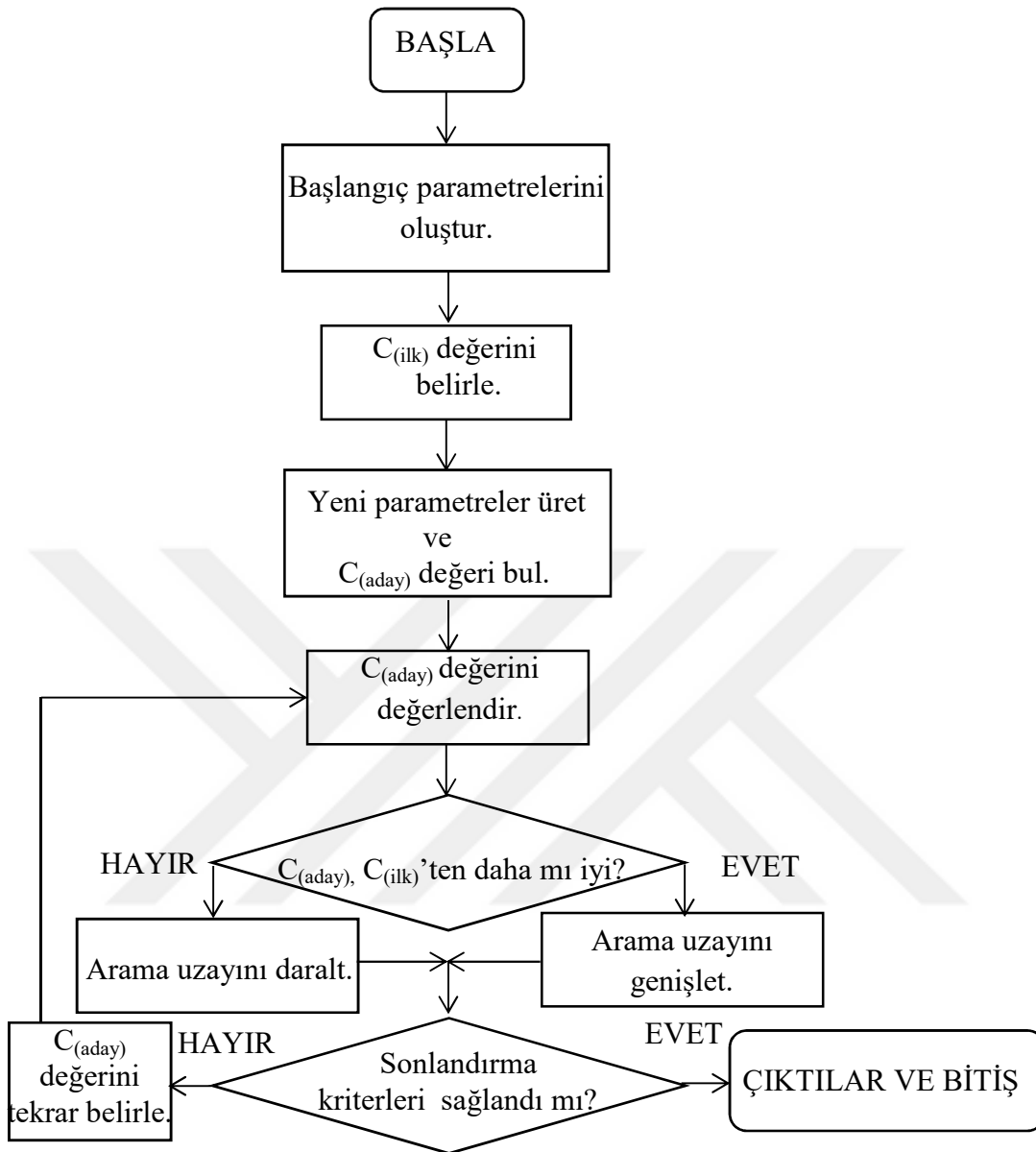
Çizelge 4.2. Sürü zekasına dayalı metasezgisel algoritmalar (Fister vd., 2013) (devam)

| Yıl | Yazar(lar) | Yöntem Adı |
|------|----------------------|--|
| 2008 | Chu vd. | Hızlı bakteriyel kaynaşma algoritması |
| 2009 | Yang ve Deb | Guguk kuşu arama algoritması |
| 2010 | Yang | Ateş böceği algoritması |
| 2010 | Iordache | Danışman rehberliğinde arama algoritması |
| 2010 | Chen vd. | Hiyerarşik sürü modeli |
| 2010 | Yang ve Deb | Kartal stratejisi |
| 2010 | Yang | Yarasa algoritması |
| 2011 | Comellas ve Martinez | Bombus arıları algoritması |
| 2012 | Ting vd. | Ağırlıksız sürü algoritması |
| 2012 | Gandomi ve Alavi | Kril sürüsü algoritması |
| 2012 | Tang vd. | Kurt arama algoritması |

Çizelge 4.3. Diğer bazı metasezgisel algoritmalar (Fister vd., 2013)

| Yıl | Yazar(lar) | Yöntem Adı |
|------|---------------------------|---------------------------------|
| 1998 | Ryan vd. | Dilbilgisel evrim |
| 2007 | Atashpaz-Gargari ve Lucas | Emperyalist rekabetçi algoritma |
| 2009 | Kashan | Şampiyonlar ligi algoritması |
| 2010 | Xu vd. | Sosyal duygusal algoritma |
| 2012 | Shayeghi ve Dadashpour | Anarşist toplum optimizasyonu |
| 2012 | Çivicioğlu Beşdok | Diferansiyel arama algoritması |
| 2013 | Çivicioğlu Beşdok | Geri izleme algoritması |
| 2013 | Çivicioğlu Beşdok | Yapay işbirliği araması |

Genel olarak metasezgisel yöntemler, belirlenmiş bir başlangıç noktası ile algoritmaya başlamakta ve çözüm uzayı içerisinde her adımda amaç fonksiyonunu kontrol ederek çok sayıda iterasyonla ilerlemektedir. Ayrıca, algoritma kullanıcısı problemin özelliğine göre süre ya da iterasyon sayısını algoritmanın sonlandırıcı unsuru olarak belirleyebilmektedir. Metasezgisel yöntemlerin türlerine bağlı olarak algoritmaları oldukça büyük farklılıklar içerse de temel işleyiş Şekil 4.1.'de görülen akış şemasında verilmektedir.



Şekil 4.1. Metasezgisel algoritmaların akış şeması (Toklu, 2014)

Bu çalışma kapsamında süre-gider analizine elverişli olması bakımından geleneksel analitik yöntemlerden doğrusal programlama, metasezgisel yöntemlerden de sırası ile genetik algoritmalar ve parçacık sürü optimizasyonuna yer verilmektedir.

4.1. Genetik Algoritmalar

Genetik algoritmalar (GA), evrimsel algoritmalar sınıfına ait bir metasezgisel yöntem türüdür. GA'yı daha iyi açıklayabilmek adına öncelikle evrimsel algoritmalarından kısaca bahsetmek gereklidir. Evrimsel algoritmalar, Darwin'in evrim teorisinden yola

çıkılarak popülasyonlar üzerinden çözüme ulaşan algoritmalarıdır. En yaygın kullanılan türü GA olmasına karşın, diferansiyel gelişim algoritması, evrimsel programlama ve evrimsel stratejiler gibi diğer önemli yöntem çeşitleri de bulunmaktadır. Bütün evrimsel algoritmalarda problemlerin karar değişkenleri genler yoluyla temsil edilmektedir ve çözüm prosedürüne başlayabilmek için karar değişkenlerine bir başlangıç çözümü oluşturacak popülasyonun belirlenmesi gerekmektedir. Karar değişkenlerinin uygunluğunu değerlendirmede kullanılan bir fonksiyon ve optimum çözüme ulaşabilmek için her bir iterasyonda çeşitliliği sağlayacak işlem operatörlerinin bulunması da evrimsel algoritmalar için bir zorunluluktur.

Evrimsel algoritmalar popülasyon tabanlı yöntemler olduğundan ortak bir özellik olarak tümünde $P(i)$ ile gösterilen çözüm uzayı mevcuttur. Bu çözüm uzayı her biri muhtemel çözümleri gösteren bireylerden oluşmakta ve bazı operatörler aracılığı ile dönüştürülerek $P(i+1)$ olarak ifade edilen yeni popülasyona evrilmektedir. Bu aşamada teoride de ortaya konulduğu gibi zayıf bireyler (optimumdan uzak olanlar) yok olmakta, yerine dayanıklı bireyler (optimum olma ihtimali yüksek olanlar) yaşamayı sürdürmektedir. Bireylerin optimum çözüm olup olmama durumları uygunluk değeri adı verilen bir hesaplama ile belirlenmektedir. Algoritma işleyişinde, uygunluk değeri yüksek bireyler popülasyonda yerini korurken, düşük değere sahip bireyler elenmektedir. Evrimsel algoritmaların genel prensiplerinden oluşturulan sözde kod (pseudo-code) aşağıda verilmektedir.

| |
|---|
| <p>$P(i)$'yi başlama durumuna getir</p> <p>$P(i)$'deki bireyleri değerlendir</p> <p>Sonlandırma ölçütü sağlanıyorsa algoritmayı bitir</p> <p>Sağlanmadıysa sonlandırma ölçütü sağlanana kadar aşağıdakileri tekrarla</p> <p style="padding-left: 40px;">Bireylere operatörleri uygula</p> <p style="padding-left: 40px;">Yeni popülasyon $P(i+1)$'i oluştur</p> <p style="padding-left: 40px;">$P(i+1)$'deki bireyleri değerlendir</p> <p>Bitir</p> |
|---|

Yukarıda görülen işlem adımlarında yapılan yer değişimleri yada uygulanan operatörlerdeki farklılıkların her biri farklı birer eniyileme yöntemi olarak karşımıza çıkmaktadır. Bu tez çalışması kapsamında GA üzerinde durulacaktır.

GA, ilk olarak 1975 yılında Amerikalı bilim adamı John Henry Holland tarafından “Adaptation in Natural and Artificial Systems” adlı eseri ile literatüre kazandırılmıştır (Holland, 1975). Holland, kitabında eniyileme problemlerindeki karmaşık unsurların GA sayesinde kodlanabileceğini öne sürmüştür.

GA, bir eniyileme yöntemi olarak canlılarda bulunan genetik yapı ve aşamaları taklit etmektedir. Bir başlangıç popülasyonu ile kendine çözüm uzayı oluşturan GA, sonraki her yeniden üremede, seçme ya da diğer bir ifade ile seçim operatöründen yararlanarak iterasyonlarına devam etmektedir. İterasyonların her biri nesil, kuşak ya da jenerasyon olarak adlandırılmaktadır. Algoritma sonlandığında en uygun birey probleme ait optimal sonucu oluşturmaktadır. Bu sonucun, bütün metasezgisel algoritmalarda da olduğu gibi her zaman optimum olmasa bile optimuma oldukça yakın olduğu bilinmektedir.

Genetik algoritmalar özellikle aşağıda sayılan durumlarda kullanışlı ve etkili olabilmektedir.

- Deterministik yöntemler ile çözümün zor olduğu ve uzun hesaplamalar gerektirdiği durumlar,
- Araştırma alanının geniş ve karmaşık olduğu durumlar,
- Problem ile ilgili yeterli verinin olmadığı durumlar ve
- Geleneksel yöntemlerden yararlanan programların çeşitli nedenlerle yetersiz kaldığı durumlarda genetik algoritmalar uygun yöntem olabilir.

Ayrıca problemin yukarıda bahsedilen elverişsiz koşulları olmasa da genetik algoritmalar bir takım faydalı yönlerinden ötürü tercih sebebi olabilmektedir. Örneğin, bir veri grubundan özel bir veriyi bulabilmek ya da uzun sürede kesin sonuç yerine kısa sürede yaklaşık fakat yeterli doğrulukta sonucu elde edebilmek gibi özellikleri genetik algoritmaları faydalı kılmaktadır.

4.1.1. Genetik Algoritma Operatörleri

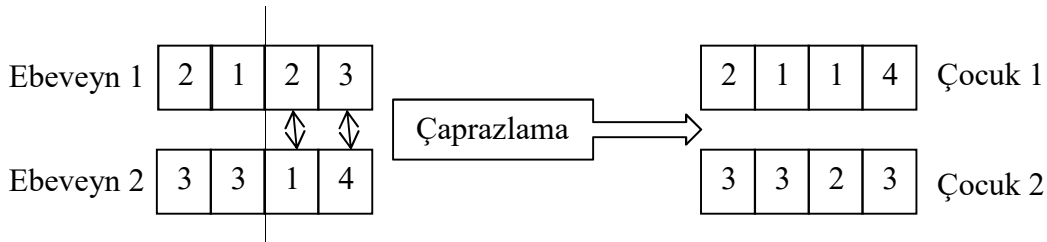
GA'da birey adı verilen ve her biri muhtemel çözümü temsil eden yapılar genetik bilimindeki kromozomların karşılığıdır. Benzer şekilde genetik biliminde kromozomları oluşturan genler ise GA'da karar değişkenleri olarak ifade edilen bitlerdir. Bu bit dizileri yani bireyler yaygın olarak ikili sistem kodlama ile temsil edilebileceği gibi onlu sistemde de kodlanabilmektedir.

GA'da yeni nesilleri oluşturmada kullanılan başlıca üç operatör bulunmaktadır. Bu operatörler, seçim, çaprazlama ve mutasyon işlemleridir. Seçme operatörünün uygulanabilmesi için bütün bireyler uygunluk değerlendirilmesinden geçmektedir. Böylece, uygunluk değeri yüksek bireyler bir sonraki nesli oluşturacak bireylerin ebeveynleri olmaktadır. Bahsedilen üç operatöre ilave olarak başlangıç popülasyonu seçim yöntemi, uygunluk fonksiyonu ve algoritmayı sonlandırma şartları da algoritmanın performansını belirleyen önemli parametrelerdir.

GA'da popülasyondaki birey çeşitliliğini artırarak yeni nesillerin daha uyumlu hale gelmesini sağlayan seçim, çaprazlama ve mutasyon operatörlerinin problem yapısına uygun kullanımı algoritmanın performansını doğrudan etkilemektedir. Bu operatörlerden ilk olarak algoritmaya giren seçim operatörü, uyum düzeyi yüksek bireylerin hayatta kalarak yeni nesil için ebeveynlik yapmasını sağlamaktadır. Seçim operatörü olarak deterministik seçim, rulet tekerleği tekniği, turnuva seçimi ya da rasgele seçim kullanılmaktadır.

Seçim operatörü ile bir sonraki nesil için ebeveyn havuzu oluşturulmasının ardından ikinci GA operatörü olan çaprazlama devreye girmektedir. Çaprazlama iki ebeveynin kromozomlarında bulunan gen özelliklerini kendinden sonraki nesle aktararak yeni bir birey meydana getirmesidir. Ebeveyn havuzundan rasgele seçilen iki birey tek noktadan ya da çok noktadan bölünerek birbirlerine gen dizilerinden parçacıklar aktarırlar. Böylelikle her iki ebeveynin de özelliklerini barındıran çocuk bireyler oluşur. Çaprazlama sonucu kullanıcının tercihine göre ebeveynler popülasyonda çocukları ile birlikte yer alabileceği gibi, yok edilmek suretiyle popülasyon büyüklüğü de korunabilir. Çaprazlama

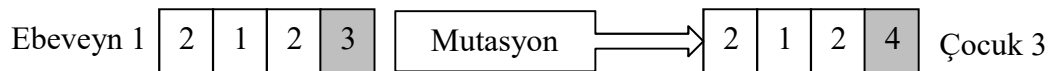
operatörünün uygulanmasına ilişkin basit bir gösterim 4 genli bir kromozom üzerinde Şekil 4.2.'de verilmektedir.



Şekil 4.2. Tek noktadan çaprazlama işlemi örneği

Çaprazlama operatörünün etkinliğini belirleyen unsur çaprazlama oranıdır. Bu oran bir sonraki nesil oluşturulurken ebeveynler arası gen aktarımının düzeyini kontrol etmektedir. Eğer kullanıcı düşük bir oran tercih ederse yeni popülasyon eski popülasyona oldukça benzer çıkacak algoritmanın yakınsaması olumsuz etkilenecektir. Aşırı yüksek çaprazlama oranları ise uygunluk değeri yüksek bireylerin, yeni nesillere bu özelliklerini aktarmadan hızlı bir şekilde kaybolmasına neden olmaktadır.

Sonucu GA operatörü olan mutasyon operatörünün işlevi ise genetik mutasyon olayının bir uyarlaması olarak kromozomda yer alan genlerden en az bir tanesinde değişimin meydana getirilmesidir. Böylece çözüm uzayında daha önce var olmayan yeni muhtemel çözümler de popülasyona katılabilmektedir. Şekil 4. .'de mutasyon işlemine ait bir örnek gösterim verilmektedir.



Şekil 4.3. Mutasyon işlemi örneği

Çaprazlama oranında olduğu gibi, mutasyonda da kullanıcının bilgi ve tecrübesi ile algoritmayı kontrol edeceği bir oran bulunmaktadır. Bu oran, çözüm uzayının araştırma açısından yeterli olup olmamasını belirleyen temel etkenlerdendir.

Yukarıda açıklanan standart GA operatörlerine ek olarak daha ileri düzeyde geliştirilmiş operatörler de bulunmaktadır. Gelişmiş operatörler sayesinde GA'nın

eniyileme kabiliyeti yükseltilmekte ve çeşitli problem türlerine göre uygulanabilirlik alanı genişletilmektedir.

GA operatörlerinin uygun seçimi ve kullanımının büyük önemi olduğu kadar algoritma performansını etkileyen diğer bir unsur da kontrol değişkenleridir. Popülasyon büyüklüğü ve maksimum nesil sayısı başlıca algoritma işleyişini kontrol eden değişkenlerdir. Popülasyon büyüklüğünün uygun seçilmesi algoritmanın çözüme ulaşmasında çok büyük rol oynamaktadır. Gereğinden büyük seçilen popülasyonlarda çözüme ulaşmak için çok uzun sürelere ihtiyaç duyulmaktadır. Bu nedenle kullanışlı ve etkin bir kontrol sağlanamamaktadır. Benzer şekilde popülasyonların fazlasıyla küçültülmesi de çözüm uzayını kısıtlayacağından muhtemel uygun çözümlerin popülasyon dışı kalmasına yol açmaktadır.

Literatür incelendiğinde GA'daki kontrol değişkenleri üzerine yapılan çalışmalar sonucu bazı araştırmacıların uygun değişken değerleri önerdikleri görülmektedir. Bu değerler problemlerin türüne göre uygun ayarlamalara ihtiyaç duysa da ön bilgi sağlaması bakımından yararlıdır. De Jong (1975), Grefenstette (1986) ve Schaffer (1989)'a dair kontrol değişken önerileri Çizelge 4.4.'te verilmektedir.

