



T. C.

**ULUDAĞ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI
YÖNEYLEM BİLİM DALI**

**AYLAK ZAMANI EN KÜÇÜKLEME AMAÇLI ARAÇ ROTALAMA
PROBLEMİNİN GENETİK ALGORİTMA İLE ÇÖZÜMÜ**

DOKTORA TEZİ

ÖMER NURİ ÇAM

BURSA - 2018



T. C.

**ULUDAĞ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI
YÖNEYLEM BİLİM DALI**

**AYLAK ZAMANI EN KÜÇÜKLEME AMAÇLI ARAÇ ROTALAMA
PROBLEMİNİN GENETİK ALGORİTMA İLE ÇÖZÜMÜ**

DOKTORA TEZİ

ÖMER NURİ ÇAM

Danışman:


Prof. Dr. H. Kemal SEZEN

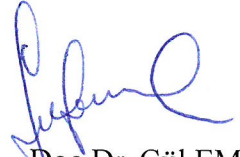
BURSA - 2018

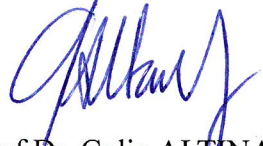
T. C.
ULUDAĞ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Ekonometri Anabilim, Yöneylem Bilim Dalı'nda 711217004 numaralı Ömer Nuri ÇAM'ın hazırladığı "Aylak Zamanı En Küçüklemeye Amaçlı Araç Rotalama Probleminin Genetik Algoritma İle Çözümü" konulu Doktora Tezi ile ilgili tez savunma sınavı, 17/05/2018 günü 10.00 - 12.30 saatleri arasında yapılmış, sorulan sorulara alınan cevaplar sonunda adayın tezinin/çalışmasının (başarılı/başarısız) olduğuna (oybirliği/oy çokluğu) ile karar verilmiştir.


Prof. Dr. H. Kemal SEZEN
Uludağ Üniversitesi
(Danışman)


Prof. Dr. Ayşe OĞUZLAR
Uludağ Üniversitesi


Doç. Dr. Gül EMEL
Uludağ Üniversitesi


Prof. Dr. Galip ALPINAY
Onyediy Eylül Üniversitesi


Doç. Dr. Hakan DEMİRGİL
Süleyman Demirel Üniversitesi

17.05.2018

Yemin Metni

Doktora tezi olarak sunduđum

“Aylak zamanı en küçükleme amaçlı araç rotalama probleminin genetik algoritma ile çözüümü” Başlıklı çalışmanın bilimsel araştırma, yazma ve etik kurallarına uygun olarak tarafımdan yazıldığına ve tezde yapılan bütün alıntıların kaynaklarının usulüne uygun olarak gösterildiđine, tezimde intihal ürünü cümle veya paragraflar bulunmadığına şerefim üzerine yemin ederim.

15.05.2018

Tarih ve İmza



Adı Soyadı:	Ömer Nuri ÇAM
Öğrenci No:	711217004
Anabilim Dalı:	Ekonometri
Programı:	Yöneylem
Statüsü:	Doktora



SOSYAL BİLİMLER ENSTİTÜSÜ
YÜKSEK LİSANS/DOKTORA İNTİHAL YAZILIM RAPORU

ULUDAĞ ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI BAŞKANLIĞI'NA

Tarih: 02/04/2018

Tez Başlığı / Konusu: AYLAK ZAMANI EN KÜÇÜKLEME AMAÇLI ARAÇ ROTALAMA PROBLEMİNİN
GENETİK ALGORİTMA İLE ÇÖZÜMÜ

Yukarıda başlığı gösterilen tez çalışmamın a) Kapak sayfası, b) Giriş, c) Ana bölümler ve d) Sonuç kısımlarından oluşan toplam 100 sayfalık kısmına ilişkin, 20/03/2018 tarihinde şahsım tarafından *TURNITIN* adlı intihal tespit programından (Turnitin)* aşağıda belirtilen filtrelemeler uygulanarak alınmış olan özgünlük raporuna göre, tezimin benzerlik oranı % 4 'tür.

Uygulanan filtrelemeler:

- Kaynakça hariç
- Alıntılar hariç/dahil
- 5 kelimedenden daha az örtüşme içeren metin kısımları hariç

Uludağ Üniversitesi Sosyal Bilimler Enstitüsü Tez Çalışması Özgünlük Raporu Alınması ve Kullanılması Uygulama Esasları'nı inceledim ve bu Uygulama Esasları'nda belirtilen azami benzerlik oranlarına göre tez çalışmamın herhangi bir intihal içermediğini; aksinin tespit edileceği muhtemel durumda doğabilecek her türlü hukuki sorumluluğu kabul ettiğimi ve yukarıda vermiş olduğum bilgilerin doğru olduğunu beyan ederim.

Gereğini saygılarımla arz ederim.

Tarih ve İmza

Adı Soyadı: Ömer Nuri ÇAM

Öğrenci No: 711217004

Anabilim Dalı: Ekonometri

Programı: Yöneylem

Statüsü: Y.Lisans X Doktora

Danışman
(Adı, Soyad, Tarih)

17.05.2018

* Turnitin programına Uludağ Üniversitesi Kütüphane web sayfasından ulaşılabilir.

Prof. Dr. H. Kemal SEZEN

ÖZET

Yazar Adı ve Soyadı : Ömer Nuri ÇAM

Üniversite : Uludağ Üniversitesi

Enstitü : Sosyal Bilimler Enstitüsü

Anabilim Dalı : Ekonometri

Bilim Dalı : Yöneylem

Tezin Niteliği : Doktora Tezi

Sayfa Sayısı : xiv + 86

Mezuniyet Tarihi : / / 20.....

Tez Danışman(lar)ı : Prof.Dr. H.Kemal SEZEN

AYLAK ZAMANI EN KÜÇÜKLEME AMAÇLI ARAÇ ROTALAMA PROBLEMİNİN GENETİK ALGORİTMA İLE ÇÖZÜMÜ

Neredeyse 60 yıl boyunca tartışılan problemlerden biri olan Araç Rotalama Problemi (ARP); farklı durumlara ilişkin yeni kısıtlar eklenerek çeşitlenmiş, 10'dan fazla temel problem türü modellenmiştir. Bu problemler incelendiğinde tümünün amaç fonksiyonlarının doğrudan veya dolaylı olarak maliyeti azaltmaya yönelik olduğu görülmektedir.

Bu çalışmada yeni bir ARP problemi tanıtılmaktadır. Problem Türkiye'de şehirler arası yolcu taşıyan bir firmanın operasyonlarının daha iyi yönetilmesi amacıyla yapılan araştırmada ortaya çıkmıştır. Diğer ARP'lerden farkı amaç fonksiyonundan kaynaklanmaktadır. Problemden; çalışıkça para kazanabilen araçların, bekleme sürelerinin (aylak zaman) en aza indirilmesi amaçlanmaktadır. Önceki problemlerle karşılaştırıldığında, araçların az çalışması değil çok çalışması- bazen en kısa yol yerine daha uzun yolu tercih etmeleri önerilmektedir. Diğer iki farkı da kısıtlarla ilgilidir; Bazı coğrafi noktalar birden fazla ziyaret edilmelidir ve alt tur oluşturulmasına izin verilebilir.

Problemin ayrıntılı sunumunda kullanılmak üzere firmanın operasyonlarının kümelendiği beş alt gruptan birinde yer alan 34 sefer örnek olarak seçilmiştir. Çözüm için genetik algoritma kullanılmıştır. Bulunan en küçük aylak zaman için verimlilik oranı %66,78 iken, en uzun aylak süre için %40,42 bulunmuştur. Bu durum eniyi çözümlerin bulunmasını garanti eden kesin (exact) yöntemlerin kullanımının önemine işaret etmektedir.