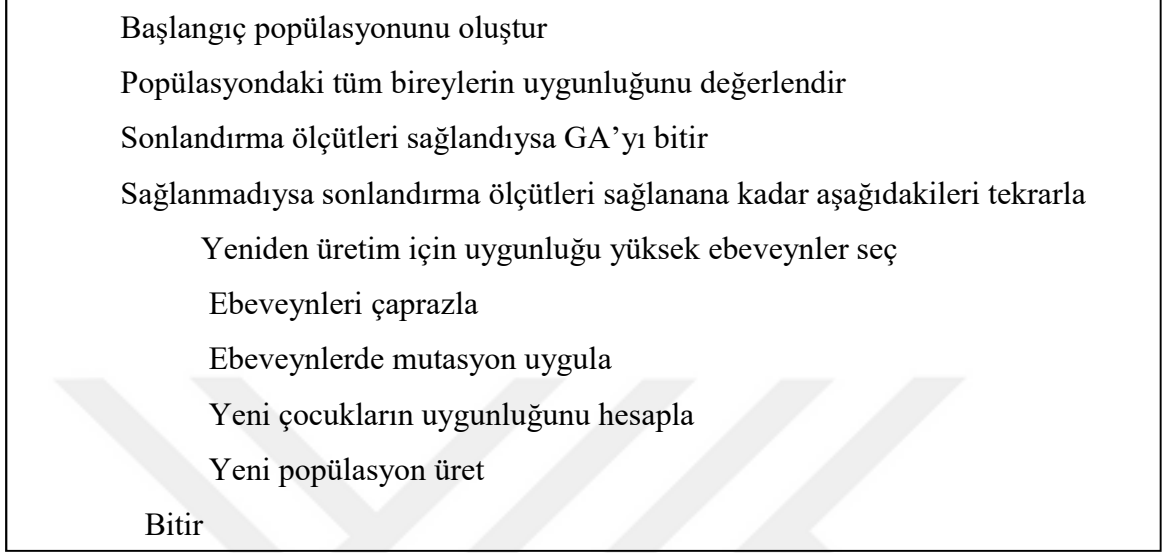
Çizelge 4.4. Araştırmacılar tarafından önerilmiş kontrol değişken değerleri (Karaboğa, 2014)

| Kontrol Değişkenleri | De Jong (1975) | Grefenstette (1986) | Schaffer (1989) |
|-----------------------------|-----------------------|----------------------------|------------------------|
| Popülasyon büyüklüğü | 50-100 | 30 | 20-30 |
| Çaprazlama oranı | 0,60 | 0,95 | 0,75-0,95 |
| Mutasyon oranı | 0,001 | 0,01 | 0,005-0,01 |

4.1.2. Genetik Algoritma Prosedürleri

GA, algoritma yapısı yönünden hem esnetilmeye hem de güçlendirilmeye oldukça uygun metasezgisellerdendir. Bununla beraber, GA'ya özgü standart işleyiş bu tip

algoritmaların çekirdeğinde yer alır. Bu bakımdan aşağıda verilen GA sözde kodu (pseudo-code) ve Şekil 4.4.'te görülen akış şeması genel işlem prosedürünü özetlemektedir:



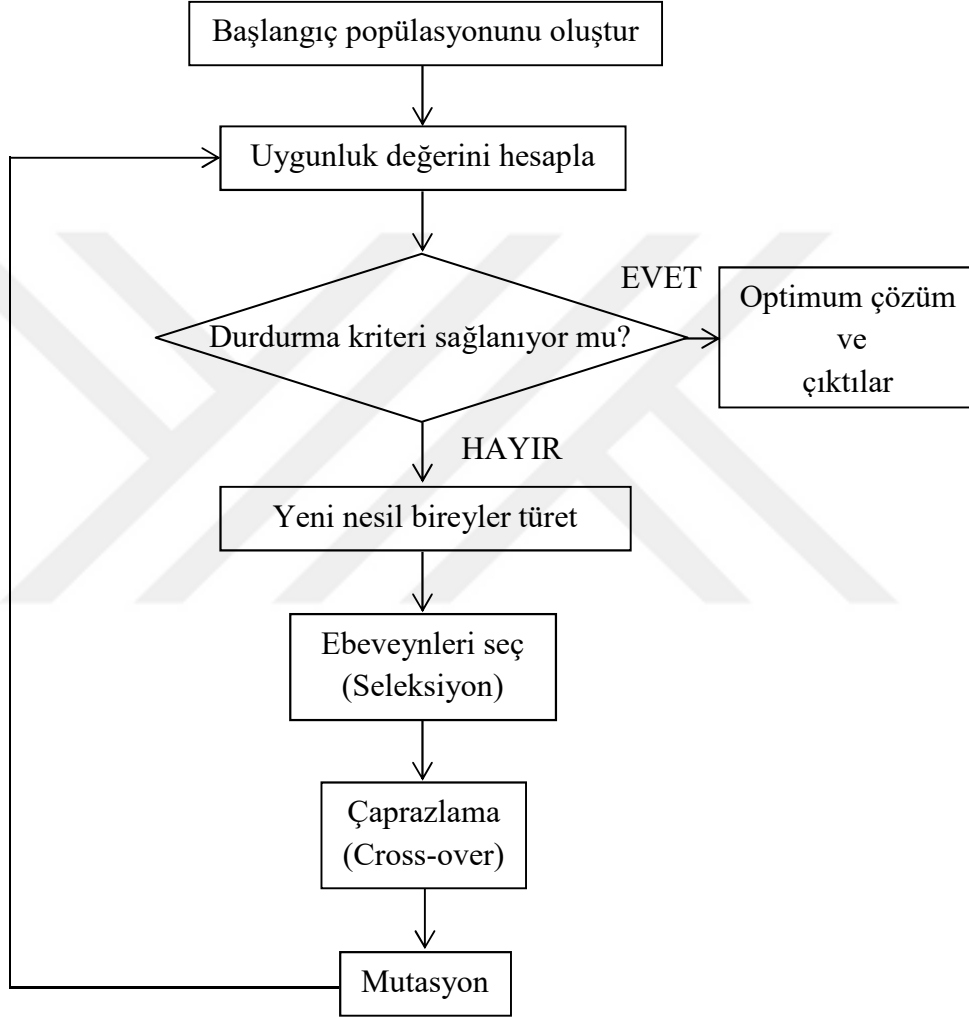
GA'da işleyiş başlangıç popülasyonunun oluşturulması ile başlamaktadır. Başlangıç popülasyonu çeşitli yollarla oluşturulabilmektedir. Kullanıcı problem sonucu hakkında öngörü sahibi ise zaman kaybını azaltmak bakımından tahmini değerler başlangıç popülasyonunu oluşturmada kullanılabilir. Ancak bunun söz konusu olmadığı problemlerde de rasgele sayı üreticisi kullanışlıdır.

GA işleyişinde ikinci aşama uygunluk değerlendirmesinin yapılmasıdır. Elde edilen çözüm adaylarının hangisinin optimuma daha yakın olduğunun belirlenmesinde uygunluk değerlendirmesinin başlıca payı vardır. Özellikle kapsamlı ve karmaşık eniyileme problemlerinde algoritmanın çözüm uzayı içerisindeki uygunluk değeri yüksek çözümlerin olduğu bölgeye yönelmesi yerel bir optimum noktasına takılmasına sebep olabilmektedir. Erken yakınsama adı verilen bu olumsuz durumda çözüm uzayının tamamen taranması mümkün olmayacaktır. Bu durumun ortaya çıkmasından kaçınmak ve küresel optimum değerini araştırmak için iraksama uygulanabilmektedir (Karcı, 2002).

GA'da uygunluk değeri hesaplanmasından sonraki aşama durdurma kriteri kontrolüdür. Durdurma kriterleri olarak adlandırılan algoritma kontrol ölçütleri ulaşılabilecek

nesil sayısı ya da belirlenmiş bir süre olabilmektedir. Seçilen kritere gelindiği anda algoritma sonlanmakta ve o ana kadar bulunan en iyi çözüm optimal kabul edilmektedir.

GA'ya ait işlem adımlarını görsel olarak ifade edebilmek adına Şekil 4.4.'te GA akış şeması yer almaktadır.



Şekil 4.4. GA akış şeması

4.2. Parçacık Sürü Optimizasyonu

Parçacık sürü optimizasyonu (PSO), sürü zekası tabanlı doğadan esinlenen bir algoritma türüdür. Bu tür algoritmaların temelinde sürü adı verilen canlı topluluklarının psikososyal unsurlara bağlı hareketleri bulunmaktadır. PSO'da her bir birey parçacık olarak isimlendirilmekte ve parçacıkların hem birbiriyle hem de çevre ile olan etkileşimleri sürü zekasını oluşturmaktadır. Özellikle arı, kuş, balık ve hatta bakterilerde dahi görülen bu durum, topluluk içerisinde bulunan parçacıkların gözlem ve kavrama kabiliyetlerinin bireysel durumlarına kıyasla daha iyi olduğuna işaret etmektedir. Yiyecek arama, göç etme ve tehlikeden kaçma gibi davranışlarda her bir parçacık kendi geçmiş deneyimleri ve içgüdüleri yoluyla sürüde bilinç ve öngörü oluşmasına katkı sağlamaktadır. Bu noktada bazı parçacıklar sürüye liderlik ederken kalan parçacıklar da sürünün uyumlu birer üyesi olarak öncü pozisyonundakileri takip etmektedirler.

PSO, 1983'te Reeves tarafından ilk parçacık-sürü fikrinin ortaya atılmasını takiben 1987'de Reynolds, 1990'da Heppner ve Grenander tarafından da adım adım geliştirilmesinin ardından, 1995 yılında Kennedy ve Eberhart tarafından her parçacığın muhtemel bir çözümü temsil ettiği yeni bir optimizasyon metodu olarak tanıtılmıştır. En genel ifade ile PSO, çözüm uzayında sürekli hareket halinde bulunan her bir parçacığın kendi deneyimleri ve komşu parçacıklarının deneyimleri sonucu pozisyonlarının değiştiği bir sistem olarak tanımlanabilir.

Engelbrecht (2006)'e göre s. zamanda çözüm uzayında yer alan i parçacığının pozisyonu $x_i(s)$ ve hızı $h_i(s)$ ile gösterilirse (s+1). zamanda i parçacığının konumu aşağıdaki gibi ifade edilmektedir.

$$x_i(s+1) = x_i(s) + h_i(s+1) \quad (4.1)$$

Yukarıdaki bağıntıda hız vektörü olarak gösterilen bileşen PSO'ya yön veren ana unsurdur. Hız vektörünün içerisinde parçacığın hem deneyimsel hem de komşu parçacıklar yönünden elde ettiği bilgi bulunmaktadır. Kişisel deneyimsel bilgi parçacığın bilişsel bileşeni olarak tanımlanırken komşulardan elde edilen bileşen de sosyal bileşen adını almaktadır. Her parçacığın bir çözümü temsil ettiği PSO'da parçacıkların her birinin

konumu her iterasyonda güncellenmektedir. Bu güncelleme, her parçacığın o ana kadarki kendi en iyi konumu l_{best} (yerel en iyi) ile sürünün en iyisi g_{best} 'e (küresel en iyi) göre yapılmaktadır. Bu nedenle hem bilişsel hem de sosyal yönden bileşenlere sahip olan hız vektörünün doğru tanımlanması oldukça önemlidir. s zamanında sürü içinde bulunan en iyi konum $g_{best}(s)$, bireysel en iyi konum $l_{best}(s)$, bilişsel ve sosyal bileşenleri temsil eden hızlandırma katsayıları c_1 ve c_2 , stokastik yaklaşım sağlayan rasgele sayılar $r_{1j}(s)$ ve $r_{2j}(s)$ atalet katsayısı p ile gösterilecek olursa, $j=1, \dots, n$ boyutunda i parçacığının hızı $h_{ij}(s+1)$ 4.2. bağıntısı yoluyla hesaplanabilir.

$$h_{ij}(s+1) = ph_{ij}(s) + c_1 r_{1j}(s)[l_{best}(s) - x_{ij}(s)] + c_2 r_{2j}(s)[g_{best}(s) - x_{ij}(s)] \quad (4.2)$$

Bağıntı (4.2)'de yer alan c_1 , c_2 katsayıları 0-2 aralığında bir değer alırken r_1 , r_2 ise 0-1 aralığındadır. Atalet katsayısı p ise orijinal PSO algoritmasında 1 iken modifiye PSO'larda 0,8-1,2 aralığında değişmektedir. Bu katsayı algoritmanın yerelde ya da küreselde çözüm arayışının kabiliyetini etkilemektedir. Değer düşürüldükçe yerle, yükseltildikçe küresele aramaya yatkınlık artmaktadır. PSO'da kullanılan tüm katsayı ve parametrelerin problemin yapısına uygun şekilde seçilmesi optimuma ulaşmayı kolaylaştırması bakımından oldukça önemlidir.

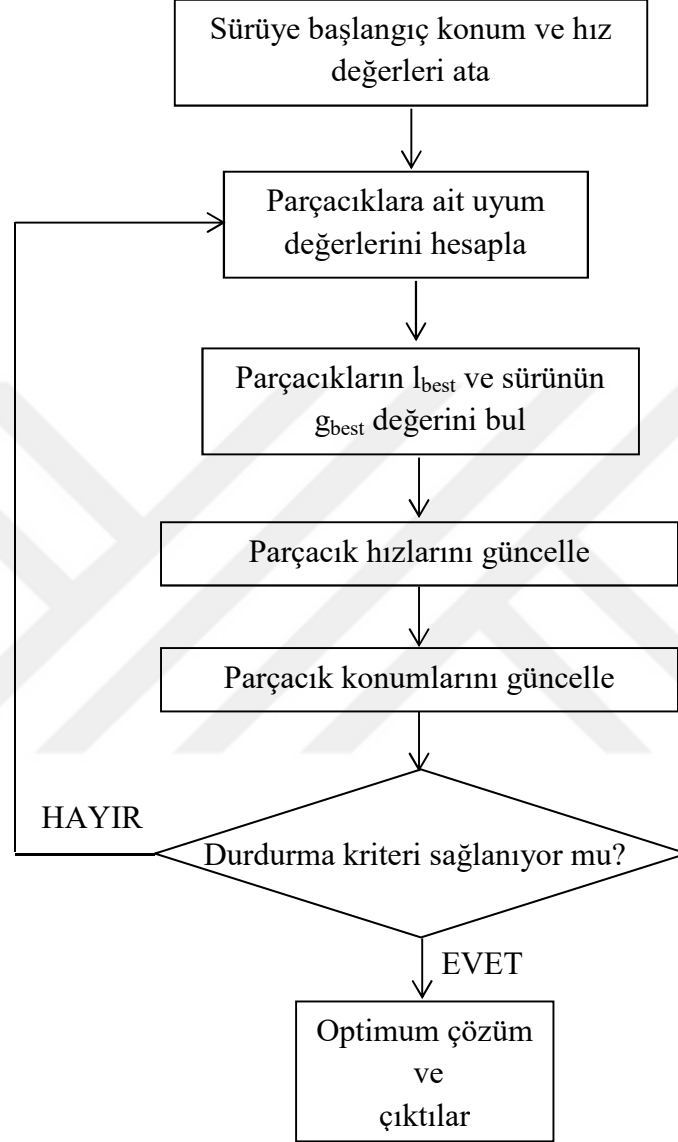
Amaç fonksiyonu minimize etmeye yönelik olan optimizasyon modellerinde $s+1$ zamanında bireysel en iyi konum (y_i) bağıntısı (4.3)'te görülmektedir.

$$y_i(s+1) = \begin{cases} y_i(s) & , \quad f(x_i(s+1)) \geq f(y_i(s)) \\ x_i(s+1) & , \quad f(x_i(s+1)) < f(y_i(s)) \end{cases} \quad (4.3)$$

(4.3) nolu ifadede verilen f fonksiyonları metasezgisel algoritmaların pek çoğunda bulunan uyum fonksiyonunu göstermektedir. Bu fonksiyon sürü içerisindeki bir parçacığın konumunun optimuma yakın ya da uzak olduğuna işaret etmektedir. Bireysel en iyi konumdan küresel en iyi konuma (y_g) geçerken bağıntı (4.4) kullanılmaktadır.

$$y_g(s) = \min \{f(x_1(s)), \dots, f(x_n(s))\} \quad (4.4)$$

PSO'nun genel işleyişine bağlı olarak oluşturulan akış şeması Şekil 4.5.'te verilmektedir.



Şekil 4.5. PSO akış şeması

5. ANALİTİK VE METASEZGİSEL YÖNTEMLER KULLANARAK SÜRE-GİDER ENİYİLEME UYGULAMASI

Birbirlerine göre bağımlılık ve ardışıklık ilişkisi belirlenmiş işlemler dizisine proje adı verilmektedir. İnşaat mühendisliği açısından ise gerekli tasarımları yapılmış bütün inşaat ve yapım işleri birer inşaat projesidir. Genel olarak bir projenin uygulanmasından sorumlu tarafın projenin tamamlanma süresi hakkında bir taahhütte bulunduğu, buna karşılık da taahhütün yerine getirilmediği durumlar için de kararlaştırılan bir ceza bedelini kabul ettiği bilinmektedir. Dolayısıyla, bir proje teklifi hazırlanırken her bir inşaat faaliyetinin normal şartlarda ne kadar zamanda ve ne kadar maliyet ile tamamlanacağı belirlenmektedir. Ancak kimi zaman çeşitli sebeplerle yüklenicinin taahhüt ettiği proje bitiş süresi aşılmaktadır. Böyle durumlarda, işlemlerin normalden daha kısa zamanda tamamlanması için fazla mesai, ilave işçi, makine ya da teknoloji temini gibi farklı yollardan ulaşılabilecek en kısa süreler araştırılmalıdır. Bu noktada önemli bir husus da bu yeni durumun getireceği ek giderler ile toplam proje süresinin kısaltmasının dolaylı giderler üzerindeki etkilerinin karşılaştırılmasıdır.

Özetle, normal işlem süreleri çerçevesinde en kısa proje tamamlanma süresinin bulunması ve projenin taahhüt edilen sürede bitirilmesi için gerekiyorsa en küçük ek giderli alternatiflerin belirlenmesi süre-gider eniyilemesinin inşaat mühendisliği alanındaki uygulamasıdır. Çalışmanın bu bölümünde, süre-gider eniyilemesine ilişkin çeşitli problemler seçilen ve önerilen bazı yöntemler ile irdelenmektedir. Bu yöntemler sırasıyla analitik yöntem, genetik algoritmalar ve parçacık sürü optimizasyonudur.

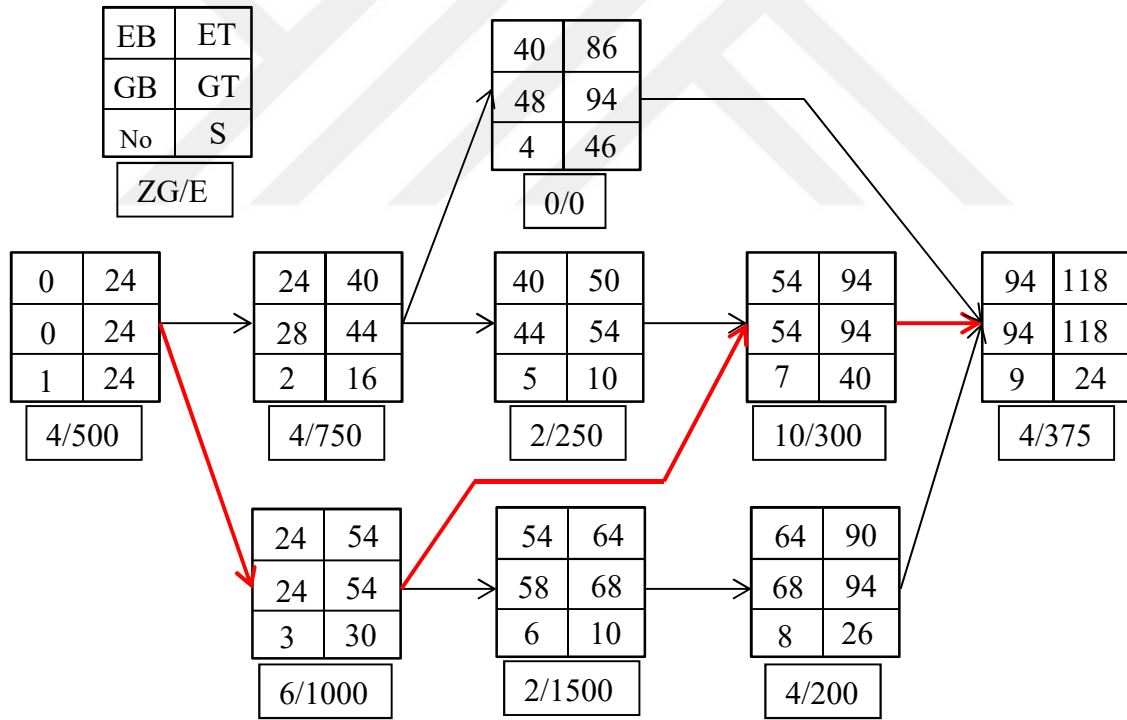
5.1. Analitik Yöntemle Süre-Gider Eniyileme Uygulaması

İşlem sayısı 9 olan küçük bir inşaat projesine ait veriler Çizelge 5.1.'de verilmektedir. Çizelgede süreler ve doğrudan gider değerleri her bir işlem için hem normal hem de zorlanmış durum olarak görülmektedir. Dolaylı giderler 300 TL/gün olarak varsayılmıştır.

Çizelge 5.1. Analitik uygulamaya ilişkin işlem bilgileri

| İşlem | Öncül işlem | Normal | | Zorlanmış | |
|-------|-------------|------------|------------|------------|------------|
| | | Süre (gün) | Gider (TL) | Süre (gün) | Gider (TL) |
| 1 | - | 24 | 14000 | 20 | 16000 |
| 2 | 1 | 16 | 10000 | 12 | 13000 |
| 3 | 1 | 30 | 8000 | 24 | 14000 |
| 4 | 2 | 46 | 10000 | 46 | 10000 |
| 5 | 2 | 10 | 2000 | 8 | 2500 |
| 6 | 3 | 10 | 6000 | 8 | 9000 |
| 7 | 3,5 | 40 | 12000 | 30 | 15000 |
| 8 | 6 | 26 | 5000 | 22 | 5800 |
| 9 | 4,7,8 | 24 | 6000 | 20 | 7500 |

Çizelge 5.1.'de verilen öncüllük ilişkilerine dayanarak işlemler düğüm noktaları ile ifade edilmek suretiyle (activity-on-node) Şekil 5.1.'deki proje serimi oluşturulmuştur.



Şekil 5.1. Proje serimi, başlangıç çözümü ve kritik yol gösterimi

Serimde yer alan düğüm noktalarında işlemlerin bilgileri bulunmaktadır. Buna göre bir düğümde “EB” erken başlama zamanını, “ET” erken tamamlanma zamanını, “GB” geç başlama zamanını, “GT” geç tamamlanma zamanını, “No” işlem adını, “S” ise işleme ait

süreyi göstermektedir. Düğüm noktasının altında yer alan kutucuklarda da işleme ait zorlanabilecek gün sayısı (ZG) ve süre-gider fonksiyonunun eğimi (E) verilmektedir.

Süre-gider ilişkisinin doğrusal olduğu durumlarda eğim aşağıdaki şekilde hesaplanmaktadır (5.1). Eğim, bir işlemin süresini 1 birim süre kısaltmak karşılığında ortaya çıkan gideri vermektedir. Eğim ne kadar dik yani değeri yüksek ise işlemi hızlandırmak o derecede maliyetli olmaktadır.

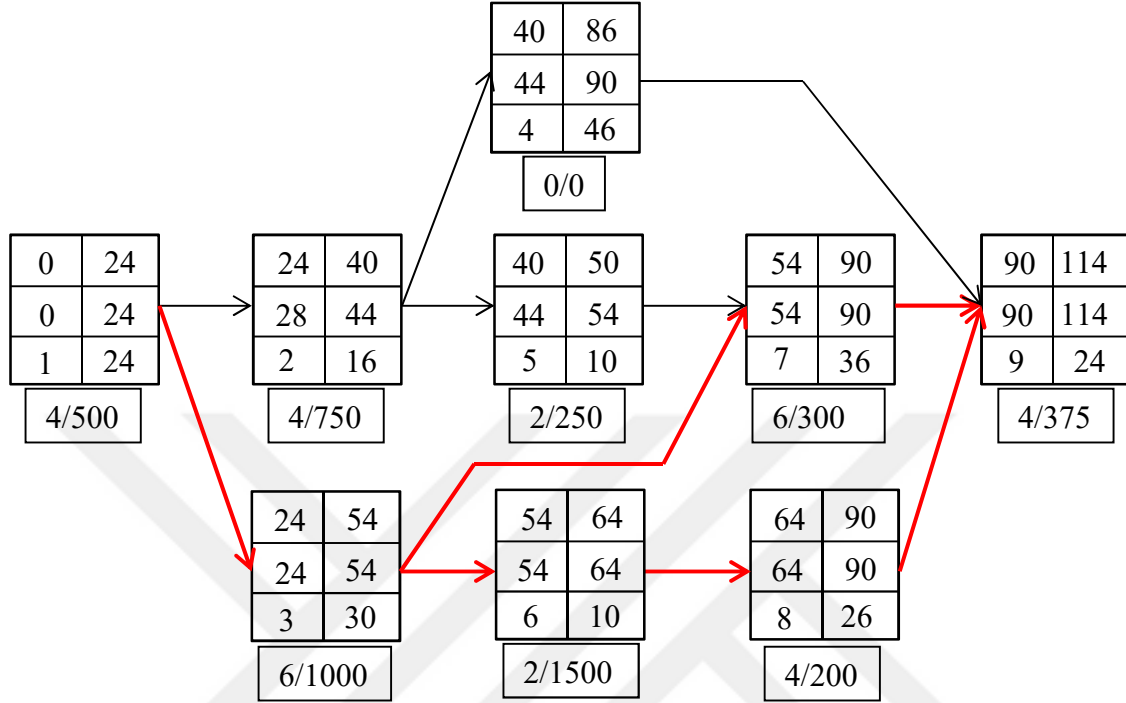
$$\text{Eğim} = (\text{Zorlanmış Gider} - \text{Normal Gider}) / (\text{Normal süre} - \text{Zorlanmış süre}) \quad (5.1)$$

Şekil 5.1.'de görüldüğü üzere problem üzerinde öncelikle başlangıç çözümü oluşturulmuştur. Bunun için normal sürelerden yararlanılarak “Kritik Yol Yöntemi-CPM” ile erken başlama-tamamlanma, geç başlama-tamamlanma zamanları bulunmuştur. Proje normal şartlar altında 118 günde tamamlanmaktadır. Gecikmeleri durumunda proje süresini doğrudan etkileyecek, diğer bir ifade ile bolluk süreleri olmayan işlemlerin oluşturduğu yörünge yani “Kritik Yol” ise “1-3-7-9” şeklinde bulunmuş, Şekil 5.1.'de kırmızı ile işaretlenmiştir.

(5.1) bağıntısı kullanılarak serim üzerindeki işlemlerin eğimleri hesaplanmış ve Şekil 5.1.'de yazılmıştır. 4 nolu işlemin hızlandırmaya uygun bir işlem olmadığı Çizelge 5.1.'den anlaşılmıştır. Dolayısıyla 4 nolu düğüm için zorlanabilecek gün (ZB) ve eğim (E) kutucuklarına sıfır değeri verilmiştir. Ayrıca normal sürelerle bağlı çözüm gerçekleştirildiğinden bu çözüme karşılık gelen toplam proje gider değeri Çizelge 5.1.'deki normal giderlerin toplanması yoluyla bulunmuştur. Buna göre hiçbir hızlandırma yapılmaması halinde söz konusu proje 118 günde 73000 TL doğrudan gider ile tamamlanmaktadır.

Kritik yol üzerinde yapılacak hızlandırmalar toplam proje süresini doğrudan etkileyeceği için “1-3-7-9” işlemleri göz önüne alınarak, eğimi en küçük olan 7 işlemi seçilmektedir. Bunun sebebi proje süresi kısaltılırken bir yandan da bunun mümkün olan en düşük ek giderle yapılmak istenmesidir. Zorlanabilecek 10 günlük imkanı olmasına rağmen 4 günden fazla zorlamalarda kritik yol değişeceğinden 7 nolu işlem maksimum 4

gün hızlandırılacaktır. Gerekli düzenlemeler yapıldıktan sonraki durum Şekil 5.2.'de verilmektedir.



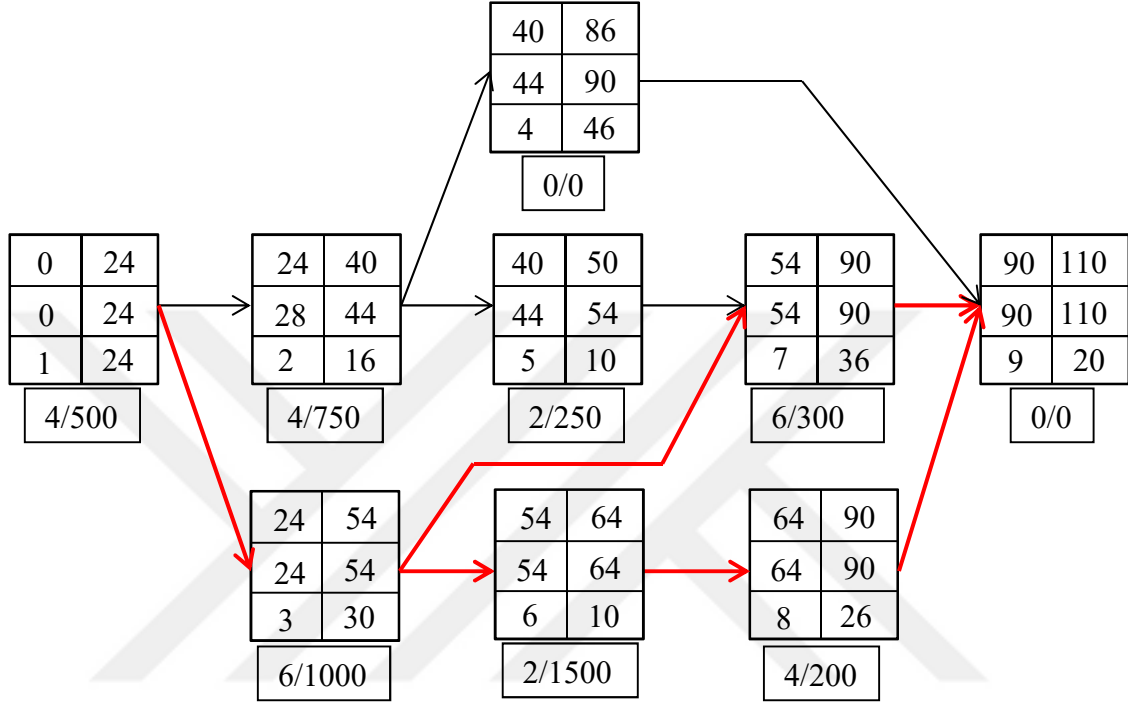
Şekil 5.2. İlk hızlandırmadan sonraki durum ve kritik yolların gösterimi

Yeni durumda proje 114 günde ve gün başına ilave 300 TL gider ile tamamlanmaktadır. Yeni gider toplam olarak 74200 TL'dir. Bu yapılan hızlandırma işlemi sonrasında yeni bir kritik yol ortaya çıkmıştır (1-3-6-8-9). Şekil 5.2.'de de kırmızı ile gösterildiği üzere 6 ve 8 işlemleri de böylelikle kritik hale gelmiştir.

İkinci hızlandırma işlemini yapmak için aşağıda verilen alternatifler mevcuttur. Son 2 alternatif tek başına kritik yolu değiştirecekleri için ancak birlikte hızlandırılabilir işlemlerden oluşmaktadır.

- 500 TL/gün gider ile 1 nolu işlem,
- 1000 TL/gün gider ile 3 nolu işlem,
- 375 TL/gün gider ile 9 nolu işlem,
- 1800 TL/gün gider ile 6 ile 7 nolu işlemler,
- 500 TL/gün gider ile 7 ile 8 nolu işlemler.

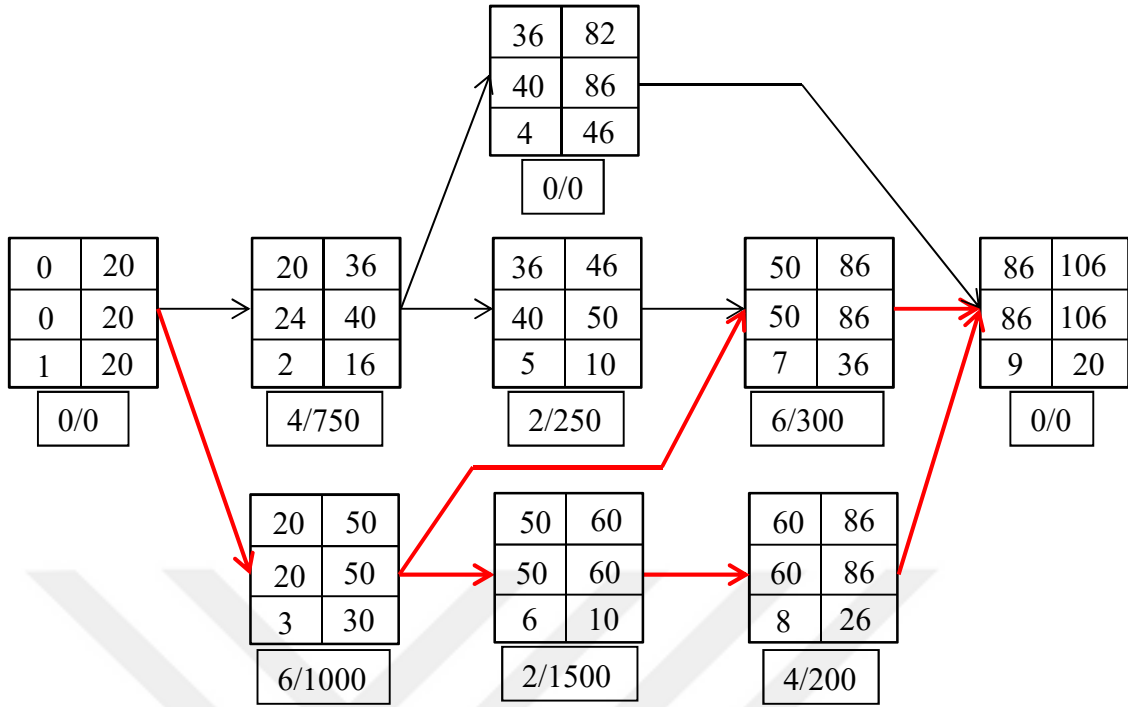
Alternatifler arasından en düşük eğime sahip 9 nolu işlem seçilmiştir. Serimin son işlemi olması bakımından diğer işlemler herhangi bir etkiye uğramadan 4 günlük zorlamanın tamamı kullanılabilir. Elde edilen yeni değerler Şekil 5.3.'te gösterilmektedir.



Şekil 5.3. İkinci hızlandırmadan sonraki durum ve kritik yolların gösterimi

İkinci hızlandırmanın ardından toplam proje süresi 110 gün olmaktadır. 375 TL/gün birim gider ile 4 günlük zorlama gerçekleştirildiğinden bir önceki çözüme 1500 TL ilave gider eklenmek suretiyle 75700 TL toplam doğrudan gider değeri bulunmaktadır.

Üçüncü hızlandırma için eğimleri aynı olan 1 nolu işlem ile “7+8” nolu işlemler seçilebilmektedir. Bu noktada, tüm işlem zamanlarını etkilerken, kritik yollarda değişiklik yapmayan 1 nolu işlem seçilmiştir. Yeni duruma ait işlem zamanları Şekil 5.4.'te görülmektedir.

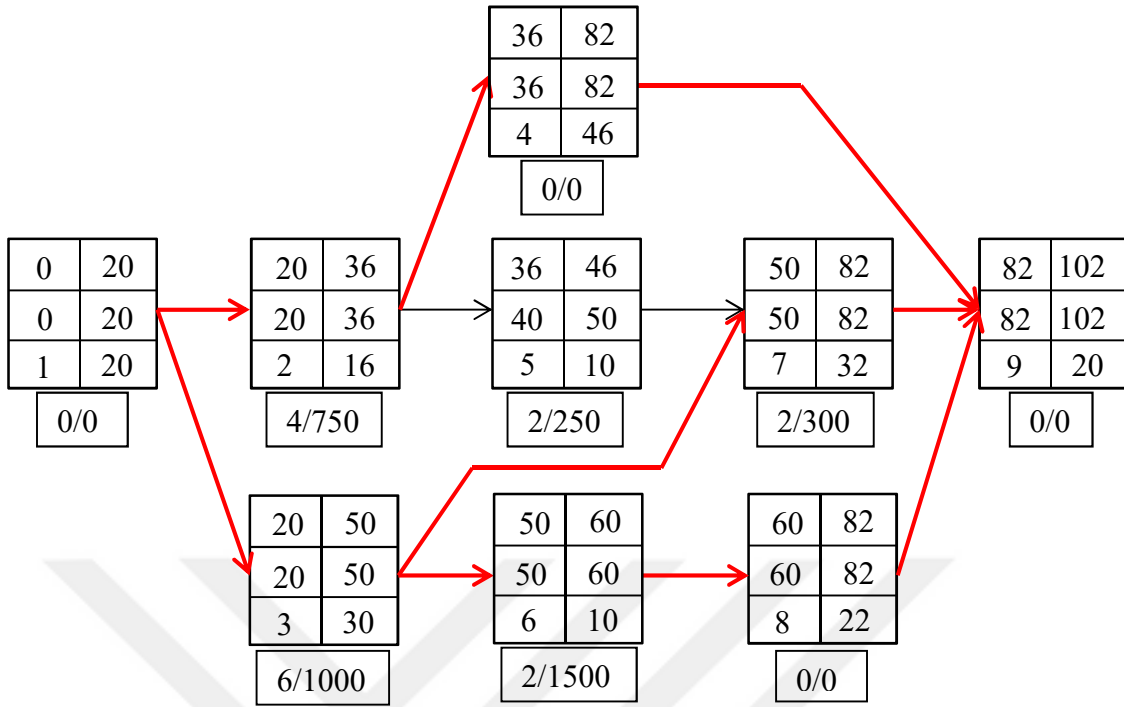


Şekil 5.4. Üçüncü hızlandırmadan sonraki durum ve kritik yolların gösterimi

Şekil 5.4.'te görüldüğü üzere üçüncü hızlandırmanın ardından toplam süre 106 güne düşmüştür. 1 nolu işlemin günlük zorlama bedeli 500 TL olduğundan 4 günlük zorlama 2000 TL olarak toplam gidere yansıtılmıştır. Böylece, toplam doğrudan gider yeni durumda 77700 TL'ye çıkmıştır.