Problem bir çok taşıma (deniz yolu, hava yolu, kara yolu) şirketi için geçerlidir. Bazı imalat firmalarında otomatik kontrollü araçların fabrika içinde yaptıkları turlar, iş sırlama problemleri de bu yaklaşımla ele alınıp iyileştirilebilir. Gelecekte sürücüler için yasal kısıtlamaların olmadığı otomatik kontrollü insansız araçların daha iyi işletilmesine de uygulanabilecektir. Aylak sürenin düşürülmesi neticesinde beklentilerden biri, fazla sayıda olan araçların azaltılmasını mümkün kılmasıdır.

Problem için geliştirilecek çözümler, araç sayısı, filo yönetimi, servis-bakım maliyetlerinin, bilet fiyatlarının, karbon salınımının, israfın azalması, dünyada refahın artması gibi doğrudan, dolaylı etkilere yol açabilecektir.

Anahtar Kelimeler; Araç rotalama problemi (ARP) , Aylak sürenin en küçüklenmesi, En kısa yol, EKAZARP, zaman boyutlu coğrafi nokta



ABSTRACT

Name and Surname : Ömer Nuri ÇAM

University : Uludag University

Institution : Institue of Social Sciences

Field : ECONOMETRICS

Branch : Operational Research

Degree Awarded : PhD

Page Number : xiv + 86

Degree Date : / / 20.....

Supervisor (s) : Prof.Dr. H. Kemal SEZEN

SOLVING VEHICLE ROUTING PROBLEMS TO ACHIEVE MINIMUM IDLE TIME BY USING GENETIC ALGORITHM

Vehicle routing problem (VRP) is one of the problems discussed for almost 60 years among researchers, branched to more than 10 general subcategories, depending on problems' constraints or/and their aim functions. When these problems are examined, their aims are focused on reducing costs. (or cost related pollution etc.)

In this study, a new VRP approach is introduced. This approach is realized when to manage better the route operations of a company carrying long-distance passengers in Turkey. The difference of the problem based on its aim function. It suggests vehicles should work more because they make profit if they work. So this aim function suggests minimizing idle time of those vehicles. Opposite of VRP function which are examined till today, vehicles should work more and sometimes they should prefer long distance route. One of the result is that it is possible to reduce the number of vehicles used.

For a problem, 34 route surveyed from bus company. To solve and introduce that problem a software developed using genetic algorithm. As a result, one approach efficiency is %66.78 and other is %40.42. Outputs showed importance of using exact algorithms.

The problem can be applied to many transport companies (ship, plane, road). Also in some manufacturing companies, tours and work scheduling problems can be handled and improved with this approach for automatic controlled vehicles in factory. It can also be adapted to better maneuvering autonomous unmanned vehicles without legal restrictions for future drivers.

The solutions to be developed for the problem may lead directly or indirectly to the number of vehicles, fleet management, service-maintenance costs, ticket prices, carbon emissions, reduction of wastage, and welfare in the world.

Keywords; Vehicle routing problem, Minimizing idle time, shortest path, VRP MIT

Önsöz

Bu çalışmanın hazırlanmasında beni teşvik eden, farklı yaklaşımlarda bulunabilmeyi öğreten, mesleğim konusunda beni teşvik eden değerli öğretmenim Prof.Dr. H.Kemal SEZEN'e çok teşekkür ederim.

Desteklerini esirgemeyen eşim Özlem Hanıma ve yorgun olduğumda beni dinlendiren çocuklarım, Said Ali ve Halis Mete'ye de çok teşekkür ederim.

Annem, Babam, Ablam ve Abim'e de çok teşekkür ederim.



İÇİNDEKİLER

1 GİRİŞ.....	1
2 TEMEL KAVRAMLAR.....	4
3 ARAÇ ROTALAMA PROBLEMLERİ (ARP) HAKKINDA BİLGİLER.....	5
3.1 KAPASİTE KISITLI ARP (KARP).....	6
3.2 ZAMAN PENCERELİ ARP (ARP-ZP).....	8
3.3 ÇOK DEPOLU ARP (ÇDARP).....	10
3.4 PERİYODİK ARP (PARP).....	12
3.5 BÖLÜNEBİLİR TAŞIMALI ARP (BTARP).....	14
3.6 TOPLA DAĞIT ARP (TD-ARP).....	16
3.7 ÖNCE DAĞIT SONRA TOPLA ARP (ÖDST-ARP).....	17
3.8 DİĞER ARP PROBLEMLERİ.....	17
4 ARAÇ ÇİZELGELEME PROBLEMLERİ (AÇP).....	21
5 ARP ÇÖZÜM YÖNTEMLERİ.....	23
5.1 GENETİK ALGORİTMA.....	23
5.1.1 Kromozom Kodlamaları.....	25
5.1.2 Seçim Yöntemi.....	26
5.1.3 Çaprazlama.....	27
5.1.4 Mutasyon.....	28
5.1.5 Uygunluk Fonksiyonu.....	29
5.1.6 Sonlandırma Kriteri.....	29
6 AYLAĞ ZAMANI EN KÜÇÜKLEME AMAÇLI ARAÇ ROTALAMA PROBLEMİNİN TANIMI.....	30
6.1 PROBLEMİN AMAÇ FONKSİYONUNUN MATEMATİKSEL MODELİ.....	36
6.2 SIRADAKİ EN KÜÇÜK AYLAĞ SÜRE (SEKAS).....	37
6.3 SIRADAKİ EN BÜYÜK AYLAĞ SÜRE (SEBAS).....	37
7 GENETİK ALGORİTMA İLE ÇÖZÜMÜ.....	48
7.1 GENETİK ALGORİTMA ADIMLARI.....	48
7.1.1 Kromozom Oluşturma.....	48
7.1.2 Çaprazlama İşlemi.....	53
7.1.3 Mutasyon.....	54
7.1.4 Genetik Algoritma Parametreleri Belirleme.....	55
7.2 PROBLEMİN ÇÖZÜMÜ.....	60
8 TARTIŞMA VE SONUÇ.....	63
EK - 1 Cordeau problem türü.....	65
EK - 2 Solomon problem türü.....	67
EK - 3 Temel Genetik Algoritma Çizelgesi.....	68
EK - 4 34 Sefere ait problem verileri.....	69
EK - 5 Seferler Arası Aylak Sürelerin Dağılımı (Dakika).....	70
EK - 6 82 Sefere ait problem verileri.....	72
EK - 7 Taguchi Deney Tasarımı İçin Sonuçlar.....	74
EK - 8 ENAZARP Programı ve Kullanımı.....	75
EK - 9 Problem çözümünde 10 deneme ve oluşan sıralar.....	79

TABLolar

Tablo 1: KARP için örnek veri biçimi ve çözüm grafiği.....	8
Tablo 2: ARP sınıflandırma çalışması çıktısı.....	18
Tablo 3: Tek noktadan çaprazlama işlemi.....	27
Tablo 4: Çift noktadan çaprazlama işlemi.....	28
Tablo 5: İkili kodlamalı mutasyon gösterimi.....	28
Tablo 6: Permutasyon kodlamada mutasyon öncesi ve sonrası.....	29
Tablo 7: Seferlere ilişkin veri yapısı, örnek 4 sefer.....	32
Tablo 8: Seferlere ilişkin veri yapısı (Orjinal problemin 4 seferi).....	34
Tablo 9: Seferlerin sıralanmasına bağlı aylak zaman.....	35
Tablo 10: Seferlerin farklı sıralanmasına bağlı aylak zaman.....	35
Tablo 11: SEKAS için başlangıç sefer numarasına bağlı olarak değişen verimlilik oranları.....	38
Tablo 12: SEBAS için başlangıç sefer numarasına bağlı olarak değişen verimlilik oranları.....	40
Tablo 13: Tamamlanamamış SEKAS için örnek çıktı.....	42
Tablo 14: Geri İzlemeli SEKAS Yaklaşımı İçin Örnek Çözüm.....	43
Tablo 15: Geri İzlemeli SEBAS Yaklaşımı İçin Örnek Çözüm.....	44
Tablo 16: Geri izlemeli SEKAS ve SEBAS tüm başlangıç noktaları için verimlilik oranları (Sefer No- Verimlilik).....	46
Tablo 17: Denenecek genetik algoritma parametreleri.....	58
Tablo 18: Taguchi tepki tablosu, parametrelerin önem derecesi.....	58
Tablo 19: Parametrelerin ANOVA sonuçları.....	59
Tablo 20: Seçilen parametreler.....	60
Tablo 21: 10 deneyin sonuçları.....	61
Tablo 22: Çözüm sonucu elde edilen daha iyi sonuçlar.....	62

RESİMLER

Resim 1: Türkiye’de büyük şehirden tatil noktalarına yapılan sefer rotalarının örnek gösterimi.....	34
---	----

ŞEKİLLER

Şekil 1: Yay-Düğüm Gösterimi.....	4
Şekil 2: ARP genel kategorileri.....	5
Şekil 3: İki günlük periyotla rotalanmış 3 müşteri problem.....	12
Şekil 4: Örnek bölünebilir taşınmalı ARP için araç rotaları.....	14
Şekil 5: Basit kromozom yapısı.....	25
Şekil 6: Permutasyon kodlamalı kromozom.....	26
Şekil 7: Rassal iki sayıya göre sağa veya sola kaydırma işlemi.....	29
Şekil 8: Öncüllük ilişkisine göre işlemi mümkün olan en sağa kaydırma işlemi. .	29
Şekil 9: Gerçekleştirilecek seferlerin gösterimi.....	34
Şekil 10: Örnek problem için kromozom kodlama.....	48
Şekil 11: 3 şehirli 6 seferli örnek soru.....	50
Şekil 12: Başladığı noktaya geri dönen, halka dizilişine sahip sıralama.....	62

GRAFİKLER

Grafik 1: Çaprazlama oranı parametresi etkisi, (%50-%90).....	57
Grafik 2: Taguchi parametre sonuçları.....	59

Kısaltmalar listesi

AÇP : Araç Çizelgelem Problemi (VSP – Vehicle Scheduling Problem)

ARP : Araç Rotalama Problemi (VRP – Vehicle Routing Problem)

ARP-ZP : Zaman Pencereli ARP

ATP : Atölye tipi çizelgeleme problemi (JSP – Job shop scheduling)

BTARP : Bölünebilir Taşımali ARP

ÇDARP : Çok depolu ARP

EKAZARP : En Küçük Aylak Zamanlı Araç Rotalama Problemi

GSP : Gezgin Satıcı Problemi (GSP – Traveling Salesman Problem)

KARP : Kapasite kısıtlı ARP

Küm-ARP : Kümülatif ARP

MARP : Mesafe Kısıtlı ARP

MB-ARP : Mevkiye Bağlı ARP

MKARP : Mesafe ve Kapasite Kısıtlı ARP

NP-zor : Karmaşıklık seviyesi zor (NP-hard)

ÖDST-ARP : Önce Dağıt sonra Topla ARP

PARP : Periyodik ARP

SEBAS : Sıradaki en büyük aylak süre

SEKAS : Sıradaki En küçük aylak süre

TD-ARP : Topla Dağıt ARP

1 GİRİŞ

Araç Rotalama Problemi (ARP), müşterilere taleplerini ulařtırmak için optimum rotanın bulunmasını arayan kombinatoryal bir problem türüdür (Cordeau *vd.*, 2007). ARP, Gezgin Satıcı Probleminin (GSP) özel bir halidir. İlk tanımı yapıldığından (Dantzig ve Ramser, 1959) bu yana arařtırmalara konu olan ARP problemleri, her boyutuyla incelenmiş deęildir. Farklı kısıt ve amaçlarla yeni alt türleri arařtırmacılar tarafından oluşturulmaya ve çözülmeye devam etmektedir. Araç sürüş dinamiklerinin giderek otonom bir hal aldığı, yakıt türlerinin çeşitlendięi gelecekte yeni yaklaşımlar ve önerilerin olacağı kesindir.

ARP probleminin önemine müşteri açısından bakılırsa, bir durakta en az süre bekletilmesini, paket bekliyorsa hemen ulařtırılmasını veya gönderi yapacaksa hemen gönderilmesini istemektedir (Tonci caric, 2008). Ayrıca bunların belirli zamanlar içerisinde olmasını isteyebilmektedir. Bunların yanında, taleplerin karřılanma süresi için müşteriye söylenen zamanların tam gerçekleşmesini beklemektedir. Müşterinin büyük hacimli, çok sayıda veya ağır olabilecek talep boyutu, çevre şartlarından etkilenerek bozulabilen veya dayanıklılık özellięi gösteren paketler veya makamına, engeline, yol şartlarına göre özel araç tahsis edilmesini istedięi durumlar da ARP'nin konusudur.

ARP problemlerinde müşteri taleplerinin yanında devlet tarafından istenilen bazı kriterler veya yaptırımlar da bulunabilir. Devlet politikaları, çevre kirlilięinin azaltılması için düşük emisyon salęılayan araçların tercih edilmesini, verimli araçların kullanımının mecburi hale getirilmesini veya farklı vergilendirmeler ile teşvik edilmesini sağlayabilir. Bazı trafik kuralları ile araç kullanıcılarının sürüş süreleri kısıtlanabilir, yolun durumuna göre araç geçişine izin verilebilir veya taşıma kapasitesi düşürülebilir.

Mesafe kat eden araçların optimum şekilde kullanılmasının amaçlandığı problem türlerinde iki temel nokta ön plana çıkmaktadır. Bunlardan biri zamanla alakalı parametreler, dięeri ise mesafe ve buna baęlı maliyetlerle alakalı parametrelerdir. Zamana baęlı parametreler, servis zamanları, trafik durumları gibi birçok nedene baęlı olarak kesin bilinemeyen parametreler içerebilirler (Gendreau, Ghiani ve Guerriero, 2015).

Araç rotalama problemleri sadece fiziksel araçlar için düşünülmemelidir. Sanal bir sistemin karar verme mekanizmasında yer alabilir. Örneğin veri alışverişi yapan sistemler için veri taşıma ve toplamada yapılan optimizasyonlar uygulanmaktadır (Campbell ve Wilson, 2014).

ARP problemleri haricinde Araç Çizelgeleme Problemleri (AÇP) olarak adlandırılan bir üst problem türü daha bulunmaktadır. Burada ekip atama, nöbet listesi oluşturma gibi işlemlerin tümü birlikte yapılmaktadır. AÇP problemleri içinde uçak-otobüs ve tren gibi toplu taşıma araçlarının çizelgelenmesi problemleri geniş yer almaktadır. Bu çalışmada da yer alan veri tipi zaten çizelgelenmiş seferlerin rotalanmasını dikkate aldığı için çizelgeleme konusuna kısaca değinilmesinde fayda görülmüştür.

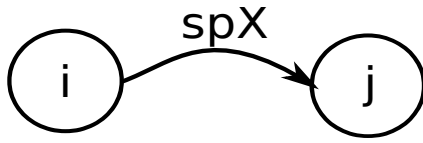
Bu çalışma; çalıştığı müddet boyunca para kazanan tek bir aracın başladığı konumuna geri dönme şartı altında, kalkış saatleri önceden belirlenmiş ve günlük olarak gerçekleştirilen seferlerin tümünü en az aylak süreyle tamamlamasını sağlayacak rotanın bulunmasını amaçlamaktadır. Böylece literatürdeki ARP problemlerinin amacında yer alan, en kısa mesafe, en az maliyet gibi amaçlardan farklı olarak çalıştığı müddet boyunca para kazanan araçların en yüksek verimlilikte kullanılmasını amaçlayan bir çözüm önerisi sunmaktadır. Çözüm önerisi bu amaca ulaşırken klasik yaklaşımın tersine bazen uzun yolun tercih edilmesini teklif edecektir. Çalışmanın kapsamı aracın çalıştıkça para kazandığı, müşterilerin taleplerinin belli, yolculuk sürelerinin kesin olduğu varsayımları ile sınırlıdır. Aracın bozulması, servis süresi, müşteri taleplerinin mevsimlere göre değiştiği, kapasitenin yetersiz oluşu, müşterinin beklediği özelliklerde araç tahsis etme veya en az kirlilik oluşturma, şoför ve ekip atama gibi konular bu çalışmanın kapsamı dışındadır. Bahsedilen durumlar mevcut çözüm önerisine eklenebilir. Fakat bu çalışmanın konusu değildir. Çözüm önerisi sezgisel olup, çözümde genetik algoritma kullanılmıştır. Doğal olarak kesin en iyi sonucu garanti etmeyecektir.