Dördüncü hızlandırma, 7 ve 8 işlemlerinin birlikte 4 gün kısaltılması ile gerçekleştirilmiştir. Şekil 5.5.'te gösterildiği üzere 1-2-4-9 işlemleri yeni bir kritik yolu meydana getirmektedir. Dolayısıyla son durumda serimde 3 farklı kritik yol olduğu görülmektedir.

Yeni duruma göre toplam proje süresi 102 gün olarak hesaplanmaktadır. 7 ve 8 işlemlerinin birlikte zorlanması ile 4 gün karşılığında 2000 TL ilave gider artışı ortaya çıkmakta, bu durumda da yeni doğrudan gider 79700 TL olmaktadır.

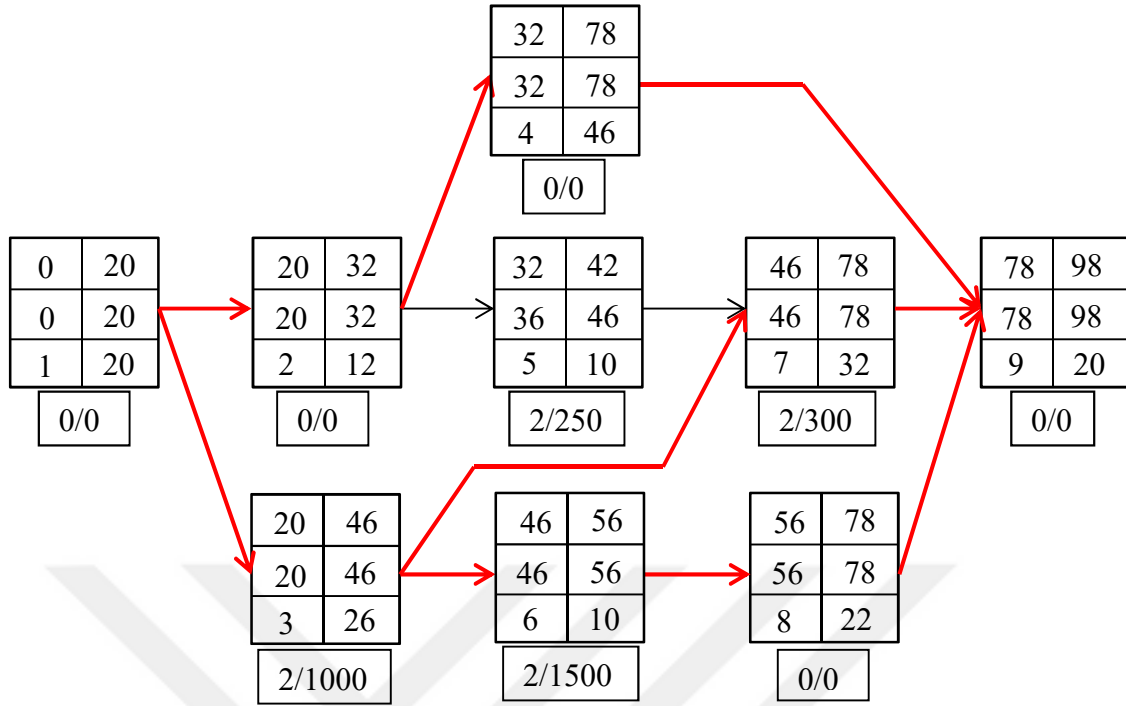


Şekil 5.5. Dördüncü hızlandırmadan sonraki durum ve kritik yolların gösterimi

Beşinci hızlandırma için iki alternatif mevcuttur. İlki, 2 ve 3 işlemlerinin birlikte zorlanmasıdır ve bu alternatifin eğimi 1750 TL/gündür. Diğeri ise 2, 6 ve 7 işlemlerinin birlikte zorlanması ile eğimi 2550 TL/gün olan alternatiftir. Toplam doğrudan gideri daha az etkilemesi bakımından ilk alternatif seçilmektedir.

Şekil 5.5.'te de görüldüğü üzere 3 nolu işlemin 6 güne kadar zorlanabilme imkanı varken 2 işlemi için 4 gün ile sınırlıdır. Bu durumda 2 ve 3 işlemleri birlikte 4 gün hızlandırılmaktadır. Elde edilen yeni işlem zamanları Şekil 5.6.'da verilmektedir.

Son hızlandırmanın ardından kritik yollarda değişiklik olmamıştır. Şekil 5.6. incelenecek olursa bu hızlandırmanın projenin son hızlandırması olduğu ve daha kısa proje süresi elde edilebilecek bir alternatif kalmadığı görülmektedir.



Şekil 5.6. Beşinci hızlandırmadan sonraki durum ve kritik yolların gösterimi

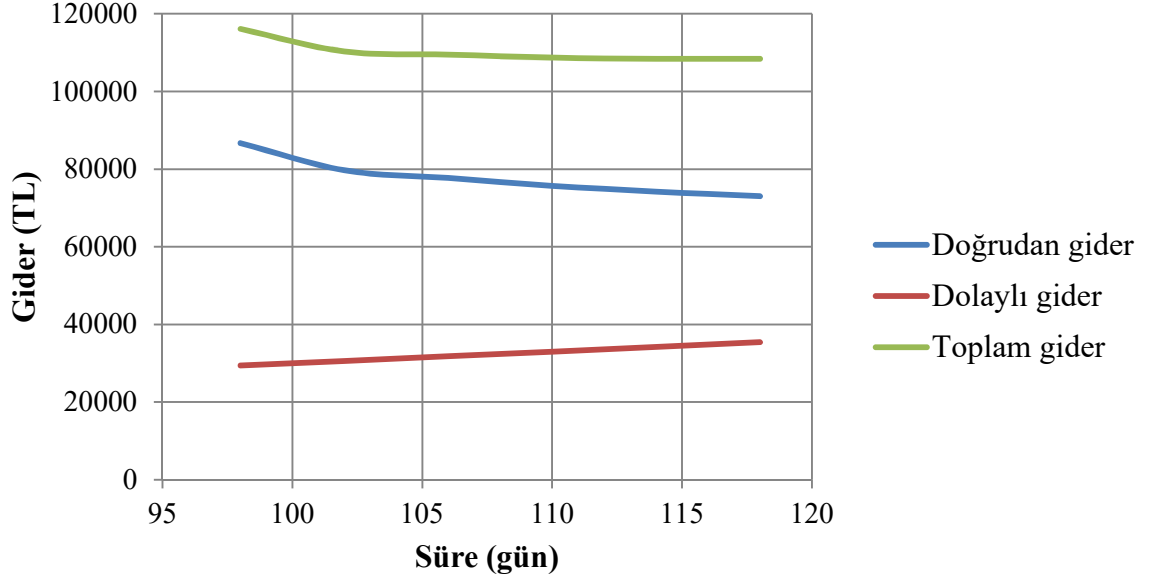
Beşinci ve son hızlandırmanın ardından toplam proje süresi 98 güne kadar düşürülebilmektedir. Bu süre değeri son ve kesin değerdir. Proje 1750 TL/gün bedeli ile 4 gün hızlandırıldığından 7000 TL ek gider oluşmaktadır. Bu gider bir önceki proje giderine eklendiğinde toplam 86700 TL proje doğrudan gideri elde edilmektedir.

Başlangıç çözümüne ilave 5 hızlandırma adımı sonrasında elde edilen tüm süre-gider eniyileme çözümleri Çizelge 5.2.'de verilmektedir.

Çizelge 5.2. Analitik süre-gider eniyilemesi sonuçları

| Toplam süre | Doğrudan gider | Dolaylı gider | Toplam gider |
|-------------|----------------|---------------|--------------|
| 118 | 73000 | 35400 | 108400 |
| 114 | 74200 | 34200 | 108400 |
| 110 | 75700 | 33000 | 108700 |
| 106 | 77700 | 31800 | 109500 |
| 102 | 79700 | 30600 | 110300 |
| 98 | 86700 | 29400 | 116100 |

Çizelge 5.2.'de verilen çözüm değerleri Şekil 5.7. ile grafik biçiminde ifade edilmektedir.

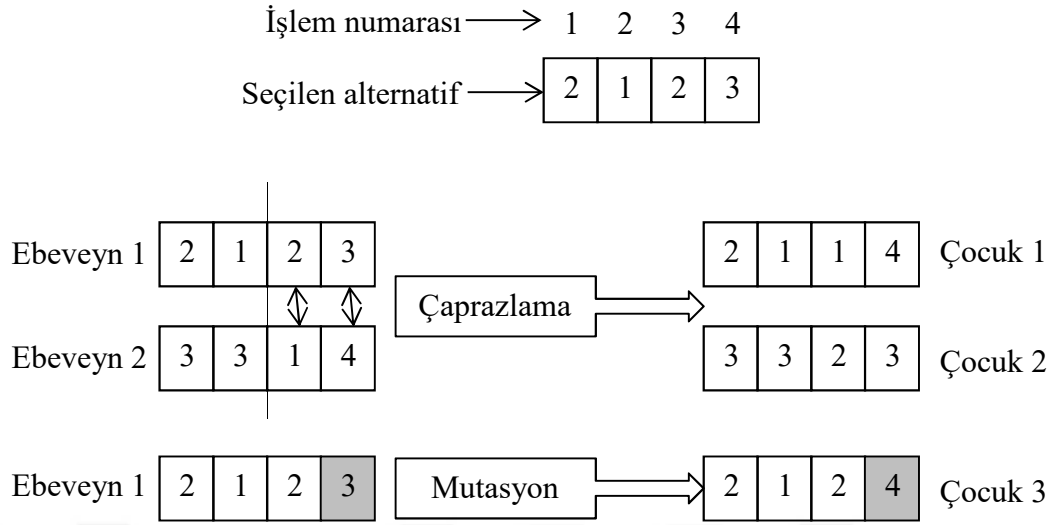


Şekil 5.7. Analitik süre-gider eniyileme çözüm grafiği

5.2. Genetik Algoritmalar (GA) ile Yaklaşım

Genetik algoritmalar (GA), optimizasyon alanında sıklıkla tercih edilen, etkin ve kullanışlı yöntemlerden biridir. Ayrıca GA'nın esnek yapısı çok çeşitli problemlere uyarlanabilmesini mümkün kılmaktadır. Bu nedenler doğrultusunda farklı mod alternatifleri bulunan bir projede süre-gider eniyilemesinin gerçekleştirilmesi amacıyla GA yöntemi seçilmiş ve aşağıda detayları anlatılmış olan probleme uygulanmıştır.

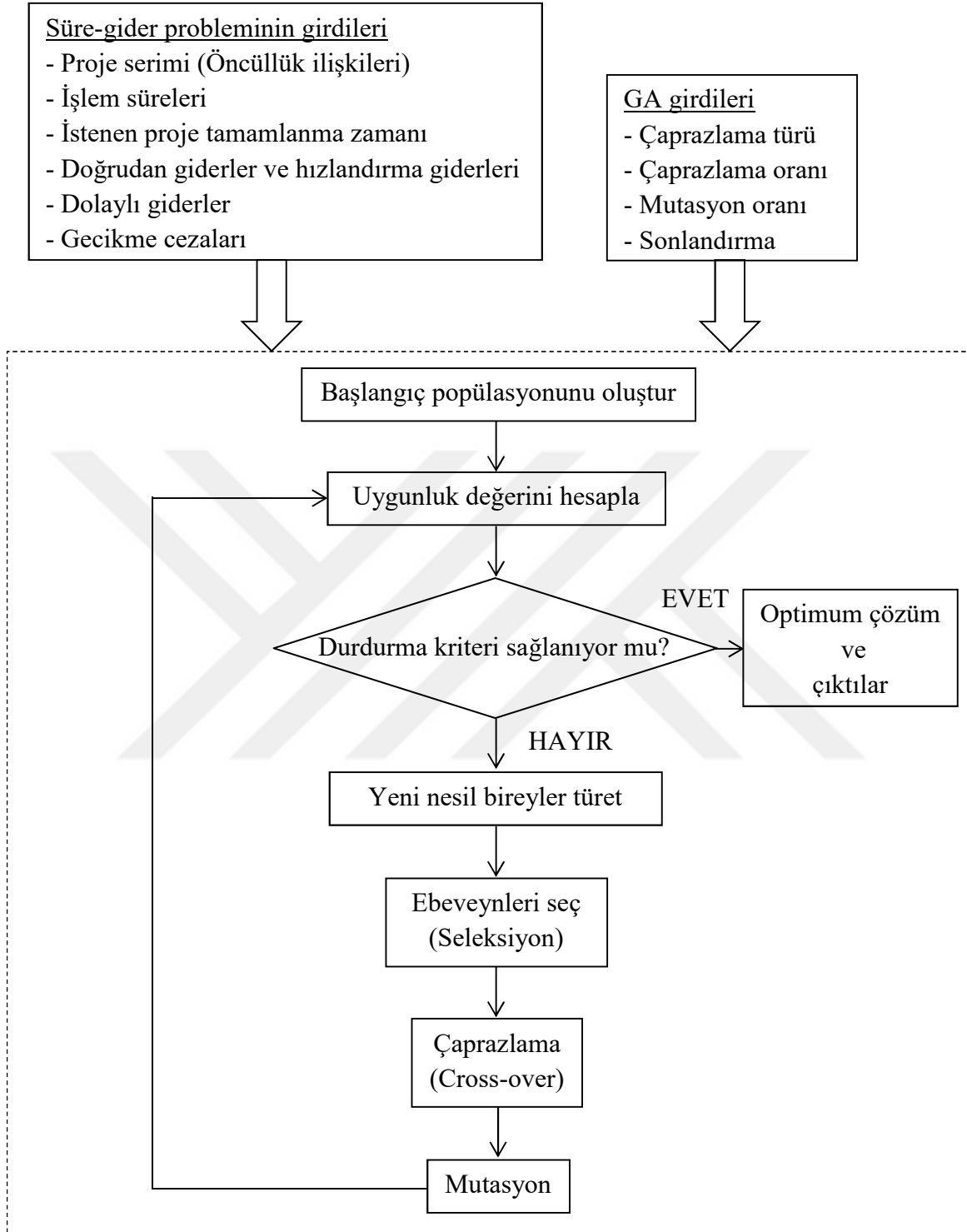
GA'da süre-gider eniyilemesinin modellenmesi amacıyla kromozom adı verilen şeritlerden yararlanılmaktadır. Toplam işlem sayısı kadar gene sahip olan bu kromozomlarda sırası ile her bir işleme ait mod seçeneklerinin bilgisi bulunmaktadır. GA'nın yapısı gereği bu kromozomların her biri muhtemel bir çözüm değerine ulaşmayı sağlayan karar değişkenlerinin dizisidir. Bu aday çözümlerin kromozom şeklinde gösterimi ile kromozomların çaprazlama ve mutasyon operatörlerine tabi tutulmasına ilişkin gösterim 4 işlemlerli bir proje verisi üzerinde Şekil 5.8.'de görülmektedir.



Şekil 5.8. GA'nın süre-gider eniyileme problemine uyarlanması

Şekil 5.8.'e göre çaprazlama ve mutasyon operatörleri algoritmadaki muhtemel çözümleri çeşitlendirirken, uyum değeri bakımından da çözümleri eleyerek en iyi olanların sonraki nesillere taşınmasını sağlamaktadırlar. Çaprazlama operatörü yardımıyla uyum değeri yüksek olması nedeniyle seçilmiş işlem dizilerinde karşılıklı olarak mod değişimleri yapılmaktadır. Böylelikle her bir çifte karşılık 2 yeni çözüm adayı daha elde edilmektedir. Mutasyon operatörü ise kromozomlara bireysel olarak uygulanmakta ve seçilen bir noktadan mod seçeneği dönüşüme uğramaktadır. Söz konusu operatörler ile genişleyen kromozom havuzunda kullanıcının tercihine göre ilk nesil kromozomlar tutulabilmekte ya da yeni neslin oluşturulmasının ardından yok edilebilmektedir.

Bölüm 4'te anlatılmış olan genetik algoritmaların akış şemasına ilave olarak süre-gider eniyilemesi adımlarından oluşan gösterim Şekil 5.9.'da sunulmaktadır.



Şekil 5.9. GA akış şemasına süre-gider eniyileme probleminin dahil edilmesi

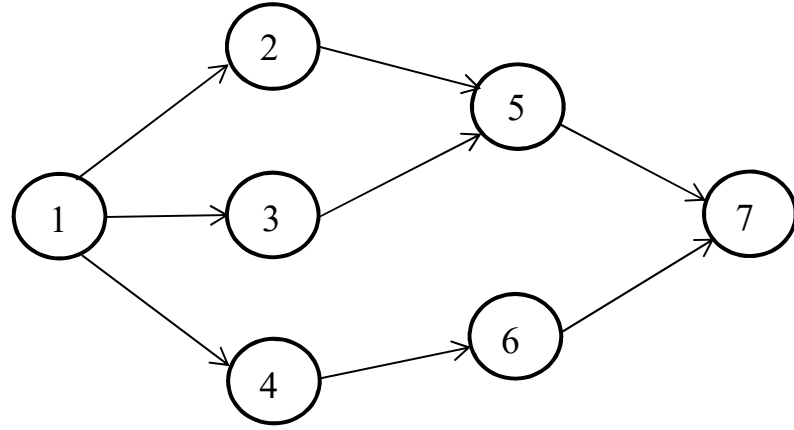
5.2.1 Genetik algoritmalar (GA) ile süre-gider eniyileme uygulaması

İşlem sayısı 7 olan projenin öncüllük ilişkileri, her bir işleme ait mod bilgileri, süre ve doğrudan gider değerleri Çizelge 5.3.'te yer almaktadır.

Çizelge 5.3. GA uygulaması için örnek proje verileri

| İşlem No | Öncül İşlem | Mod | Süre (gün) | Doğrudan Gider (TL) |
|----------|-------------|-----|------------|---------------------|
| 1 | - | 1 | 14 | 11500 |
| | | 2 | 20 | 9000 |
| | | 3 | 24 | 6000 |
| 2 | 1 | 1 | 15 | 6000 |
| | | 2 | 18 | 5000 |
| | | 3 | 20 | 3500 |
| | | 4 | 23 | 3000 |
| | | 5 | 25 | 2000 |
| 3 | 1 | 1 | 15 | 2500 |
| | | 2 | 22 | 2000 |
| | | 3 | 33 | 1500 |
| 4 | 1 | 1 | 12 | 22500 |
| | | 2 | 16 | 17500 |
| | | 3 | 20 | 15000 |
| 5 | 2, 3 | 1 | 22 | 10000 |
| | | 2 | 24 | 9000 |
| | | 3 | 28 | 7500 |
| | | 4 | 30 | 5000 |
| 6 | 4 | 1 | 14 | 20000 |
| | | 2 | 18 | 16000 |
| | | 3 | 24 | 10000 |
| 7 | 5, 6 | 1 | 9 | 15000 |
| | | 2 | 15 | 12000 |
| | | 3 | 18 | 11000 |

Çizelge 5.3.'te detayları verilen örnek projeye ilişkin serim gösterimi Şekil 5.10.'da verilmektedir.