Çalışmada ilk olarak literatürdeki ARP problemlerine değinilecektir. Böylece araştırmacılar temel ARP problemleri hakkında genel olarak bilgi edineceklerdir. Diğer ARP problemlerinin temel ARP problemlerinin birleşimi veya alt kısıtlarından oluşan türleri olduğundan bahsedilecektir. Temel ARP problemlerine ait matematik modelleri

kısaca verilecektir. Bu aşamada dikkat edilmesi gereken bir nokta da; verilen matematiksel modellerin daha fazla deęişken veya az deęişkenle çözülen, yeni teklif edilen alternatifleri bulunabilir. Literatürdeki çalışmalarda yer alan probleme ait örnek veriler ve biçimleri hakkında da kısaca bilgi verilecektir. Böylece teze konu olan problemin önceki verilerden farkı kolayca görülebilecektir. Probleme ait örnek verilerin çoęu için ARP konusunun anlatıldığı ve problem veri örneklerinin yayınlandığı <http://neo.lcc.uma.es/vrp/> sitesinden alınmıştır (*Vehicle Routing Problem | NEO Research Group, 2013*). Bahsi geçen örnek problem veri biçimleri kısaca açıklanmaya çalışılmıştır. Ardından aylak zamanı en küçükleme amaçlı ARP probleminin tanımı yapılarak, çözümün önemi gösterilecek, problem genetik algoritma ile çözülecek, sonuçlar yorumlanarak tez çalışması tamamlanacaktır.

2 TEMEL KAVRAMLAR

Araştırmanın ilerleyen noktalarında bahsedilecek bir kaç terimi burada açıklamak uygun olacaktır. Gidilecek hedef için düğüm veya nokta (node, vertex) tanımı kullanılacaktır. Ağ türü problemlerde gidilecek yol, yapılan iş genellikle yay (arch) üzerinde gösterilmektedir. Bu çalışmada da sefer numaraları yapılan iş olarak yay üzerinde gösterilmektedir. Ancak düğüm üzerinde işlerin sıraları değil, gidilecek hedef noktaları gösterilmektedir. Ayrıca bu çalışmadaki probleme özel olarak her düğümün tüm diğer düğümlere gidilememektedir. Bu nedenle öncelik ilişkisi olarak adlandırılabilir kısıt bulunmaktadır. Öncelik ilişkileri oklarla gösterilmiş olup, her i noktasından j noktasına giden araç için sefer numarası olarak spX tanımlanmış olup, yay üzerinde gösterilmiştir.



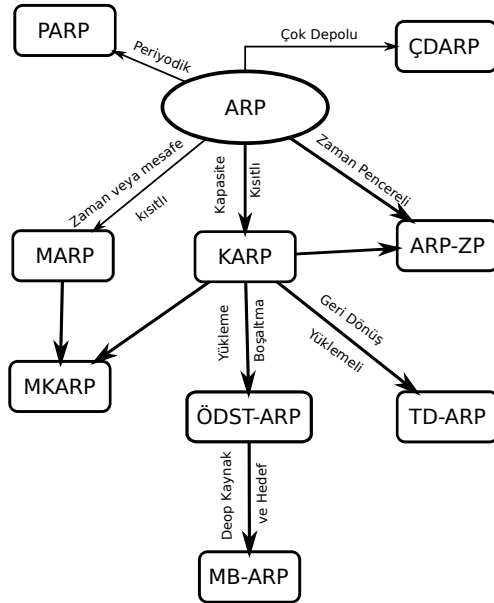
Şekil 1: Yay-Düğüm Gösterimi

3 ARAÇ ROTALAMA PROBLEMLERİ (ARP) HAKKINDA BİLGİLER

Araç Rotalama Problemleri (ARP) Gezgin Satıcı Problemlerinin (GSP) bir alt problemi olarak ele alınabilir. İlk etapta kapasite kısıtı ile farklılaşan ARP problemleri yeni kısıtlarla farklı alt sınıflara ayrılmışlardır. ARP problemlerinin çözümü GSP problemlerinin çözümünden daha zordur. Literatürde binlerce nokta içeren bir GSP probleminin kesin çözümü bulunabilirken, ARP problemleri için ancak yüz nokta ile kesin çözümünün mümkün olduğu görülmektedir (Laporte, 2009). Gelişen donanımlar ile güçlenen hesaplama kabiliyeti sonucu hesaplanabilen nokta sayısında artış görülebilir ancak aradaki zorluk farkı aynı ölçekte kalmaya devam edecektir.

Temeli çok öncelere dayanan GSP problemi ilk kez Dantzig ve arkadaşları tarafından tamsayı programlama ile formülize edildikten (Dantzig, Fulkerson ve Johnson, 1954) 5 yıl sonra, yeni kısıtlar ile ARP problemi literatürde yerini almıştır (Dantzig ve Ramser, 1959).

ARP problemlerinin günümüzde çok sayıda türü bulunmaktadır ve zamanla yeni türler de eklenmektedir. Tüm ARP problemleri temel problem kümelerinin üzerine inşa edilmiştir. (Montoya-Torres *vd.*, 2015) çalışmalarında temel problem kümelerinin etkileşimlerini kısaca Şekil 2’de gösterildiği gibi özetlemişlerdir.



Şekil 2: ARP genel kategorileri

ARP probleminin temelini oluşturan GSP probleminin çok sayıda matematiksel modeli bulunmaktadır. Bunlar içindeki en performanslı model (Matai, R., Singh, S. P., & Mittal, 2010) çalışmasından alıntılanarak kısaca gösterilmiştir.

$G = (V,E)$ düğüm ve yayın tutulduğu bir küme, n müşteri/şehir sayısı olmak üzere $V_c = \{ 1,2,\dots,n\}$ ve 0 numaralı düğüm depoyu temsil etmektedir. E , bir düğümden diğerine olan gidişi temsil etmektedir. $\{i, j\} \in E$

c_{ij} , i şehirden j şehrine olan mesafe.

x_{ij} , i şehirden j şehrine gidilen optimal tur içinde ise 1, aksi halde 0 değerini alan ikili tamsayı değişken olmak üzere

Amaç Fonksiyonu;

$$\min \sum_{i < j} c_{i,j} * x_{i,j} \quad (1)$$

Kısıtlar;

$$\sum_{i < k} x_{i,k} + \sum_{j > k} x_{k,j} = 2 \quad (k \in V) \quad (2)$$

$$\sum_{i,j \in S} x_{i,j} \leq |S| - 1 \quad (S \subset V, 3 \leq |S| \leq n - 3) \quad (3)$$

$$x_{i,j} = 0 \text{ veya } 1 \quad (i, j) \in E \quad (4)$$

3.1 KAPASİTE KISITLI ARP (KARP)

ARP, GSP problemlerinden ayrılarak farklı bir problem türü olarak tanımlanan ilk problem türüdür. Başladığı noktaya dönmek kaydı ile kapasitesi sınırlı bir aracın en az maliyetle müşterilerin taleplerini yerine getirme amacı taşıyan problemdir. Bu problem de kendi altında birden fazla sınıflara ayrılmıştır. Kapasiteleri farklı veya aynı (homojen kapasiteli veya heterojen kapasiteli) araçlarla yapılan gerçek hayatı daha iyi temsil eden türleri vardır. KARP için;

- Tüm müşteriler bir kez ziyaret edilmiş olmalıdır.
- Her araç başladığı depoya geri dönmelidir.
- Herhangi bir rotadaki toplam talep araç kapasitesini aşmamalıdır.

$d_{i,j}$, i düğümünden j düğümüne olan negatif olmayan maliyeti temsil etmektedir. Her düğüm negatif olmayan bir talebe (q_i) sahiptir. Her aracın kapasitesi Q herhangi bir talebe eşit veya daha büyük ($q_i \leq Q$) olmalıdır.

KARP için çeşitli matematiksel modeller sunulmuştur. Laporte ve arkadaşları tarafından 1985'te sunulan iki indisli farklı araç kapasiteli (heterojen) KARP probleminin matematiksel modeli aşağıda verilmiştir (Cordeau *vd.*, 2007; Baldacci *vd.*, 2011).

x_{ij} tamsayı bir değişken olup 0,1 değerleri alırken, j indeksli elemanlar için $\{0,1,2\}$ değerleri almaktadır. $x_{ij}=1$ ise ziyaret gerçekleştirilmiş, $x_{0,j}=2$ ise çözümde tek müşteri seçilmiştir. m adet araç vardır.

$\forall \{i, j\} \in E \{ \{*, j\} : j \in V_c \}$ olup değerleri $\{0,1,2\}$ 'dir .

$\forall \{0, j\}, j \in V_c$

$S \subseteq V_c \delta(S) = \{ \{i, j\} \in E : i \in S, j \notin S \text{ veya } i \notin S, j \in S \}$

$\zeta = \{ S : S \subseteq V_c, |S| \geq 2 \}$

Matematiksel modeli

Amaç Fonksiyonu;

$$\min \sum_{\{i,j\} \in E} d_{i,j} x_{i,j} \quad (5)$$

Kısıtları;

$$\sum_{\{i,j\} \in \delta(h)} x_{i,j} = 2 \quad (\forall h \in V_c) \quad (6)$$

$$\sum_{\{i,j\} \in \delta(S)} x_{i,j} \geq 2 \lceil q(S)/Q \rceil \quad (\forall S \in \zeta) \quad (7)$$

$$\sum_{j \in V_c} x_{0,j} = 2m \quad (8)$$

$$x_{i,j} \in \{0,1\} (\forall \{i,j\} \in E \{ \{0,j\} : j \in V_c \}) \quad (9)$$

$$x_{0,j} \in \{0,1,2\} (\forall \{0,j\}, j \in V_c) \quad (10)$$

Probleme ait örnek veri biçimi

Tablo 1’de <http://neo.lcc.uma.es/vrp/vrp-instances/capacitated-vrp-instances/> sitesinden alınan KARP için kullanılan örnek veri verilmektedir. Örnek veri 37 noktadan ve 6 araçtan oluşmaktadır. Veri biçiminin anlaşılması için verinin bir kısmı verilmiştir. Tablonun sağ tarafında örnek çözüm grafiği gösterilmiştir. Problemin ilk kısmında her noktanın koordinatları, ikinci kısmında ise her noktanın talepleri belirtilmiştir.

Tablo 1: KARP için örnek veri biçimi ve çözüm grafiği

<pre>NAME : A-n37-k6 COMMENT : (Augerat et al, Min no of trucks: 6, Optimal value: 949) TYPE : C ARP DIMENSION : 37 EDGE_WEIGHT_TYPE : EUC_2D CAPACITY : 100 NODE_COORD_SECTION 1 86 22 2 29 17 3 4 50 DEMAND_SECTION 1 0 2 1 3 23 4 23 DEPOT_SECTION 1 -1</pre>	
--	--

Augerat’ın veri biçimi olarak geçen bu tür haricinde, Breedam, Fisher gibi diğer veri biçimleri de bulunmaktadır (*Vehicle Routing Problem | NEO Research Group*, 2013). Breedam biçimi için daha ayrıntılı bilgi edinmek için (Reinelt, tarih yok) dökümanına başvurulabilir.

3.2 ZAMAN PENCERELİ ARP (ARP-ZP)

Klasik ARP probleminin daha geliştirilmiş ve günlük hayat kısıtlarını daha fazla karşılayan bir problem türüdür. Müşterinin müsait olduğu bir zaman aralığı [a_i (müşterinin müsait olduğu ilk zaman), b_i (müşterinin müsait olduğu son zaman)] vardır. Bu süreden (a_i) önce gidildiğinde beklenmek zorundadır. Sürenin (b_i) geçilmesi mümkün sonuç olarak kabul edilmemektedir. Aynı süreler depo için de verilebilmektedir. $a_{depo} = 0$ ve $b_{depo} = \infty$ olduğu durum klasik KARP problemidir.

Yukarıda geçen klasik tanımın üzerine, $|V| = n+2$ 'dir. Çünkü başlangıç olan depo 0 indisindedir ve son depoya dönüş de $n+1$ nolu indistedir. t_{ij} 'den j 'ye giderken geçen süre olmak üzere, probleme özgü şu kısıtlar yer alacaktır (Cordeau *vd.*, 2007);

$$a_0 = \min_{i \in N} \{a_i - t_{0,i}\}, b_0 = \max_{i \in N} \{b_i - t_{0,i}\}$$

$$a_{n+1} = \min_{i \in N} \{a_i + s_i + t_{i,n+1}\}, b_{n+1} = \min_{i \in N} \{b_i + s_i + t_{i,n+1}\}$$

N tüm noktaların olduğu $1, \dots, n$ kümesi ve s_i i noktasındaki servis süresidir. K araçların bulunduğu $1, \dots, k$ araç kümesidir.

$\delta^+(i) = \{j : (i, j) \in E\}$ ve $\delta^-(j) = \{i : (i, j) \in E\}$. Problemin depo zaman penceresi aralığı $[a_0, b_0]$ ve $[a_{n+1}, b_{n+1}]$ olacaktır.

Matematiksel modeli

Amaç Fonksiyonu

$$\min \sum_{k \in K} \sum_{(i,j) \in E} c_{i,j} x_{i,j}^k \quad (11)$$

Kısıtları

$$\sum_{k \in K} \sum_{j \in \delta^+(i)} x_{i,j}^k = 1, i \in N \quad (12)$$

$$\sum_{j \in \delta^+(0)} x_{0,j}^k = 1, k \in K \quad (13)$$

$$\sum_{i \in \delta^-(j)} x_{i,j}^k - \sum_{i \in \delta^+(j)} x_{j,i}^k = 0, k \in K, j \in N \quad (14)$$

$$\sum_{i \in \delta^-(n+1)} x_{i,n+1}^k = 1, k \in K \quad (15)$$

$$x_{i,j}^k (\omega_i^k + s_i + t_{i,j} - \omega_j^k) \leq 0, k \in K, (i,j) \in E \quad (16)$$

$$a_i \leq \omega_i^k \leq b_i, k \in K, i \in V \quad (17)$$

$$\sum_{i \in N} q_i \sum_{j \in \delta^+(i)} x_{i,j}^k \leq Q, k \in K \quad (18)$$

$$x_{i,j}^k \in \{0,1\}, k \in K, (i,j) \in E \quad (19)$$

Probleme ait örnek veri biçimi

Breedam'ın ürettiği T1 problemleri, 100 nokta, 100 kapasiteli, 1 depolu ve sonsuz araçlı bir problem kümesidir. Verinin bir kısmı aşağıdaki gibidir. Bu biçim başka problem türleri için de kullanıldığı için fazla sütunları bulunmaktadır. ARP-ZP için, 0 olarak bırakılmıştır. İlk satır deponun olduğu satırdır. İlk sütun düğümleri, ikinci ve

üçüncü sütunlar x ve y koordinatlarını, 4,5,6 ve 7. sütunlar başlama ve bitiş sürelerini gösteren sırasıyla birinci ve ikinci zaman penceresi parametreleridir. 8. sütun talebi, 9. sütun servis zamanını ve 10.sütun da ilgili noktaya eşya bırakıldığını veya alındığını göstermektedir. (Yalın zaman pencereli problem için 9 ve 10. sütunlar 0 verilmiştir. Yani servis süresi ve bırakma-alma dikkate alınmamıştır.)