Şekil 5.10. GA uygulaması için örnek proje serimi

Uygulama verilerinin bir optimizasyon programına tanıtılabilmesi için öncelikle probleme ilişkin matematiksel modelin kurulması gerekmektedir. Matematiksel model, problemin özelliğine göre şekillenen bir takım sınır koşulları ile probleme doğrudan etki eden karar değişkenlerinin oluşturduğu amaç fonksiyonundan meydana gelmektedir. Amaç fonksiyonu bir sonuç değerini maksimuma çıkarmayı ya da minimumda tutmayı hedefleyebilmektedir. Süre-gider eniyileme probleminde proje planlayıcıları hem sürenin hem de giderin mümkün olan en küçük değerlere çekilebilmesini hedeflediklerinden söz konusu problemin amaç fonksiyonu en küçükleyen fonksiyon olarak da adlandırılan minimizasyon türünde olmaktadır.

Örneğe ilişkin kısıt fonksiyonları (5.2), (5.3), (5.4) ve (5.5) nolu denklemlerde tanımlanmaktadır. Buna göre 1 nolu denklem ile algoritma 0. günden başlamaktadır. (5.3) nolu eşitsizlik ile son işlem olan 7. işlemin başlama zamanı ile 7. işlemin j. moddaki süresinin toplamının, izin verilen proje tamamlanma süresine eşit veya ondan küçük olması gerektiği ifade edilmektedir. (5.4) nolu eşitsizlik, her a öncül işleminin başlama zamanı ile j. moddaki süre değeri toplamının kendinden sonra gelen (ardılı) b işleminin başlama zamanına eşit veya ondan küçük olması gerektiği anlamına gelmektedir. Son kısıt denklemi olan (5.5) nolu ifade ise i'den n'ye kadar tüm işlemlerde j'den m'ye kadar yalnız 1 mod seçiminin yapılabileceğini göstermektedir. Böylelikle x_{ij} karar değişkeninin ikili (binary) değişken olarak da adlandırılan 0-1 değişken türünde olduğu anlaşılmaktadır.

Kısıt fonksiyonları:

$$S_1 = 0; \quad (5.2)$$

$$S_7 + \sum_{j=1}^{m_7} S_{7j} \cdot x_{7j} \leq S_{max}; \quad (5.3)$$

$$S_a + \sum_{j=1}^{m_a} S_{aj} \cdot x_{aj} \leq S_b; \quad a \rightarrow b \text{ tüm öncüller için } a, b=1, \dots, 7; \quad (5.4)$$

$$\sum_{i=1}^n \sum_{j=1}^{m_i} x_{ij} = 1; \quad (5.5)$$

Amaç fonksiyonu (Minimizasyon fonksiyonu):

$$g_t = \sum_{i=1}^n \sum_{j=1}^{m_i} g_{ij} x_{ij} + g_d [S_n + \sum_{j=1}^{m_n} (S_{ij} \cdot x_{ij})] \quad (5.6)$$

(5.6) nolu amaç fonksiyonu toplam proje giderini minimize etmeyi hedeflemektedir. Bunun için denklem 3 ayrı parçada ele alınabilir. İlk kısımda her bir işlemin j'den m'ye kadar tüm modlardaki doğrudan gider değerleri hesaplanmaktadır. İkinci kısımda ise sırasıyla i'den n'ye kadar işlemlerin başlama zamanına her moda ait süre değeri eklenerek toplam proje süresi bulunmakta, bu süreyi günlük dolaylı gider ile çarparak toplam dolaylı gider hesaplanmaktadır.

Projenin özelliğine bağlı olarak erken tamamlama teşviki ya da gecikme cezası olması durumunda ise denkleme üçüncü bir bileşen daha katılmaktadır. Bu durumda proje tamamlanma süresi mod alternatiflerine göre hesaplanarak izin verilen son tamamlanma zamanından çıkarılmakta, bu fark değeri birim erken tamamlanma teşvikiyle ya da birim gecikme cezası ile çarpılmaktadır. Denklem ilk iki hesaplamının toplamından üçüncü sıradaki teşvik kazancının çıkarılması ya da gecikme cezasının eklenmesi sonucu minimum proje tamamlanma giderini vermektedir. Bu durum (5.7) nolu denklemde gösterilmektedir.

$$g_t = \sum_{i=1}^n \sum_{j=1}^{m_i} g_{ij} x_{ij} + g_d [S_n + \sum_{j=1}^{m_n} (S_{ij} \cdot x_{ij})] - k \cdot \{S_{max} - [S_n + \sum_{j=1}^{m_n} (S_{ij} \cdot x_{ij})]\} \quad (5.7)$$

Yukarıdaki denklemlerde g_t , proje toplam giderini, g_{ij} , i. işleme ait j. mod giderini, x_{ij} , i. işleme j. modun atanmasını, g_d , günlük dolaylı gideri, S_n , n. işlemin başlama zamanını, m_n , mod seçeneklerini, n, projedeki işlem sayısını, S_{ij} , i. işlemin j. mod süresini, k, erken tamamlanma teşvikini, S_{max} , aşılmaması gereken tamamlanma süresini göstermektedir. Hesaplamalarda $S_{max} = 70$ gün, $g_d = 500$ TL/gün olarak alınmıştır. Erken tamamlama ya da gecikme durumlarında herhangi bir kazanç/kayıp öngörülmemiştir.

Proje süre ve gider bilgileri yukarıda verilmiş olan uygulama için GA modeli MATLAB R2012a programında kodlanmış, Intel Core2, 4GB RAM, Windows10 64-bit özelliklerine sahip bir bilgisayarda çalıştırılmıştır.

GA modeline ilişkin parametreler test edilmiş ve en uygun değerler aşağıdaki şekilde tanımlanmıştır:

Popülasyon büyüklüğü = 50 ;

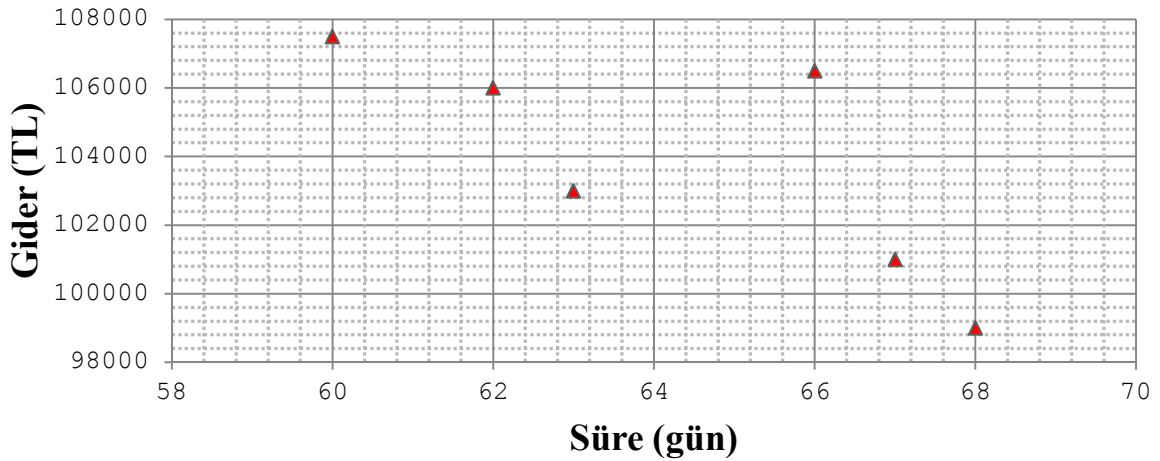
Çaprazlama oranı = 0,65 ;

Mutasyon oranı = 0,005.

Yapılan süre-gider eniyilemesi sonucunda Çizelge 5.4.'te verilen değerlere ulaşılmaktadır. Sonuçlara göre projenin en erken 60 gün ve 107500 TL gider ile tamamlanabileceği görülmektedir. Ayrıca bu sonuca ilişkin her bir işleme ait seçilmiş modlar da yine çizelgeden görülmektedir.

Çizelge 5.4. GA ile süre-gider eniyileme sonuçları

| İşlem No. | | | | | | | | Proje Tamamlanma Süresi (gün) | Proje Toplam Gideri (TL) |
|---------------|---|---|---|---|---|---|---|-------------------------------|--------------------------|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | |
| | 1 | 1 | 1 | 3 | 4 | 3 | 1 | 68 | 99000 |
| | 1 | 1 | 1 | 3 | 3 | 3 | 1 | 67 | 101000 |
| | 1 | 1 | 1 | 3 | 3 | 2 | 1 | 66 | 106500 |
| Modlar | 1 | 1 | 1 | 2 | 2 | 3 | 1 | 63 | 103000 |
| | 1 | 1 | 1 | 3 | 2 | 2 | 1 | 62 | 106000 |
| | 1 | 1 | 1 | 1 | 1 | 3 | 1 | 60 | 107500 |



Şekil 5.11. GA ile süre-gider eniyileme grafiği

Elde edilen sonuçlar Şekil 5.11.'de gösterilmiştir. Aşılmaması gereken 70 gün süre sınırı göz önüne alındığında en düşük giderli alternatif 99000 TL gider ile 68 günde tamamlama seçeneği olmaktadır. Minimum süre değerini veren sonuç ile minimum gider değerini veren sonuç alternatifleri arasında 4 farklı süre-gider çifti daha yer almaktadır. Bu sonuçlar içerisinde 66 gün ile 106500 TL gider değerini veren sonucun optimal olmadığı görülmektedir.

5.3. Parçacık Sürü Optimizasyonu (PSO) ile Yaklaşım

Süre-gider eniyilemesi problemi hem sürenin hem de giderin minimize edilmek istenmesi bakımından kendi içerisinde iki amaç barındıran problemlerdir. Optimizasyon problemlerinde konvansiyonel yöntemler yalnızca tek amaca yönelik çözüm üretebildiklerinden PSO gibi metasezgisel algoritmaların Pareto eğrisi çıkararak çözümleri sunabilmesi oldukça avantajlıdır (Zhang ve Li, 2010).

5.3.1. Parçacık sürü optimizasyonu (PSO) ile süre-gider eniyileme uygulaması

Tez çalışmasının bu bölümünde, önceki bölümde detayları anlatılmış olan PSO'nun, süre-gider eniyilemesi yönünden kullanımı ve etkinliğini göstermek amacıyla, seçilen örnek proje üzerinde uygulamalar gerçekleştirilmektedir. Bu amaç doğrultusunda öncelikle, betonarme tek katlı sanayi yapısı ele alınarak sadeleştirilmiş inşaat projesinin verileri açıklanmaktadır.

Çizelge 5.5.'te öncüllük ilişkileri, alternatif uygulama modları, süreleri, kaynak gereksinimleri ve doğrudan gider değerleri görülmekte olan proje 37 işlemten meydana gelmektedir. Bu projede doğrudan gider oluşturması bakımından göz önüne alınan kaynak işçiler olup, kaynak kısıtı günlük 12 işçi olarak hesaba katılmaktadır.

Çizelge 5.5. PSO uygulaması için örnek proje bilgileri

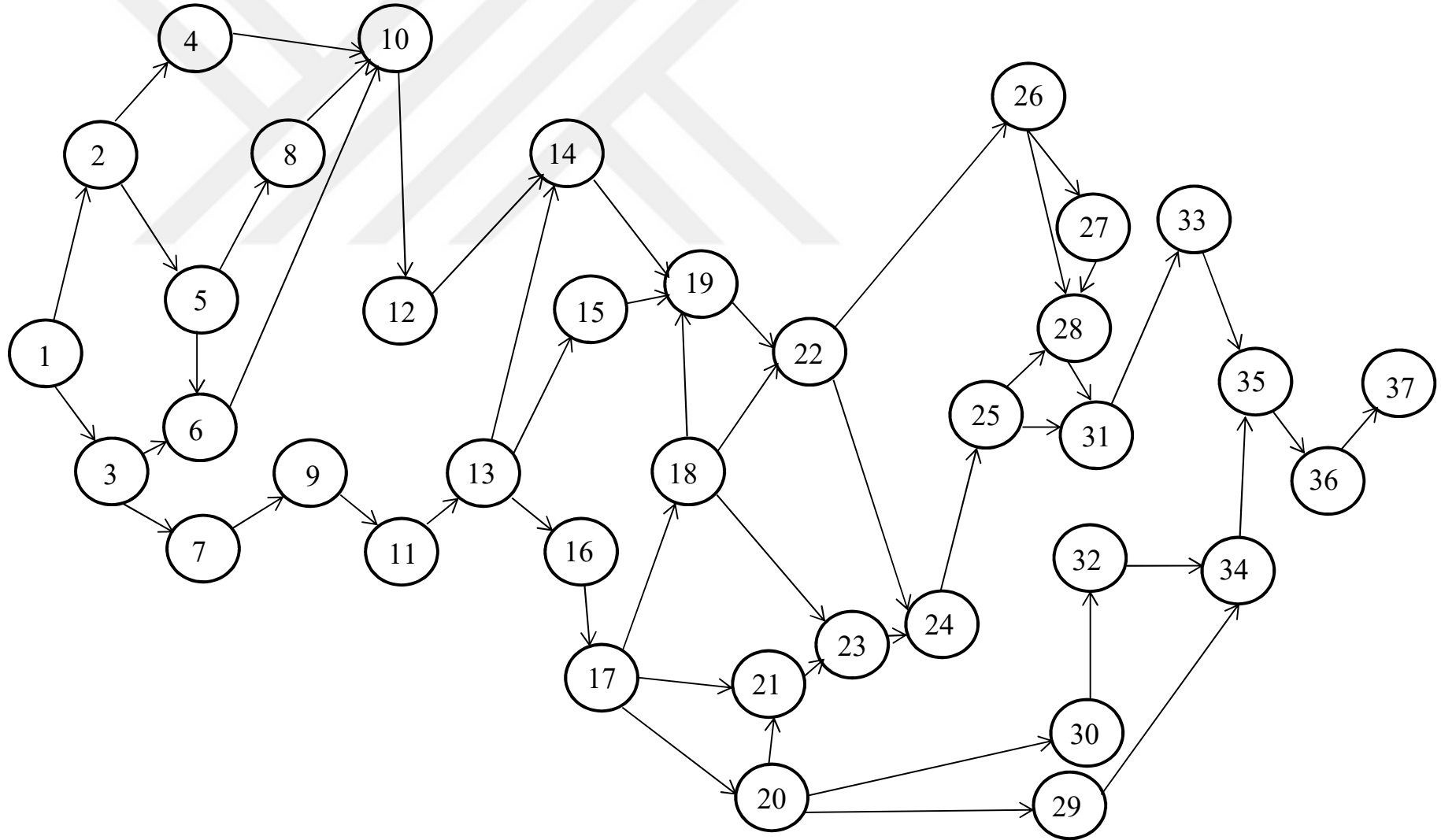
| İşlem No | İşlem Tanımı | Öncül İşlem | Mod | Süre (gün) | Kaynak Gereksinimi (işçi) | Doğrudan Gider* (TL) |
|----------|---------------------------------|-------------|-----|------------|---------------------------|----------------------|
| 1 | Başlangıç | - | - | 0 | - | - |
| 2 | Şantiye kurulumu | 1 | 1 | 15 | 2 | 3000 |
| | | 1 | 2 | 10 | 4 | 4000 |
| 3 | Zemin testleri | 1 | 1 | 11 | 2 | 2200 |
| 4 | Hafriyat | 2 | 1 | 20 | 4 | 8000 |
| | | 2 | 2 | 15 | 6 | 9000 |
| 5 | Kazık işleri | 2 | 1 | 21 | 5 | 10500 |
| | | 2 | 2 | 18 | 6 | 10800 |
| 6 | Kazık başlık kirişi | 3, 5 | 1 | 12 | 4 | 4800 |
| | | 3, 5 | 2 | 9 | 6 | 5400 |
| 7 | Kazık yükleme testleri | 3 | 1 | 10 | 2 | 2000 |
| 8 | Dolgu işleri | 5 | 1 | 10 | 3 | 3000 |
| | | 5 | 2 | 8 | 4 | 3200 |
| 9 | Kolon donatıları | 7 | 1 | 10 | 4 | 4000 |
| 10 | Döşeme betonu | 4, 6, 8 | 1 | 7 | 4 | 2800 |
| 11 | Kolon kalıpları | 9 | 1 | 9 | 4 | 3600 |
| | | 9 | 2 | 7 | 6 | 4200 |
| 12 | Çatı kiriş ve döşeme kalıpları | 10 | 1 | 12 | 5 | 6000 |
| | | 10 | 2 | 9 | 7 | 6300 |
| 13 | Kolon betonu | 11 | 1 | 10 | 4 | 4000 |
| 14 | Çatı kiriş ve döşeme donatıları | 12, 13 | 1 | 10 | 5 | 5000 |
| 15 | Çatı parapet duvar işleri | 13 | 1 | 8 | 5 | 4000 |
| 16 | Mekanik işler | 13 | 1 | 7 | 4 | 2800 |
| 17 | Çatı döşeme betonu | 16 | 1 | 7 | 4 | 2800 |
| 18 | Duvar imalatı | 17 | 1 | 15 | 4 | 6000 |
| | | 17 | 2 | 11 | 6 | 6600 |
| 19 | Elektrik işleri | 14,15,18 | 1 | 7 | 4 | 2800 |
| 20 | Pencere montajı | 17 | 1 | 7 | 3 | 2100 |
| 21 | Tavan işleri | 17, 20 | 1 | 7 | 4 | 2800 |
| 22 | Sıva işleri | 18, 19 | 1 | 10 | 4 | 4000 |
| 23 | Islak hacim işleri | 18, 21 | 1 | 14 | 3 | 4200 |
| | | 18, 21 | 2 | 9 | 5 | 4500 |
| 24 | Tesisat işleri | 22, 23 | 1 | 10 | 4 | 4000 |
| 25 | Bina dışı tabliye betonu | 24 | 1 | 7 | 4 | 2800 |
| 26 | Boya işleri | 22 | 1 | 14 | 3 | 4200 |
| | | 22 | 2 | 11 | 4 | 4400 |

Çizelge 5.5. PSO uygulaması için örnek proje bilgileri (devam)

| İşlem No | İşlem Tanımı | Öncül İşlem | Mod | Süre (gün) | Kaynak Gereksinimi (işçi) | Doğrudan Gider* (TL) |
|----------|---------------------------------|-------------|-----|------------|---------------------------|----------------------|
| 27 | Kapı montajı | 26 | 1 | 7 | 3 | 2100 |
| 28 | Demir doğrama işleri | 25,26,27 | 1 | 12 | 5 | 6000 |
| 29 | Dış cep. yalıtım ve sıva işleri | 20 | 1 | 10 | 4 | 4000 |
| | | 20 | 2 | 8 | 6 | 4800 |
| 30 | Çeşitli kurulum işleri | 20 | 1 | 7 | 2 | 1400 |
| | | 20 | 2 | 5 | 3 | 1500 |
| 31 | Ana giriş düzenleme ve montaj | 25, 28 | 1 | 3 | 3 | 900 |
| 32 | Dış cephe boya | 30 | 1 | 7 | 4 | 2800 |
| 33 | Tamamlama ve temizlik | 31 | 1 | 4 | 2 | 800 |
| 34 | Çevre düzenlemesi | 29, 32 | 1 | 15 | 2 | 3000 |
| | | 29, 32 | 2 | 11 | 3 | 3300 |
| 35 | Kontrol ve denetleme | 33, 34 | 1 | 5 | 2 | 1000 |
| 36 | Kusurların giderilmesi | 35 | 1 | 10 | 2 | 2000 |
| | | 35 | 2 | 6 | 4 | 2400 |
| 37 | Teslim | 36 | 1 | 1 | 1 | 100 |

*Doğrudan gider = süre*kaynak gereksinimi*100TL/işçi/gün.