0	50	50	0	960	960	960	0	0	0
1	5	5	0	60	60	60	10	0	0
2	15	5	0	60	60	60	10	0	0
3	25	5	0	60	60	60	10	0	0
4	35	5	0	60	60	60	10	0	0
5	45	5	0	60	60	60	10	0	0

.....

Cordeau ve Solomon'un soru biçimi ve çözüm önerisi biçimi ek EK - 1 ve EK - 2'de verilmiştir.

3.3 ÇOK DEPOLU ARP (ÇDARP)

Kargo dağıtımlarında birden fazla noktadan müşterilere ulaşmak söz konusu olmaktadır. Bu durumda klasik tek depolu ARP problemi yetersiz kalmaktadır. Problemi ayrı ayrı tek bir KARP problemi şeklinde çözmek de mümkündür. Fakat depolar dağınık ise optimum çözümün bulunması mümkün olmayacaktır. Ayrıca araçların homojen veya heterojen kapasitelerine göre çözümü farklı olacaktır. Zaman pencereli ve ek kısıtlarla alt problemleri de vardır.

Matematiksel modeli

Önceki tanımlamaları kullanarak aşağıdaki matematiksel modeli yazmak mümkündür (Montoya-Torres *vd.*, 2015). Bu heterojen kapasiteli problemde klasik probleme ek olarak V_d kümesi eklenmiştir. Bu küme depoların tutulduğu bir listedir. M adet depo vardır. P_k her aracın kapasiteleridir.

Amaç Fonksiyonu

$$\min \sum_{i=1}^{N+M} \sum_{j=1}^{N+M} \sum_{k=1}^K (c_{i,j} x_{i,j,k}) \quad (20)$$

Kısıtları

$$\sum_{i=1}^{N+M} \sum_{k=1}^K x_{i,j,k} = 1 \quad j=1, \dots, N \quad (21)$$

$$\sum_{j=1}^{N+M} \sum_{k=1}^K x_{i,j,k} = 1 \quad i=1, \dots, N \quad (22)$$

$$\sum_{i=1}^{N+M} x_{i,h,k} - \sum_{j=1}^{N+M} x_{i,j,k} = 0, \quad k=1, \dots, K, h=1, \dots, N+M \quad (23)$$

$$\sum_{i=1}^{N+M} q_i - \sum_{j=1}^{N+M} x_{i,j,k} \leq P_k, \quad k=1, \dots, K \quad (24)$$

$$\sum_{i=1}^{N+M} \sum_{j=1}^{N+M} c_{i,j} x_{i,j,k} \leq T_k, \quad k=1, \dots, K \quad (25)$$

$$\sum_{i=N+1}^{N+M} \sum_{j=1}^N x_{i,j,k} \leq 1, \quad k=1, \dots, K \quad (26)$$

$$\sum_{j=N+1}^{N+M} \sum_{i=1}^N x_{i,j,k} \leq 1, \quad k=1, \dots, K \quad (27)$$

$$y_i - y_j + (M+N)x_{i,j,k} \leq N+M-1, \quad 1 \leq i \neq j \leq N \text{ ve } 1 \leq k \leq K \quad (28)$$

$$x_{i,j,k} \in \{0,1\} \quad \forall i, j, k \quad (29)$$

Kısıt, (21) ve (22) her müşterinin bir defa ve bir araçla hizmet gördüğünü, 23 rotanın devamlılığını, 24 ve, 25 toplam rota maliyeti ve araç kapasitesi kısıtlarını sağlamaktadır. (26) aracın müsait olduğunu, (27) alt rota oluşumunu engellemektedir. (28) araç kapasitesinin aşılmasını engellemektedir (Montoya-Torres vd., 2015).

Probleme ait örnek veri biçimi

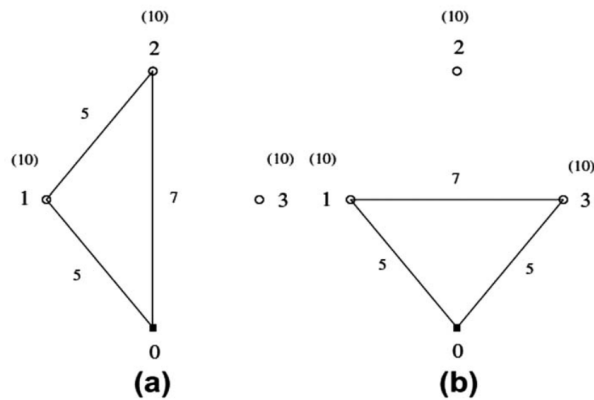
Cordeau'nun hazırladığı biçimde (açıklama için EK - 1'a bakınız.) ÇDARP sorusu aşağıdaki gibidir. Soru tipinin 2, yani ÇDARP olduğunu, 4 araç ve 50 müşteri bulunduğunu, işlemin 4 günlük periyotta yapılacağını söylemektedir. Diğer dört satır her araç için 80 birim yük kapasitesi ve maksimum rotasının sonsuz (0) olduğunu söylemektedir.

2 4 50 4
0 80
0 80
0 80
0 80
1 37 52 0 7 1 4 1 2 4 8
2 49 49 0 30 1 4 1 2 4 8
3 52 64 0 16 1 4 1 2 4 8
4 20 26 0 9 1 4 1 2 4 8
5 40 30 0 21 1 4 1 2 4 8
.....

3.4 PERİYODİK ARP (P ARP)

Bazı bölgeler için araçların birkaç günde bir uğraması yeterli olmaktadır. Bu durumda stoğu hızlı biten yer için sık uğranılacağı, diğer yerler için az uğranılacağı bir plan yapılmalıdır. Veri ulaştırma ve toplama işlemlerinde de kullanılmaktadır. İlk kez belediye çöp toplama araçlarının rotalanması problemiyle tanıtılmıştır (Campbell ve Wilson, 2014). Örneğin iki günlük periyot ve üç nokta için, 1 noktasına her gün uğranılması gerekiyorsa iki günlük rota Şekil 3'deki gibi olmaktadır.

Periyodik ARP problemleri için kapasite kısıtı geçerli olup, zaman pencereli olarak da alt kısıtlarla farklı türleri bulunmaktadır.



Şekil 3: İki günlük periyotla rotalanmış 3 müşterili problem

Matematiksel modeli

Önceki tanım ve kısıtlar haricinde p günlük bir planlama söz konusudur. f_i servis sıklığını (uğrama tekrarı) belirtirken, C_i , izin verilen f_i lerin kümesidir. Talep edilen q_i her uğranıldığında müşteriye verilmelidir. $V = V^1 \cup V^2$ olup V^1 uğrama sıklığı bir olanların V^2 ise uğrama sıklığı en az ikiye eşit olanların listesidir. Uğranılan günler matrisi $[a_{ks}]$, p gün kadar satır içerir ve (0-1) şeklinde kolondan oluşur. $a_{ks} = 1$ sadece gün kombinasyonu s için k gününde mümkünse olur (Baldacci vd., 2011).

$V_k = \{i \in V : \sum_{s \in C_i} a_{k,s} \geq 1\}$ R_k, V_k müşterisini k gününde $k \in P$ ziyaret ettiğini gösteren küme olsun. $R_i^k \subseteq R^k$ V_k nin elemanı olan i müşterisinin rotalarını göstermektedir. R_i^k ve c_i^k rotanın k günündeki maliyetini göstermektedir. $l \in R^k, k \in P$ Y_{is} 0-1 ikili değerlerinde olup sadece $i \in V^2$ müşterisine atanmış gün kombinasyonu içinde $s \in C_i$ ise 1'dir. x_i^k ikili değişkeni sadece $l \in R^k$ durumu $k \in P$ çözümde ise 1'dir.