Çizelge 5.5 ile işlem öncülleri tanımlanan projeye ait serim oluşturulmuş ve Şekil 5.12.'de verilmiştir.



Şekil 5.12. PSO uygulaması için örnek proje serimi

Çizelge 5.5.'ten de görüldüğü üzere projeye ait işlemlerin bazıları işçi sayısının artışı ile daha kısa sürelerde bitirmeye uygun olduğundan, alternatif durumlara ait süre-gider değerleri, süre-gider ilişkisi kesikli tipte olacak şekilde verilmektedir.

PSO algoritmasının probleme uyarlanabilmesi amacıyla öncelikle aşılmaması gereken proje tamamlanma süresine ihtiyaç duyulmaktadır. Bu amaçla çeşitli planlama tekniklerinden yola çıkılarak yapılan tahminlere göre 200 gün kısıtlayıcı değer olarak atanmıştır. Ardından muhtemel bir aralıktan başlangıç popülasyonu rasgele şekilde üretilmektedir. Elde edilen her değer bir sonuç adayı olarak parçacıklar tarafından temsil edilir. Parçacık konumları amaç fonksiyonuna göre uyum değerlerini göstermektedir. Bu işlemler devam ederken eğer bir parçacık öncekinden daha iyi uyum değeri gösterirse mevcut konumu güncellenir ve l_{best} olarak adlandırılır. Daha sonra parçacıklar komşu parçacıklar ile kıyaslanarak sürünün en yüksek uyum değeri belirlenir ve global en iyi konum g_{best} adını alır. Önceki bölümde anlatılan hız denklemleri ile hız güncellemesi yapılarak parçacığın sınırlar dahilinde yer değiştirip değiştirmediği kontrol edilir. Sınırı geçen parçacıklar için konum güncellemesi gerekli iken içerideki parçacıklarda herhangi bir işlem yapılmaz. Sonlandırma kriterine kadar tekrarlı bir şekilde sürdürülen bu prosedür, kriterin sağlanmasının ardından g_{best} ile çözümü vermektedir.

Parçacık sürü optimizasyonu (PSO) kullanarak, süre-gider eniyilemesi prosedürleri uygulanan proje için, yukarıda bahsedilen çözüm algoritmaları MATLAB R2012a programında kodlanmış, Intel Core2, 4GB RAM, Windows10 64-bit özelliklerine sahip bir bilgisayarda çalıştırılmıştır. PSO algoritmasını içeren kodlama satırlarının bir bölümü Şekil 5.13.'te verilmektedir. Buradan da görülebileceği üzere, çok sayıda test ve deneme analizleri sonrasında iterasyon sayısı 500 olarak uygun bulunmuş ve sürü büyüklüğü 20 seçilmiştir. Detayları Bölüm 4'te anlatılmış olan PSO parametrelerinden hızlandırma katsayıları $c_1 = c_2 = 2$, atalet katsayıları ise $p_{min} = 0,9$, $p_{max} = 1,1$ alınarak algortmada tanımlanmıştır. Uygulama sonrası elde edilen sonuçlar aşağıda detaylı şekilde verilmektedir.

```
Editor - C:\Users\G2\Desktop\PSO_DR_galbayrak.m
File Edit Text Go Cell Tools Debug Desktop Window » »
+ 1.0 + ÷ 1.1 x % % %
1  LB=[];
2  UB=[];
3  m=37;
4  n=20;
5  wmax=1.1;
6  wmin=0.9;
7  c1=2;
8  c2=2;
9  maxite=500;
10 maxrun=10;
11 for run=1:maxrun
12     run;
13     for i=1:n
14         for j=1:m
15             x0(i,j)=round(LB(j)+rand()*(UB(j)-LB(j)));
16         end
17     end
18     x=x0;
19     v=0.1*x0;
20     for i=1:n
21         f0(i,1)=obfun(x0(i,:));
22     end
23     [fmin0,index0]=min(f0);
24     pbest=x0;
25     gbest=x0(index0,:);
26     ite=1;
27     tolerance=1;
28     while ite<=maxite && tolerance>10^-12
29         w=wmax-(wmax-wmin)*ite/maxite;
30         for i=1:n
31             for j=1:m
32                 v(i,j)=w*v(i,j)+c1*rand()*(pbest(i,j)-x(i,j))+.
33                 +c2*rand()*(gbest(1,j)-x(i,j));
34             end
35         end
36         for i=1:n
37             for j=1:m
38                 x(i,j)=x(i,j)+v(i,j);
39             end
40         end
41         for i=1:n
42             for j=1:m
43                 if x(i,j)<LB(j)
44                     x(i,j)=LB(j);
```

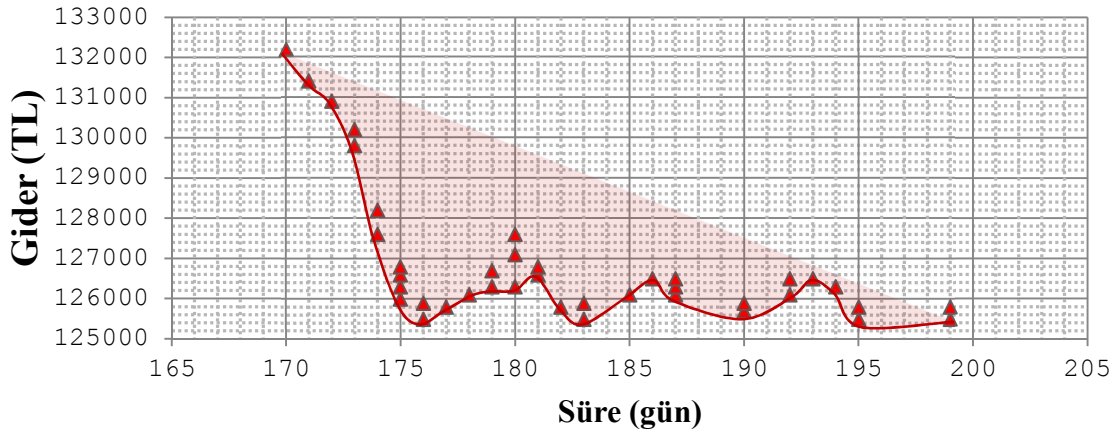
Şekil 5.13. PSO algoritmasına ait MATLAB kodlamaları

PSO ile yapılan çözümlenmeye göre süre-gider eniyileme değerleri 170-199 gün ile 125500-132200 TL gider değerleri arasında değişmektedir. Bu çözüm değerleri Çizelge 5.6.'da görülmektedir.

Çizelge 5.6. PSO ile süre-gider eniyileme sonuçları

| Çözüm No | Proje Tamamlanma Süresi (gün) | Doğrudan Gider (TL) | Çözüm No | Proje Tamamlanma Süresi (gün) | Doğrudan Gider (TL) |
|----------|-------------------------------|---------------------|----------|-------------------------------|---------------------|
| 1 | 170 | 132200 | 21 | 181 | 126600 |
| 2 | 171 | 131400 | 22 | 181 | 126800 |
| 3 | 172 | 130900 | 23 | 182 | 125800 |
| 4 | 173 | 129800 | 24 | 183 | 125500 |
| 5 | 173 | 130200 | 25 | 183 | 125900 |
| 6 | 174 | 127600 | 26 | 185 | 126100 |
| 7 | 174 | 128200 | 27 | 186 | 126500 |
| 8 | 175 | 126000 | 28 | 187 | 126100 |
| 9 | 175 | 126300 | 29 | 187 | 126300 |
| 10 | 175 | 126600 | 30 | 187 | 126500 |
| 11 | 175 | 126800 | 31 | 190 | 125700 |
| 12 | 176 | 125500 | 32 | 190 | 125900 |
| 13 | 176 | 125900 | 33 | 192 | 126100 |
| 14 | 177 | 125800 | 34 | 192 | 126500 |
| 15 | 178 | 126100 | 35 | 193 | 126500 |
| 16 | 179 | 126300 | 36 | 194 | 126300 |
| 17 | 179 | 126700 | 37 | 195 | 125500 |
| 18 | 180 | 126300 | 38 | 195 | 125800 |
| 19 | 180 | 127100 | 39 | 199 | 125500 |
| 20 | 180 | 127600 | 40 | 199 | 125800 |

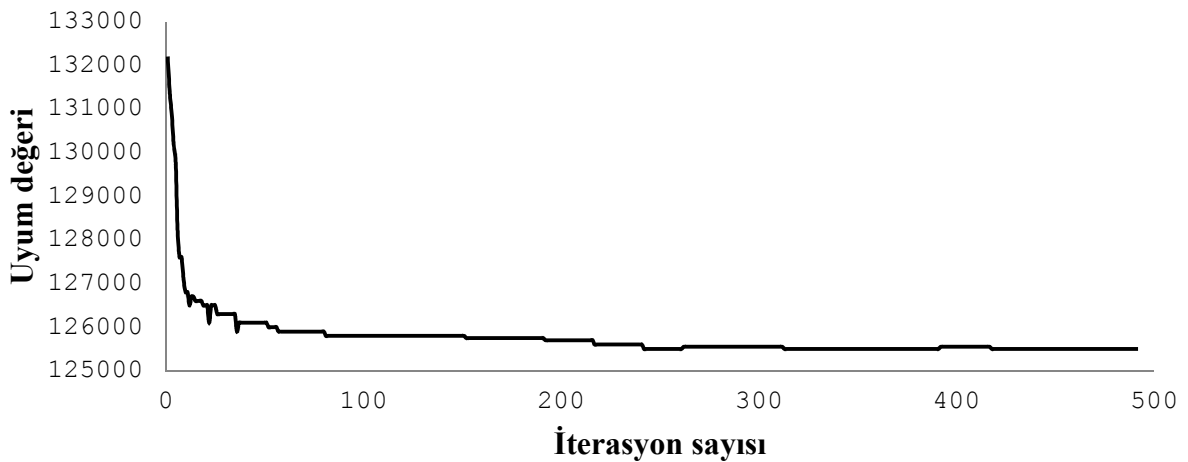
Yukarıda verilmekte olan süre-gider eniyilemesi sonuç çizelgesi Şekil 5.14. ile görselleştirilmiştir. Böylelikle söz konusu uygulama için Bölüm 3'te açıklanmış olan Pareto noktaları oluşturulmuştur. Buna göre varolan tüm kısıtlar ve sınır şartları altında proje minimum 170 günde tamamlanabilmektedir. Bu süre değerine karşılık gelen doğrudan gider değeri ise 132200 TL olmaktadır. Benzer şekilde proje 199 güne kadar uzatılabilmekte ve mümkün olan en düşük gider değeri olan 125500 TL değerine ulaşabilmektedir. Ancak sonuçlar toplu halde incelendiğinde bu tercihin anlamlı olmayacağı 12 nolu çözüme karşılık gelen 176 gün-125500 TL eniyilemesinden görülebilmektedir. Mümkün olan en düşük gider değerini veren en kısa proje tamamlanma süresi olan bu nokta “optimal değer çifti” olarak adlandırılmaktadır.



Şekil 5.14. PSO ile süre-gider eniyileme sonuç grafiği

Şekil 5.14.'te görülen Pareto noktalarının birleştirilmesi sonucu Pareto eğrisi oluşturulmuştur. Bu eğrinin üzerinde yer alan süre-gider çiftlerinin tamamı optimaldir. Bu eğrinin yukarısında kalan renkli bölgedeki sonuçlar ise süre, gider ya da hem süre hem gider yönünden optimal olmayan çözümlerdir.

500 iterasyon ile sınırlandırılan algoritmada, Pareto eğrisini oluşturan değerler ilk 150 iterasyonda gözlenmiştir. Bu duruma ilişkin uyum değeri-iterasyon sayısı grafiği Şekil 5.15.'te verilmektedir. Verilen grafiğe göre, seçilen algoritma parametrelerinin ve iterasyon sayısının yeterli olduğu sonucuna varılmaktadır. Grafiğin anlık olarak düşüş gösterdiği noktalarda sürü içerisindeki parçacıkların yerel optimum noktalarından kaçındığı görülebilmektedir.



Şekil 5.15. Uyum değeri-iterasyon sayısı grafiği

6. BULGULAR VE TARTIŞMA

İnşaat projelerinde süre-gider eniyilemesi büyük ölçekli optimizasyon problemleri arasında yer almaktadır. Bu tez çalışması kapsamında birbirlerine kıyasla farklı nitelikleri, üstünlükleri, zayıflıkları ve farklı kullanım alanları bulunan tekniklerin süre-gider eniyilemesi bakımından değerlendirilmesi hedeflenmiştir. Bu amaçla öncelikle geçmiş 60 yıl öncesine dayanan “serim akış algoritması” ile probleme çözüm aranmaktadır. İşlem süreleri ve hızlandırma bedelleri bilinen problem serim akış algoritması ile yalnızca tek modlu olarak çözülebilmektedir. Buna göre Bölüm 3.3.2.’deki çözümlerden aşağıdaki bulgulara varılmaktadır:

- Başlangıçtaki hesaplama göre proje tamamlanma süresi 21 gün, toplam gideri ise 215000TL olmaktadır. Yapılan hızlandırmalar sonrasında süre 11 gün kısalmaya karşılık %52,3 azalarak 10 güne kadar düşürülmüş, bu iyileştirme toplam gidere ise %14,4 oranında artış olarak yansiyarak 246000 TL toplam gider değerine ulaşılmıştır.

- Yapılacak yatırımın türüne göre, proje planlaması ile elde edilmek istenen, en düşük süreye sahip seçenek değil de en düşük giderli seçenek ise bu durumda 17 gün süreye karşılık olarak 211000 TL toplam gider elde edilmektedir. Böylelikle ilk duruma göre toplam gider %2 oranında azaltılabilmektedir. Oran düşük görünse de bu seçenekte yalnızca giderden tasarruf değil aynı zamanda süreden de %19 oranında kazanç olduğu bulunmaktadır.

- Toplam gideri oluşturan unsurlar ayrı ayrı ele alınırsa 11 günlük kısaltma sonrasında işçilik giderleri %172 oranında artış göstermiş, malzeme giderleri öngörüleceği üzere süreden etkilenmemiştir. Dolaylı giderler ise hızlandırma sonrası şantiye ortamının daha kısa süre açık tutulmasına bağlı olarak %44 oranında azalmaktadır.

- Uygulama sonrasında serim akış algoritması yöntemi ile süre-gider eniyileme problemi karşısında etkin çözüm almanın mümkün olduğu, deterministik bir yöntem olduğu için de elde edilen çözüm değerlerinin kesinlik taşıdığı görülmekte ancak yöntemin uygulanabileceği problemlerin yalnızca tek modlu olabileceği sonucuna varılmaktadır.

Bununla beraber yöntem elle ve grafik yardımıyla uygulandığından hem emek hem de süre yönünden pratikliği bulunmamaktadır. Özellikle işlem sayısının artışı hesaplamalarda hatayı da bereberinde getireceği gibi bir noktadan sonra çözümü de imkansız hale sokabilecektir.

İkinci olarak Bölüm 5.1.'de uygulanan “analitik yöntem” kritik yol yöntemi esasına dayalı geleneksel bir doğrusal süre-gider eniyileme çözüm tekniğidir. Uygulama sonrasına ulaşılan bulgular aşağıda verilmektedir:

- Proje ilk hesaplamaya göre 118 gün süre ve 108400 TL toplam gider ile tamamlanabilmektedir. Yapılan hızlandırmalar neticesinde süre maksimum %17 oranında kısaltılabilmektedir. Bu durumda proje 98 günde tamamlanmakta ancak bu hızlandırma toplam giderde %7'lik artışa sebep olduğundan proje 116100 TL değerine yükselmektedir.

- Proje başlangıçta 118 güne karşılık 108400 TL süre-gider değerleri ile tamamlanabilmekteydi. Yapılan hızlandırma sonucunda 108400 TL toplam gider değerini veren daha düşük süreye sahip bir seçenek olduğu da görülmektedir. Bu seçeneğe göre toplam gider değişmeksizin projenin 114 günde tamamlanabilmesi mümkündür. Bunun sebebi başlangıç durumuna kıyasla doğrudan giderlerdeki artışın, dolaylı giderdeki azalışa eşit olup toplam gidere pozitif ya da negatif bir yansımanın meydana gelmemesidir.

- Başlangıç çözümünde doğrudan gider 73000 TL, dolaylı gider 35400 TL iken minimum süre olan 98 günde doğrudan gider %18,8 artış, dolaylı gider %17 azalış göstererek sırasıyla 86700 TL ve 29400 TL olmuşlardır.

- Analitik yöntem, serim akış algoritmasında olduğu gibi deterministik olduğundan ortaya çıkan çözümler kesindir ve herhangi bir yaklaşıklık ya da rassallık içermez. Benzer şekilde analitik yöntem de süre-gider fonksiyonları doğrusal olan problemlere uygulanabilmektedir. Bu nedenle işlem sayısı az ve kesin sonucun önemli olduğu tek modlu süre-gider problemlerinde bu yöntemler avantajlıdır. Ancak her iki yöntem de serim üzerinden ilerlediği için kapsamlı bir projede kullanışlı ve pratik olmayacaktır.

Çalışma kapsamında üçüncü uygulama Bölüm 5.2.1.'de, metasezgisel yöntemlerin evrimsel algoritmalar sınıfından seçilen genetik algoritmalar (GA) ile gerçekleştirilmiştir. Elde edilen bulgular aşağıda sıralanmaktadır:

- Proje tamamlanma süresi 60-68 gün aralığında değişmektedir. Gerçekte proje daha uzun sürelerde de tamamlanabilmektedir. Ancak oluşturulan genetik algoritma kodlamasında proje tamamlanma süresine 70 gün sınırlaması getirilerek GA'nın kolaylıkla müdahale edilebilen yapısı vurgulanmıştır. Bu durum elde edilen sonuçlarda da ilk mod yüksek sayıda bulunurken, daha fazla süre gerektiren 2., 3. ve 4. modların az sayıda yer almasını açıklamaktadır. Sonuçlar arasında 5. modun olmayışı da aynı sebebe dayanmaktadır.

- Proje minimum süre olan 60 günde tamamlanmak istenirse bu durumda 68 güne kıyasla %8,6 daha yüksek bir gidere neden olmaktadır. 60 gün sonucuna karşılık gelen mod alternatifleri her bir işlem için verilmektedir. Ayrıca diğer süre-gider çiftlerine ait bulgular da elde edilmiştir. Bu süre-gider çiftleri arasında 66 gün-106500 TL değerleri hem süreyi hem de gideri arttırmış olması bakımından karar vericiler açısından uygun bir çözüm olmamaktadır.

- GA ile çok modlu ve kesikli süre-gider fonksiyonuna sahip bir problem üzerinde süre-gider enyilemesi yapılmıştır. Konvansiyonel yöntemlere kıyasla bu özellik daha çok problemin çözülebilmesi bakımından önemli bir avantajdır. Ayrıca gerçek uygulamalarda doğrusal ilişkinin görüldüğü durumlar oldukça azdır. Doğrusallığın yerine kesikli süre-gider ilişkisi gerçek durumu yansıtmada daha başarılıdır.

- GA diğer metasezgisel yöntemlerde de olduğu gibi stokastik bir yöntemdir. Bu durum GA ile elde edilen sonuçların problemin karmaşıklığına da bağlı olarak her zaman optimum kesin çözümü vermeyeceğini ifade eder. Bununla beraber belirli bir yaklaşıklığın kabul edildiği durumlarda optimal çözümü almak bakımından GA çok modlu problemlere uygulanabilmektedir.

- GA süre-gider eniyilemesi için yeterli bir teknik altyapı ile bir kez oluşturulduktan sonra başka projelere de çeşitli uyarılama ve modifikasyonlarla adapte edilebilir. En baştan

tasarlama ve kodlama ihtiyacı duyulmaması yöntemle pratiklik kazandıran önemli bir unsurdur.

Süre-gider eniyilemesi probleminde kullanılan son yöntem Bölüm 5.3.1.'de uygulanan parçacık sürü optimizasyonudur (PSO). Yöntemin uygulandığı problemde alınan sonuçlar aşağıdaki gibidir:

- Proje çeşitli mod alternatifleri göz önüne alındığında 170-199 gün arasında tamamlanabilmektedir. En olumsuz mod durumlarının kombinasyonu ile oluşan 199 gün tamamlanma süresi, minimum süre olan 170 güne çekilmek istenirse %5,3 gider artışı meydana gelmektedir.

- PSO ile yapılan optimizasyon sonucu minimum 125500 TL toplam proje gideri bulunmaktadır. Bu gider değerini veren birden fazla tamamlanma süresi olduğu görülmektedir. Bu durumda minimum gidere karşılık gelen minimum süre seçilerek Pareto noktası elde edilir. Böylece 176 gün değeri optimal sonuç olmaktadır.

- PSO ile çözümü gerçekleştirilen probleme diğer yöntemlerden farklı olarak 12 işçi olarak belirlenen kaynak kısıtı eklenmiştir. Bunun yanısıra işlem sayısı artırılarak tek modlu olarak düşünüldüğünde dahi analitik yöntemle çözümü güç olan bir proje çok modlu ve Pareto çözümlü olarak ele alınmıştır. Böylelikle problemin yapısı karmaşıklaştırılarak PSO'nun avantajlı yönleri gösterilmektedir.

- PSO, GA gibi bir stokastik yöntemdir. Bu nedenle elde edilen sonuçlarda tam kesinlikten bahsedilemez. Ancak uygun algoritma parametrelerinin ve iterasyon sayılarının kullanılması durumunda çoğunlukla optimuma yaklaşmaktadır. Tam ve kesin sonucun istendiği problemlerde kullanılması doğru olmamakla birlikte, yeterli yaklaşımla kapsamlı problemlerde dahi optimum sonuç üretebilmektedir.

- Küçük ölçekli projelerde PSO algoritmasını oluşturmak ve probleme uygun biçimde kodlamak zahmetli olabilmektedir. Ancak bir kez oluşturulduktan sonra GA'da da olduğu gibi başka projelere adaptasyonu ve mevcut programa yeni özellikler eklemek konusunda oldukça esneklik.

7. SONUÇLAR VE ÖNERİLER

İnşaat projelerinde süre-gider ilişkisinin öneminin vurgulanması ve bu ilişkinin doğru şekilde modellenerek, uygun yöntemler yardımıyla, eniyilenmiş süre ve gider değerlerinin araştırılması bu çalışmanın esasını teşkil etmektedir. Bu amaç doğrultusunda öncelikle süre-gider eniyileme problemi tanıtılarak teorik bilgiler sunulmuştur. Ardından verilen bilgiler ışığında gerçekte karşılaşılabilecek problemlerin çözümüne yardımcı olacak uygulamalar geliştirilmiştir. Uygulamalarda tercih edilen gerek analitik, gerekse metasezgisel yöntemlerin birbirlerine göre üstün ve zayıf yanları bulunmaktadır. Bu noktada önemli olan yöntemlerin tanınması ve probleme uygun yöntemin seçilerek sonuçların doğru yorumlanabilmesidir.

Çalışma kapsamında süre-gider eniyilemesi problemi karşısında analitik ile metasezgisel yöntemler kıyaslanmaktadır. Yapılan örnek uygulamalar sonrasında az sayıda işleme sahip, kesin çözüm gerektiren, tek modlu, doğrusal süre-gider ilişkisi bulunan ve karmaşıklığı düşük bir problemde öncelikli olarak analitik yöntemlere başvurmak gerektiği sonucuna varılmıştır. Bu tür problemlerde algoritma yazım ve kodlama aşamalarına gerek duyulmaksızın konvansiyonel çözüm yöntemleriyle deterministik sonuçlar elde etmek olasıdır.

Çok sayıda işlemi bulunan, çok modlu, kesikli süre-gider ilişkisine sahip ve karmaşıklığı yüksek süre-gider eniyileme problemlerinde analitik yöntemler yetersiz kalmaktadır. Bu aşamada ise metasezgisel algoritmalar içerisinde seçilen genetik algoritmalar (GA) ve parçacık sürü optimizasyonu (PSO) yöntemlerinin süre-gider eniyileme problemine uyarlanmasının yerinde olduğu sonucu ortaya çıkmıştır. GA ve PSO yöntemlerinin kullanımında dikkat edilmesi gereken husus algoritma parametrelerinin uygun seçimidir. Deneme-yanılma, tecrübe ya da referans alma gibi şekillerde olabilen bu durum algoritmanın performansını doğrudan etkilemektedir.

Elde edilen sonuçlar göstermiştir ki, bir süre-gider fonksiyonu karşısında daha geniş uygulanabilirlik kapasitesi, esneklik ve adaptasyon yeteneği bakımından GA ve PSO, analitik yöntemlerden oldukça üstündür. GA ve PSO yöntemleri süre-gider eniyilemesi

bakımından kıyaslandığında ise nispeten daha yeni bir yöntem olan PSO'nun yerel optimumlardan kaçınma yönünden avantajı bulunmaktadır. Ancak PSO'ya önemli özellikler kazandıran algoritması GA'ya kıyasla daha karmaşık yapıda olduğundan daha uzun kodlama ihtiyacı doğurmaktadır.

İnşaat projelerinin her biri eşsiz ve kendine özgü koşullar barındıran niteliktedir. Bu nedenle projelerin türüne göre öncelikli unsur farklılık gösterebilir. İnşaat projelerinin tamamında süre ve gider önemli iken, bir kısmında süre ya da giderden biri oldukça kritik bir unsur olarak yer almaktadır. Bu durum projenin başlangıcında öngörülebileceği gibi, kimi zaman da proje uygulanmaya başladıktan sonra ortaya çıkan aksaklıklardan dolayı da ortaya çıkabilmektedir. Böyle durumlar karşısında yapılan hızlandırma hesaplamaları sonucunda birbirinden farklı pek çok süre-gider optimal çözümü elde edilebilir. Bu noktada önemli olan hangi proje koşulunda hangi optimal çözümün uygun olduğunu iyi analiz edebilmektir. Örneğin süre bakımından kritik durumdaki bir proje için giderin en küçüklenmesi optimum olmayacağı gibi, küçük bütçeli yatırımlardaki hızlandırmalarda da sürenin en küçüklenmeye çalışılması problemi doğru çözümden uzaklaştıracaktır. Pareto noktası olarak ifade edilen kavram burada anlam kazanmaktadır. Süre ya da giderden biri en küçüklenirken diğerinde meydana gelecek artışın minimumda tutulması ile elde edilen bu nokta optimal çözüme işaret etmektedir. Pareto noktalarını geleneksel yöntemler ile tespit edebilmek ancak tesadüf eseri mümkün olabilirken, uygulanan metasezgiseller aracılığı ile bu noktalara kolaylıkla ulaşılabilmektedir.

Bu çalışmada metasezgisel yöntemler olarak GA ve PSO göz önüne alınmıştır. Bu iki yöntem dışında sayıları oldukça fazla olan metasezgisellerin süre-gider eniyilemesi probleminin çözümüne uygun olup olmadıkları araştırılmaya açık bir konudur. Gelecek çalışmalarda kapsamın genişletilerek farklı yöntemlerin de bu alanda denenmesi önerilmektedir.

Ayrıca çözüm yöntemlerinin yanı sıra, süre-gider eniyileme probleminin farklı kriterlerle de ele alınması başka çalışmalara konu teşkil edebilir. Örneğin risk ya da finansman gibi üçüncü bir unsura problemde yer verilebileceği gibi, belirsiz ortam koşullarının da problemin teorisine olan etkisi araştırılmaya değer görülmektedir.

KAYNAKLAR DİZİNİ

- Abbass, H. A., 2001, Mbo: Marriage in honey bees optimization-a haplometrosis polygynous swarming approach, In *Evolutionary Computation, Proceedings of the 2001 Congress on IEEE*, Vol. 1, 207–214.
- Abdel-Raheem, M., Khalafallah, A., 2011, Using electimize to solve the time-cost tradeoff problem in construction Engineering, *Computing in Civil Engineering Proceedings*, 250-257.
- Abd El Razek, R. H., Diab, A. M., Hafez, S. M., Aziz, R. F., 2010, Time-cost-quality trade-off software by using simplified genetic algorithm for typical repetitive construction projects, *International Journal of Civil, Environmental, Structural, Construction and Architectural Engineering*, 4(1), 22-31.
- Afruz, E. N., Najafi, A. A., Roghanian, E., Mazinani, M., 2014, A multi-objective imperialist competitive algorithm for solving discrete time, cost and quality trade-off problems with mode identity and resource-constrained situations, *Computers and Operations Research*, 50, 80–96.
- Afshar, A , Kaveh, A., Shoghli, O. R., 2007, Multi-objective optimization of time-cost-quality using multi-colony ant algorithm, *Asian Journal of Civil Engineering (Building and Housing)*, 8(2), 113-124.
- Afshar, A., Ziaraty, A., Kaveh, A., Sharifi, F., 2009, Nondominated archiving multicolony ant algorithm in time–cost trade-off optimization, *Journal of Construction Engineering and Management*, 135(7), 668–674.
- Aladini, K., Afshar, A., Kalhor, E., 2011, Discounted cash flow time-cost trade-off problem optimization; ACO approach, *Asian Journal Of Civil Engineering (Building And Housing)*, 12(4), 511-522.
- Albayrak, G., Özdemir, İ., 2016, Yapı projelerinin süre-maliyet optimizasyonunda metasezgisel algoritma kullanımı, *Adıyaman Üniversitesi Mühendislik Bilimleri Dergisi*, 5 (2016), 39-49.
- Atashpaz-Gargari, E., Lucas, C., 2007, Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition, *Evolutionary Computation*.
- Azaron, A., Tavakkoli-Moghaddam, R., 2007, Multi-objective time-cost trade-off in dynamic PERT networks using an interactive approach, *European Journal of Operational Research*, 180 (3), 1186-1200.
- Burns, S., Liu, L., Feng, C., 1996, The LP/IP hybrid method for construction time-cost trade-off analysis, *Construction Management and Economics*, 14,265-276.

KAYNAKLAR DİZİNİ (devam)

- Castro-Lacouture, D., Suer, G. A., Gonzalez-Joaqui, J., Yates, J. K., 2009, Construction project scheduling with time, cost, and material restrictions using fuzzy mathematical models and critical path method, *Journal of Construction Engineering and Management*, 135 (10), 1096-1104.
- Chen, T. C., Tsai, P. W., Chu, S. C., Pan, J. S., 2007, A novel optimization approach: bacterial-ga foraging, In *Innovative Computing, Information and Control Second International Conference*, 391-391.
- Chen, S. P., Tsai, M. J., 2011, Time-cost trade-off analysis of project networks in fuzzy environments, *European Journal of Operational Research*, 212 (2), 386–397.
- Charnes, A., Cooper, W. W., Thompson, G. L., 1964, Critical path analyses via chance constrained and stochastic programming, *Operations Research*, 12 (3), 460-470.
- Chu, Y., Mi, H., Liao, H., Ji, Z., Wu, Q. H., 2008, A fast bacterial swarming algorithm for high-dimensional function optimization, *Evolutionary Computation*, 3135-3140.
- Çivicioğlu Beşdok, P., 2012, Transforming geocentric cartesian coordinates to geodetic coordinates by using differential search algorithm, *Computers & Geosciences*, 46, 229-247.
- Çivicioğlu Beşdok, P., 2013, Artificial cooperative search algorithm for numerical optimization problems, *Information Sciences*, 229, 58-76.
- Çivicioğlu Beşdok, P., 2013, Backtracking search optimization algorithm for numerical optimization problems, *Applied Mathematics and Computation*, 219 (15), 8121-8144.
- De, P., Dunne, E. J., Wells, C. E., 1995, The discrete time-cost tradeoff problem revisited, *European Journal Operational Research*, 81, 225-238.
- De, P., Dunne, E. J., Ghosh, J. B., Wells, C. E., 1997, Complexity of the discrete time-cost tradeoff problem for project networks, *Journal of Operations Research*, 45, 302-306.
- Deckro, R. F., Hebert, J. E., Verdini, W. A., Grimsrud, P. H., Venkateshwar, S., 2006, Nonlinear time/cost tradeoff models in project management, *Computers and Industrial Engineering*, 28 (2), 219-229.
- Dorigo, M., 1992, Optimization, learning and natural algorithms, Ph.D Thesis, Politecnico di Milano, Italy.

KAYNAKLAR DİZİNİ (devam)

- Drias, H., Sadeg, S., Yahi, S., 2005, Cooperative bees swarm for solving the maximum weighted satisfiability problem, In *Computational Intelligence and Bioinspired Systems*, 318-325
- El-Rayes, K., Kandil, A., 2005, Time-cost-quality trade-off analysis for highway construction, *Journal of Construction Engineering and Management*, 131 (4), 477-486.
- Eshtehardian, E., Afshar, A., Abbasnia, R., 2008, Time-cost optimization: using ga and fuzzy sets theory for uncertainties in cost. *Construction Management and Economics*, 26(7), 679–691.
- Eshtehardian, E., Afshar, A., Abbasnia, R., 2009, Fuzzy-based moga approach to stochastic time-cost trade-off problem, *Automation in Construction*, 18(5), 189-198.
- Elmaghraby, S. E., 1993, Resource allocation via dynamic programming in activity networks, *European Journal of Operational Research*, 64, 199-215.
- Eusuff, M. M, Lansey, K. E., 2003, Optimization of water distribution network design using the shuffled frog leaping algorithm, *Journal of Water Resources Planning and Management*, 129(3),210-225.
- Ezeldin, S. A., Soliman, A., 2009, Hybrid time-cost optimization of nonserial repetitive construction projects. *Journal of Construction Engineering and Management*, 135 (1), 42-55.
- Falk, J., Horowitz, J. L., 1972, Critical path problems with concave cost-time curves. *Manage Sci*, 19, 446-455.
- Feng, C. W., Liu, L., Burns, S. A., 1997, Using genetic algorithms to solve construction time-cost trade-off problems, *Journal of Computing in Civil Engineering*, 11 (3), 184–189.
- Feng, C. W., Liu, L., Burns, S. A., 2000, Stochastic construction time-cost trade-off analysis, *Journal of Computing in Civil Engineering*, 14 (2), 117.
- Ferreira, C., 2001, Gene expression programming: a new adaptive algorithm for solving problems, *Complex Systems*, 13(2), 87-129.
- Fister, I. J., Yang, X. S., Fister, I., Brest, J. and Fister, D., 2013, A brief review of nature-inspired algorithms for optimization, *Electrotechnical Review*, 80(3), 1-7.

KAYNAKLAR DİZİNİ (devam)

- Foldes, S., Sourmis, F., 1993, PERT and crashing revisited: mathematical generalizations, *European Journal of Operational Researches*, 64, 286-294.
- Fondahl, J. W., 1961, A non-computer approach to the critical path method for the Construction Industry, Technical Report, No. 9, The Construction Institution, Dept. of Civil Engineering, Stanford University, Stanford, California.
- Ford, L. R., Fulkerson, D. R., 1956, Maximal flow through a network. *Canadian Journal of Mathematics*. 8, 399–404.
- Fulkerson, D. R., 1961, A network flow computation for project cost curves, *Manage Sci*, 7, 167-178.
- Gandomi, A. H., Alavi, A. H., 2012, Krill herd: a new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*.
- Goldberg, D. E., 1989, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA.
- Golenko-Ginzburg, D., Gonik, A., 1997, Stochastic network project scheduling with non-consumable limited resources, *International Journal of Production Economics*, 48 (1), 29–37.
- Golpiri, H., Hejazi, S., 2014, A scenario based stochastic multi-objective modeling for time-cost-quality trade-off problem, *Project Management Development-Practice and Perspectives Third International Scientific Conference on Project Management in the Baltic Countries*, University of Latvia, Riga, Latvia.
- Golzarpoor, B., 2012, Time-cost optimization of large-scale construction projects using constraint programming, University of Waterloo, Master Thesis, Applied Science in Management Sciences, Waterloo, Ontario, Canada, 96 p.
- Haque, K. M. A., Hasin, M., 2012, Genetic algorithm for project time-cost optimization in fuzzy environment, *Journal of Industrial Engineering and Management*, 5(2), 364-381.
- Gutjahr, W. J., Strauss, C., Wagner, E., 2000, A stochastic branch-and-bound approach to activity crashing in project management, *Journal on Computing*, 12(2), 125–135.
- Havens, T. C., Spain, C. J., Salmon, N. G., Keller, J. M., 2008, Roach infestation optimization, In *Swarm Intelligence Symposium IEEE*, 1-7.

KAYNAKLAR DİZİNİ (devam)

- He, S., Wu, Q. U., Saunders, J. R., 2009, Group search optimizer: an optimization algorithm inspired by animal searching behavior, *Evolutionary Computation, IEEE Transactions*, 13(5), 973-990.
- Hedayatzadeh, R., Salmassi, F. A., Keshtgari, M., Akbari, R., Ziarati, K., 2010, Termite colony optimization: A novel approach for optimizing continuous problems, In *Electrical Engineering (ICEE) Conference*, 553-558.
- Hendrickson, C., Au, T., 1989, *Project Management for Construction*, Prentice Hall, Englewood Cliffs, NJ, USA.
- Hegazy, T., 1999, Optimization of construction time-cost trade-off analysis using genetic algorithms, *Canadian Journal of Civil Engineering*, 26 (6), 685–697.
- Hegazy, T., 2001, *Computer-Based Construction Project Management*, Pearson, p. 398.
- Hegazy, T., Ayed, A., 1999, Simplified spreadsheet solutions, Models for critical path method and time-cost tradeoff analysis, *Cost Engineering*, 47 (7), 26-33.
- Hegazy, T., Menesi, W., 2010, Critical path segments scheduling technique, *Journal of Construction Engineering and Management*, 136 (10), 1078-1085.
- Henderickson, C., Au, T., 1989, *Project Management for Construction*, Prentice-Hall, Englewood Cliffs, New Jersey.
- Hernandez, H., Blum, C., 2012, Distributed graph coloring: an approach based on the calling behavior of japanese tree frogs, *Swarm Intelligence*, 6(2), 117–150.
- Hillier, F. S., Lieberman G. J., 2012, *Introduction to Operations Research*, McGraw Hill; 9th edition.
- Hooshyar, B., Rahmani, A., Shenasa, M. A., 2008, Genetic algorithm to time cost trade off in project scheduling, presented *IEEE Congress on Evolutionary Computation*.
- Huang Y. S., Deng, J. J. Zhang Y. Y., 2008, TI time-cost-quality tradeoff optimization in construction project based on modified ant colony algorithm, *Proceedings of the Seventh International Conference on Machine Learning and Cybernetics*, Kunming, China, 1031-1035.
- Iranmanesh, H., Sikandari, M. R., Allahverdiloo, M., 2008, Finding pareto optimal front for multimode time, cost quality tradeoff in project scheduling, *World Academy of Science, Engineering and Technology*, 40, 346-350.