Amaç Fonksiyonu

$$\min \sum_{k \in P} \sum_{l \in R^k} c_l^k x_l^k \quad (30)$$

Kısıtları

$$\sum_{k \in P} \sum_{l \in R_i^k} x_l^k = f_i (\forall i \in V) \quad (31)$$

$$\sum_{l \in R_i^k} x_l^k - \sum_{s \in C_i} a_{k,s} y_{i,s} = 0 (\forall i \in V^2, \forall k \in P) \quad (32)$$

$$\sum_{l \in R^k} x_l^k \leq m_k (\forall k \in P) \quad (33)$$

$$x_l^k \in \{0,1\} (\forall l \in R^k, \forall k \in P) \quad (34)$$

$$y_{i,s} \in \{0,1\} (\forall s \in C_i, \forall i \in V^2) \quad (35)$$

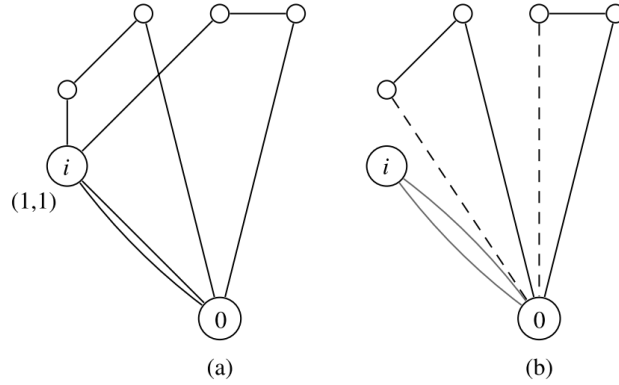
Probleme ait örnek veri biçimi

Cordeau biçiminde örnek bir veri;

```
1 3 51 2
0 160
0 160
0 30 40 0 0 0 0
1 37 52 0 7 1 2 1 2
2 49 49 0 30 1 2 1 2
3 52 64 0 16 1 2 1 2
4 20 26 0 9 1 2 1 2
```

3.5 BÖLÜNEBİLİR TAŞIMALI ARP (BTARP)

Klasik kapasite kısıtlı ARP probleminde, müşteri talebi araç kapasitesinden daha fazla olamazdı. Bu problem türünde araç müşterinin olduğu yere birden fazla kez gelerek talebini karşılayabilmektedir. Bu problem de NP-zor'dur. Zaman pencereli, topla dağıt problemleri gibi diğer ARP türleri ile karışık olarak kullanılmaktadır. Şekil 4'te grafik üzerinden anlaşılacağı üzere i noktasındaki talebe araç kapasitesi dolduğu için iki kez gidilmesi gerekmektedir (Archetti, Savelsbergh ve Speranza, 2006).



Şekil 4: Örnek bölünebilir taşımali ARP için araç rotaları

Matematiksel modeli

Burada önceki klasik problemin farkı konulup, diğer parametreler yazılmayacaktır. Daha fazla bilgi için (Dror, Laporte ve Trudeau, 1994) çalışmasına başvurulabilir.

Amaç Fonksiyonu

Klasik formül (11)'in aynısıdır.

Kısıtları

$q_i \leq Q_k$ şartı kalktığı için, aşağıdaki kısıtlar eklenmiştir.

$$\sum_{i=1}^N q_i y_{i,k} \leq Q_k, \quad k = \{1, \dots, K\} \quad (36)$$

$$\sum_{j=0}^N x_{i,j,k} \geq y_{i,k}, \quad i = \{1, \dots, N\}, k = \{1, \dots, K\} \quad (37)$$

Kısıt (36) araç kapasitesinin aşılmayacağını gösterirken, (37) ise müşterinin taleplerinin karşılanması için konulmuştur.

Bir noktaya defalarca araç uğramasının problemin optimum çözümü yanlış olacağı için, iki kısıta daha ihtiyaç vardır. Öncelikle araç sayısının kaç olması gerektiği de bir başka önceden çözülmesi gereken problemdir. Bunun için kutulama problemi olarak çözülmesi gerekmektedir. $V(S)$ en küçük az araç sayısı olmak üzere;

$$\sum_{i \in S} \sum_{j \in S} x_{i,j} \geq 1, \quad ((S \subseteq V) \setminus \{0\}; 2 \leq |S| \leq n-1) \quad (38)$$

$$\sum_{k=1}^K \sum_{j \in S} x_{i,j,k} \leq |S| - V(S), \quad ((S \subseteq V) \setminus \{0\}; 2 \leq |S|) \quad (39)$$

Probleme ait örnek veri biçimi

Klasik problemdeki talebin bölünebilir olduğu yaklaşımı ile mevcut sorular çözülebilir.

3.6 TOPLA DAĞIT ARP (TD-ARP)

Klasik problemde her zaman depodan müşteriye bir mal ulaştırma üzerine yoğunlaşmıştır. Ancak birçok sebepten dolayı (müşterinin bir malı iade etmesi gibi, veya kargocuya gönderilecek bir ürün vermesi gibi) müşteriden de mal toplanması gerekmektedir. Araç kapasitesinin sınırlı olması nedeniyle aracın dolduğu durumda depoya dönmesi gerektiği veya müşteriye birkaç kez uğraması gibi durumlar oluşmaktadır. Araç tur süresinin bu durumdan kötü etkilenmesi kaçınılmazdır. Bu problemin çözülebilmesi için, her müşterinin sadece bir kez ziyaret edilmesi kısıtı kaldırılabilir. Veya malların müşterilerden toplanması, tüm dağıtım yapıldıktan sonra da gerçekleşebilir.

Zamanımızda internet alışverişinin de çok artması nedeniyle daha fazla oluşan bu durum önemini arttırmaktadır. Bunun önemini gösteren bir bilgi olarak, Amerika'daki bir şirket her gün, yaklaşık 8 milyon müşteriye uğrayarak, 15,6 milyon paket almaktadır (García-Nájera, Bullinaria ve Gutiérrez-Andrade, 2015). Önceki modellerin de kısıtlarının ilavesi ve yeni kısıtların oluşması ile farklı alt problemler oluşmuştur. Araçların kapasitelerinin farklı olması (Barış Keçeci, Fulya Altıparmak, 2015), dağıtımın ve toplanan malların bölünebilmesi, senkronize hareket edilmesi (Şahin *vd.*, 2013; Bayrak ve Özyörük, 2017) ve zaman pencereli kısıtlı (Jing-Quan Li, 2006) problemler TD-ARP'nin bazı türleridir.

Problemin amaç fonksiyonu en az maliyet olan klasik amaç fonksiyonudur. Kısıtları ise klasik kısıtların üzerine, ilgili noktadaki aracın toplayan ve dağıtan araç olduğu, her aracın depoya bir kez uğradığı kısıtları vardır (Şahin *vd.*, 2013). Problemin ek özelliklerine göre kısıtlar artmaktadır.

Probleme ait örnek veri biçimi

Önceki Breedam'ın biçiminde son sütun, toplama veya dağıtım yaptığını göstermektedir. Son sütun 1 ise toplama yapıyordur.

0	50	50	0	9999	9999	9999	0	0	0
1	5	5	0	9999	9999	9999	10	0	1
2	15	5	0	9999	9999	9999	10	0	0
3	25	5	0	9999	9999	9999	10	0	1

.....

3.7 ÖNCE DAĞIT SONRA TOPLA ARP (ÖDST-ARP)

Bu problem klasik Topla-Dağıt araç rotalama probleminin aynısıdır. Fakat daha gerçekçi bir senaryo ile tüm toplamaların dağıtımdan sonra yapılacağı planlamasını yapar. Zaman pencere, karışık kapasiteli gibi gerçek hayata yakın kısıtları eklenen türleri vardır (Berghida ve Boukra, 2015). Eş zamanlı veya eş zamansız araç rotalamalarının bulunduğu problem çözümleri başka ek kısıtlar içermektedir (Toth ve Vigo, 1999). Diğer pek çok ARP problemleri için çözüm önerilerinde olduğu gibi, sezgisel yaklaşımlar yaygın olarak kullanılmaktadır (García-Nájera, Bullinaria ve Gutiérrez-Andrade, 2015).