KAYNAKLAR DİZİNİ (devam)

- Issa, U. H., Eid, M. A., 2013, An application of genetic algorithms to time-cost-quality trade-off in construction industry, *Civil and Environmental Research*, 3 (12), 11-19.
- Jung, S. H., 2003, Queen-bee evolution for genetic algorithms. *Electronics Letters*, 39(6), 575-576.
- Kandil, A., El-Rayes, K., 2005, Parallel computing framework for optimizing construction planning in large-scale projects, *Journal of Computing in Civil Engineering*, 19 (3), 304–312.
- Kandil, A., El-Rayes, K., 2006, Parallel genetic algorithms for optimizing resource utilization in large-scale construction projects, *Journal of Construction Engineering and Management*, 132 (5), 491-498.
- Karaboğa, D., 2014, Yapay zeka optimizasyon algoritmaları, Nobel Yayın, 231 s.
- Karaboğa, D., Baştürk, B., 2007, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, *Journal of Global Optimization*, 39 (3), 459-471.
- Karcı, A., 2002, Genetik algoritmalarda ıraksama ve yerel çözümde kalma problemlerinin giderilmesi, Fırat Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Doktora tezi, 78 s.
- Kasaeian, A., Reza, O., Afshar, A., 2007, Nondominated archiving genetic algorithm for multi-objective optimization of time-cost trade-off, *Proceedings of the 8th WSEAS International Conference on Evolutionary Computing*, Vancouver, British Columbia, Canada.
- Kashan, A. H., 2009, League championship algorithm: a new algorithm for numerical function optimization, *Soft Computing and Pattern Recognition*, 43-48.
- Kaveh, A., Farhoudi, N., 2013, A new optimization method: Dolphin echolocation, *Advances in Engineering Software*, 59, 53-70.
- Kaveh, A., Khanzadi, M., Alipour, M., Rajabi Naraky, M., 2015, CBO and CSS algorithms for resource allocation and time-cost trade-off, *Periodica Polytechnica Civil Engineering*, 59(3), 361–371.
- Ke, H., Ma, W., Ni, Y., 2009, Optimization models and a GA-based algorithm for stochastic time-cost trade-off problem, *Applied Mathematics and Computation*, 215 (1), 308-313.

KAYNAKLAR DİZİNİ (devam)

- Kelly, J. E. Jr., 1961, Critical path planning and scheduling: mathematical basis, *Operations Research*, 9 (3),167-179.
- Kennedy, J., Eberhart, R., 1995, Particle swarm optimization. In *Neural Networks Proceedings*, vol. 4, 1942-1948.
- Klerides, E., Hadjiconstantinou, E., 2010, A decomposition-based stochastic programming approach for the project scheduling problem under time/cost trade-off settings and uncertain durations, *Journal of Computers and Operations Researches*, 37 (12), 2131-2140.
- Leu, S. S., Yang, C. H., 1999, GA-based multicriteria optimal model for construction scheduling, *Journal of Construction Engineering and Management*, 125 (6), 420-427.
- Li, H., Love, P. E. D., 1997, Improved genetic algorithms for time-cost optimization, *Journal of Construction Engineering and Management*, 123 (3), 233-237.
- Li, H., Cao, J. N., Love, P. E. D., 1999, Using machine learning and GA to solve time-cost tradeoff problems, *Journal of Construction Engineering and Management*, 125 (5), 347-353.
- Li, H., Hussein, M. A., Lei, Z., 2011, Incentive genetic algorithm based time-cost trade-off analysis across a build-operate-transfer project concession period, *Canadian Journal of Civil Engineering*, 38, 166-174.
- Li, X.L., Shao, Z. J., Qian, J. X., 2002, Optimizing method based on autonomous animats: Fish-swarm algorithm, *Xitong Gongcheng Lilun yu Shijian/System Engineering Theory and Practice*, 22 (11), 32.
- Liang, L., Burns, S. A., Chung-Wei, F., 1995, Construction time-cost trade-off analysis using LP/IP hybrid method, *Journal of Construction Engineering and Management*, 121 (4), 446-454.
- Liao, T. W., Egbelu, P. J., Sarker, B. R., Leu, S. S., 2011, Metaheuristics for project and construction management: A state-of-the-art review, *Automation in Construction*, 20 (5), 491-505.
- Lima F. B., Lins, A. J. J. C., Nascimento, A. I. S., Lima, M. P., 2008, A novel search algorithm based on fish school behavior, In *Systems, Man and Cybernetics IEEE International Conference*, 2646-2651.

KAYNAKLAR DİZİNİ (devam)

- Lucic, P., Teodorovic, D., 2001, Bee system: modeling combinatorial optimization transportation engineering problems by swarm intelligence, Triennial Symposium on Transportation Analysis, 441-445, 2001.
- Maia, R. D., Castro, L. D., Caminhas, W. M., 2012, Bee colonies as model for multimodal continuous optimization: The optbees algorithm. In Evolutionary Computation (CEC) Congress, 1-8.
- Mehrabian, A. R., Lucas, C., 2006, A novel numerical optimization algorithm inspired from weed colonization, Ecological Informatics, 1(4), 355-366.
- Meyer, W. L., Shaffer, L. R., 1963, Extensions of the Critical Path Method Through the Application of Integer Programming, Department of Civil Engineering University of Illinois, Illinois.
- Moder, J. J., Phillips, C. R., Davis, E. W., 1995, Project management with CPM, PERT and precedence diagramming, 3rd edition, Wisconsin: Blitz Publishing Company.
- Mohammadi, G., 2011, Using genetic algorithms to solve industrial time cost tradeoff problems, Indian Journal of Science and Technology, 4(10), 1273-1278.
- Mokhtari, H., Kazemzadeh, R. B., Salmasnia, A., 2011, Time-cost tradeoff analysis in project management: an ant system approach, IEEE Transactions on Engineering Management, 58 (1), 36-43.
- Mon, D. L., Cheng, C. H., Lu, H. C., 1995, Application of fuzzy distributions on project management, Fuzzy Sets and Systems, 73 (2), 227-234.
- Monghasemi, S., Nikoo, M. R., Fasae, M. A. K., Adamowski, J., 2015, A novel multi criteria decision making model for optimizing time-cost-quality trade-off problems in construction projects, Expert Systems with Applications, 42, 3089-3104.
- Moselhi, O., 1993, Schedule compression using the direct stiffness method, Canadian Journal of Civil Engineering, 20, 65 -72.
- Moussourakis, J., Haksever, C., Flexible model for time/cost tradeoff problem, Journal of Construction Engineering and Management, 130 (3), 307-314.
- Mozaffari, A., Fathi, A., Behzadipour, S., 2012, The great salmon run: a novel bio-inspired algorithm for artificial system design and optimisation, International Journal of BioInspired Computation, 4(5), 286-301.

KAYNAKLAR DİZİNİ (devam)

- Mucherino, A., Seref O., 2007, Monkey search: a novel metaheuristic search for global optimization, *Data Mining, Systems Analysis and Optimization in Biomedicine*, vol. 953, 162–173.
- Mungle S., Benyoucef L., Son Y. J., Tiwari M. K., 2013, A fuzzy clustering based genetic algorithm approach for time-cost-quality trade-off problems: A case study of highway construction project, *Engineering Applications of Artificial Intelligence*, 26 (8), 1953–1966.
- Ng, S. T., Zhang, Y., 2008, Optimizing construction time and cost using ant colony optimization approach, *Journal of Construction Engineering and Management*, 134 (9), 721–728.
- Özdemir, İ., 1987, Yatırımlarda süre-maliyet optimizasyonu, XI. Ulusal Yöneylem Araştırması Kongresi Tebliği, 120 s.
- Padro, F. P. C., Navarro, J.M., 2011, Bumblebees: a multiagent combinatorial optimization algorithm inspired by social insect behaviour.
- Pagnoni, A., 1990, *Project Engineering: Computer Oriented Planning and Operational Decision Making*, Springer-Verlag, Berlin.
- Parpinelli, R. S., Lopes, H. S., An eco-inspired evolutionary algorithm applied to numerical optimization, In *Nature and Biologically Inspired Computing (NaBIC) Third World Congress*, 466-471.
- Passino, K. M., 2002, Biomimicry of bacterial foraging for distributed optimization and control, *Control Systems*, 22(3), 52-67.
- Pathak, B. K., Srivastava, S., 2007, MOGA-based time cost tradeoffs: responsiveness for project uncertainties, presented at IEEE Congress on Evolutionary Computation.
- Pathak, B. K., Srivastava, S., Srivastava, K., 2008, Neural network embedded with multi-objective genetic algorithm to solve nonlinear time cost trade-off problem of project scheduling. *Journal of Scientific and Industrial Research*, 67, 124-131.
- Pathak, B. K., Srivastava, S., 2015, Effects of project uncertainties on nonlinear time-cost tradeoff profile, *Iranian Journal of Fuzzy Systems*, 12(4), 79-100.
- Patterson, J. H., Huber, D., 1974, A horizon-varying, zero-one approach to project scheduling, *Management Science*, 20 (6), 990-998.

KAYNAKLAR DİZİNİ (devam)

- Phillips, S., Dessouky, M. I., 1977, Solving the project time/cost tradeoff problem using the minimal cut concept, *Management Science*, 24 (4), 393–400.
- Prager, W., 1963, A structural method of computing project cost polygons, *Management Science*, 9 (3), 394-404.
- Premaratne, U., Samarabandu, J., Sidhu, T., 2009, A new biologically inspired optimization algorithm, In *Industrial and Information Systems (ICIIS) International Conference*, 279-284.
- Rahimi, M., Iranmanesh, H., 2008, Multi-objective particle swarm optimization for discrete time cost quality trade-off problems, *World Applied Sciences Journal*, 4(2), 270-276.
- Robinson, D. R., 1975, A dynamic programming solution to cost-time tradeoff for CPM, *Management Science*, 22 (2), 158-166.
- Ryan, C., Collins, J. J., O'Neill, M., 1998, Grammatical evolution: Evolving programs for an arbitrary language. In *Genetic Programming*, 83-96, Springer.
- Salmasnia, A., Mokhtari, H., Kamalabadi, I. N., 2011, A robust scheduling of projects with time, cost and quality considerations, *International Journal of Advanced Manufacturing Technology*, 60, 631-642.
- Shahsavari Pour N., Modarres M., Tavakkoli Moghadam R., 2012, Time-cost-quality trade-off in project scheduling with linguistic variables, *World Applied Sciences Journal*, 18 (3), 404-413.
- Shayeghi, H., Dadashpour, J., 2012, Anarchic society optimization based pid control of an automatic voltage regulator (avr) system, *Electrical and Electronic Engineering*, 2 (4), 199-207.
- Shi, Y., 2011, An optimization algorithm based on brainstorming process, *International Journal of Swarm Intelligence Research (IJSIR)*, 2(4), 35-62, 2011.
- Shrivastava R., Singh, Dubey G.C., 2012, Multi objective optimization of time cost quality quantity using multi colony ant algorithm, *International Journal of Contemporary Mathematical Sciences*, 7 (16), 773-784.
- Siemens, N., 1971, A simple CPM time-cost tradeoff algorithm, *Management Science*, 17(6), 354-363.
- Simon, D., 2008, Biogeography-based optimization. *Evolutionary Computation*, IEEE Transactions, 12(6), 702-713.

KAYNAKLAR DİZİNİ (devam)

- Sönmez R, Bettemir H., 2012, A hybrid genetic algorithm for the discrete time-cost trade-off problem, *Expert Systems with Applications*, 39 (13), 11428-11434.
- Storn, R., Price, K., 1997, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, 11(4), 341-359.
- Su, S., Wang, J., Fan, W., Yin, X., 2007, Good lattice swarm algorithm for constrained engineering design optimization, *Wireless Communications, Networking and Mobile Computing Conference*, 6421-6424.
- Sur, C., Sharma, S., Shukla, A., 2013, Egyptian vulture optimization algorithm—a new nature inspired metaheuristics for knapsack problem, In *The 9th International Conference on Computing and Information Technology*, 227-237, Springer.
- Talbot, F. B., 1982, Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case, *Management Science*, 28 (10), 1197–1210.
- Tang, R., Fong, S, Yang, X. S., Deb, S., 2012, Wolf search algorithm with ephemeral memory. *Digital Information Management Conference*, 165-172, 2012.
- Tareghian H. R., Taheri S. H., 2007, A solution procedure for the discrete time, cost and quality tradeoff problem using electromagnetic scatter search, *Applied Mathematics and Computation*, 190 (2), 1136-1145.
- Tavana, M., Abtahi, A. R., Khalili-Damghani, K., 2014, A new multi-objective multi-mode model for solving preemptive time-cost-quality trade-off project scheduling problems, *Journal of Expert Systems with Applications*, 41, 1830–1846.
- Teodorovic, D., Dell’Orco, M., 2005, Bee colony optimization—a cooperative learning approach to complex transportation problems, *Advanced OR and AI Methods in Transportation: Proceedings of 16th Mini–EURO Conference*, Poznan: Publishing House of the Polish Operational and System Research, 51-60.
- Ting, T. O., Man, K. L., Guan, S. U., Nayel, M., Wan, K., 2012, Weightless swarm algorithm (wsa) for dynamic optimization problems, In *Network and Parallel Computing*, 508-515.
- Vanhoucke, M., Debels, D., Sched, J., 2007, The discrete time/cost trade-off problem: Extensions and heuristic procedures, *Journal of Scheduling*, 10, 311–326.
- Wedde, H. F., Farooq, M., Zhang, Y., 2004, Beehive: An efficient fault-tolerant routing algorithm inspired by honey bee behavior, *Lecture Notes in Computer Science* 3172 LNCS, 83-94.

KAYNAKLAR DİZİNİ (devam)

- Xiong, Y., Kuang, Y., 2008, Applying an ant colony optimization algorithm-based multiobjective approach for time–cost trade-off. *Journal of Construction Engineering and Management*, 134(2), 153-156.
- Xu, Y., Cui, Z., Zeng, J., 2010, Social emotional optimization algorithm for nonlinear constrained optimization problems, In *Swarm, Evolutionary, and Memetic Computing*, 583-590.
- Yan, G. W., Hao, Z. J., 2013, A novel optimization algorithm based on atmosphere clouds model, *International Journal of Computational Intelligence and Applications*.
- Yang, I., 2007, Using elitist particle swarm optimization to facilitate bicriterion time-cost trade-off analysis, *Journal of Construction Engineering and Management*. 133 (7), 498-505.
- Yang, Q., 2009, Application of time-cost-quality tradeoff optimization model based on improved PSO algorithm to construction project, *Proceedings of Asia-Pacific Conference on Information Processing, Volume 02, IEEE Computer Society, Washington, DC, USA*.
- Yang, X. S., 2005, Engineering optimizations via nature-inspired virtual bee algorithms. vol. 3562, 317-323.
- Yang, X. S., 2008, *Nature-Inspired Metaheuristic Algorithms*, Luniver Press, UK.
- Yang, X. S., 2010, Firefly algorithm, stochastic test functions and design optimisation, *International Journal of Bio-Inspired Computation*, 2 (2), 78-84.
- Yang, X. S., 2012, Flower pollination algorithm for global optimization, *Unconventional Computation and Natural Computation*, 240–249.
- Yang, X. S., Deb, S., 2009, Cuckoo search via levy flights, In *Nature & Biologically Inspired Computing*, 210-214.
- Yavuz, F., 1992, İnşaat yatırımlarında süre-gider optimizasyonu-bir uygulama, *Anadolu Üniversitesi, Fen Bilimleri Enstitüsü, Yüksek lisans tezi*, 89 s.
- Yüksel, O., 2012, *Proje Tipi Üretimlerde Planlama*, Ege Üniversitesi Basımevi, 211 s.
- Zareei, M., Hassan-Pour, H. A., 2015, A multi-objective resource-constrained optimization of time-cost trade-off problems in scheduling project, *Iranian Journal of Management Studies*, 8(4), 653-685.

KAYNAKLAR DİZİNİ (devam)

- Zhang, H., 2014, The fuzzy time-cost-quality-environment-trade-off analysis of resource construction systems for large-scale hydroelectric projects, 7th International Conference on Management Science and Engineering Management, Lecture Notes on Electrical Engineering 241, Springer.
- Zhang H., Xing F., 2010, Fuzzy-multi-objective particle swarm optimization for time-cost-quality tradeoff in construction, *Automation in Construction*, 19 (8), 1067-1075.
- Zhang, L. M., Dahlmann, C., Zhang, Y., 2009, Human-inspired algorithms for continuous function optimization, In *Intelligent Computing and Intelligent Systems IEEE International Conference*, vol. 1, 318- 321.
- Zhang, Y., Ng, S., 2012, An ant colony system based decision support system for construction time-cost optimization. *Journal of Civil Engineering and Management*, 18(4), 580-589.
- Zheng, D. X. M., Ng, S. T., Kumaraswamy, M. M., 2004, Applying a genetic algorithm-based multiobjective approach for time-cost optimization, *Journal of Construction Engineering and Management*, 130 (2), 168.
- Zheng, D. X. M., Ng, S. T., Kumaraswamy, M. M., 2005, Applying pareto ranking and niche formation to genetic algorithm-Based Multiobjective Time-Cost Optimization. *Journal of Construction Engineering and Management*, 131(1), 81.
- Zhou, P. E. D., Wang, X, Teo, K. L. and Irani, Z., 2013, A review of methods and algorithms for optimizing construction scheduling,” *Journal of the Operational Research Society*, 64, 1091–1105.

ÖZGEÇMİŞ

Gülçağ ALBAYRAK, 1987 yılında Ankara'da doğmuştur. 2009 yılında Eskişehir Osmangazi Üniversitesi (ESOGÜ) İnşaat Mühendisliği Bölümü'nde lisans öğrenimini tamamlamasının ardından 2010 yılında Anadolu Üniversitesi İşletme Bölümü'nden mezun olmuştur. 2012 yılında ESOĞÜ Fen Bilimleri Enstitüsü İnşaat Mühendisliği Anabilim Dalı, Yapı Bilim Dalı'nda yüksek lisans öğrenimini tamamlamıştır. 2010 yılından bu yana ESOĞÜ İnşaat Mühendisliği Bölümü'nde araştırma görevlisi olarak görev yapmaktadır. Evli ve 1 çocuk annesidir.