Probleme ait örnek veri biçimi

Problemin örnek sorusu, Topla-Dağıt problem türlerindeki gibidir.

3.8 DİĞER ARP PROBLEMLERİ

ARP'nin buraya kadar bahsedilen temel kategoriler haricinde çok sayıda temel türleri bulunmaktadır. Bu yapılan çalışma da yeni bir ARP türü ve çözümünü sunmaktadır. Bu türler yeni amaç fonksiyonları veya kısıtları nedeniyle klasik türden ayrılmaktadır. Çok kullanılan bazı türlerden kısaca bahsedilecektir.

Açık ARP, dağıtıcının bir daha depoya dönmediği problem türüdür. Özellikle fason iş yaptıran işletmeler için geçerli bir senaryodur.

MB-ARP (mevkiye bağlı [site dependent]) (Baldacci, Toth ve Vigo, 2010) ilgili noktaya özel gönderilecek araçları dikkate alır. Kişinin taleplerini karşılayacak bir araç gönderimini dikkate alır. Daha fazla kapasite anlamı taşıdığı gibi, ek özelliklerin (engelli, VIP vb.) de olduğu araçlar olabilir.

Problemdaki tüm parametrelerin önceden bilinen ve kesin olmadığı gerçek hayat durumları vardır. Bu durumda stokastik ARP olarak adlandırabileceğimiz, müşteri sayısı, talep ve zamanlarının olasılıklı olduğu problem türüdür. Olasılıklı olacak ihtimaller araçların bozulması, değişen sefer süreleri ve tamir süreleri gibi gerçek hayat kısıtları ile daha da arttırılabilir. (Daneshzand, 2011).

Stokastik ARP için önceden toplanan verilerle oluşturulmuş ve belirli dağılımlara sahip olasılıklar söz konusudur. Bu şekilde veri olmadığı durumlar için Bulanık ARP kullanılmaktadır.

Küm-ARP ve alt problemleri olan müşteriye en az gecikme (zaman penceresinden hariç olarak, müşteriye ulaşana kadar geçen süre) ile ulaşma problemleri vardır. Bu yaklaşım kargo dağıtıcısı ve tamirci problemi olarak da bilinmektedir (Tonci caric, 2008).

Kirliliğe bağlı rotalama (pollution-routing problem) aracın yüküne bağlı olarak, hızı ve mesafesine bağlı en az kirlilik oluşturması amacına uygun rotalamadır (Laporte, 2013). Benzer şekilde daha az yakıt amacı ile yapılan yeşil araç rotalama (Green-ARP) literatürde yer alan bir başka ARP türüdür (Lin *vd.*, 2014) (Koç ve Karaoglan, 2016) (Kuo ve Wang, 2011).

Son olarak (Eksioglu, Vural ve Reisman, 2009)'nun yaptıkları ARP problemlerinin sınıflandırma çalışmasının çevirisi Tablo 2'de verilerek konuya başlıklar altında bakma imkanı verilmiştir. Bu çalışma ile ARP çalışmalarının içerik, kısıt ve türlerine göre nelerden oluştuğu çalışılmıştır. Bir başka araştırma için bakınız: (Braekers, Ramaekers ve Van Nieuwenhuyse, 2015).

Tablo 2: ARP sınıflandırma çalışması çıktısı.

1	Çalışmanın Tipi
1.1	Teori
1.2	Uygulanan Yöntemleri
1.2.1	Kesin Yöntemler
1.2.2	Sezgiseller
1.2.3	Benzetim
1.2.4	Gerçek zamanlı çözüm yöntemleri
1.3	Uyarılma uygulamaları
1.4	Literatür taraması, inceleme veya yarı araştırmalar.
2	Senaryo Özellikleri
2.1	Rota üzerindeki nokta sayısı
2.1.1	Bilinen/Rastsal olmayan (deterministik)
2.1.2	Bir kısmı bilinen, kalanı olasılıklı
2.2	Yük bölünebilmesi kısıtı
2.2.1	Bölünebilmeye izin verilen
2.2.2	Bölünebilmeye izin verilmeyen
2.3	Müşteri memnuniyeti, talep kalitesi
2.3.1	Bilinen
2.3.2	Sezgisel
2.3.3	Bilinmeyen

- 2.4 Yeni müşterilerin talep sayıları
 - 2.4.1 Bilinen
 - 2.4.2 Sezgisel
 - 2.4.3 Bilinmeyen
- 2.5 Müşteride bekleme ve servis süresi
 - 2.5.1 Bilinen
 - 2.5.2 Zaman bağımlı
 - 2.5.3 Araç tipine bağlı
 - 2.5.4 Sezgisel
 - 2.5.5 Bilinmeyen
- 2.6 Zaman pencereyi yapı
 - 2.6.1 Gevşek zaman pencereyi
 - 2.6.2 Sıkı zaman pencereyi
 - 2.6.3 İkisinin karışımı
- 2.7 Zaman ufku
 - 2.7.1 Tek periyot
 - 2.7.2 Çok periyot
- 2.8 Ana Taşıyıcılar
 - 2.8.1 Düğümlerde hem toplama hem de dağıtımın eş zamanlı olduğu
 - 2.8.2 İşlemin sadece toplama veya dağıtım olarak yapıldığı, eş zamanlı olmadığı tip
- 2.9 Yay/Düğüm örtme kısıtları
 - 2.9.1 Öncelik ve eş kısıtlar
 - 2.9.2 Alt küme örtme kısıtları
 - 2.9.3 Yeniden uğrama izni
- 3 Problemin Fiziksel Özellikleri
 - 3.1 Taşımacılık ağı tasarımı
 - 3.1.1 Doğrudan ağ
 - 3.1.2 Dolaylı ağ
 - 3.2 Müşteri adresleri konumları
 - 3.2.1 Müşterilerin düğümde olduğu
 - 3.2.2 Yay rotalama örnekleri
 - 3.3 Coğrafik müşteri konumları
 - 3.3.1 Kentsel (belli bir dokuya göre dağılmış.)
 - 3.3.2 Kırsal (rastgele dağılmış)
 - 3.3.3 İkisinin karışımı
 - 3.4 Merkez nokta sayısı
 - 3.4.1 Tek merkezli
 - 3.4.2 Çok merkezli
 - 3.5 Yükleme/Boşaltma noktası sayısı
 - 3.5.1 Tek depolu
 - 3.5.2 Çok depolu
 - 3.6 Zaman penceresi tipi
 - 3.6.1 Müşterilerde bulunan kısıtlar
 - 3.6.2 Yollarda bulunan kısıtlar
 - 3.6.3 Depo veya dağıtım noktasındaki kısıtlar
 - 3.6.4 Araç veya sürücüye bağlı kısıtlar
 - 3.7 Araç sayısı
 - 3.7.1 Tam araç sayısı ile N adet (TSP ye benzer)
 - 3.7.2 N adet araca kadar
 - 3.7.3 Sınırsız sayıda araç

- 3.8 Kapasite kısıtları
 - 3.8.1 Kapasite kısıtlı
 - 3.8.2 Sınırsız kapasiteli
- 3.9 Araçların homojenliği (Kapasite)
 - 3.9.1 Benzer araçlar
 - 3.9.2 Yüke uygun araçlar. (Çöp, harç kamyonu vb.)
 - 3.9.3 Karışık araçlar
 - 3.9.4 Müşteriye uygun araçlar (VIP, engelli araç vb.)
- 3.10 Sefer süresi
 - 3.10.1 Bilinen
 - 3.10.2 Fonksiyona bağlı
 - 3.10.3 Sezgisel
 - 3.10.4 Bilinmeyen
- 3.11 Taşımacılık maliyeti
 - 3.11.1 Zamana bağlı
 - 3.11.2 Mesafeye bağlı
 - 3.11.3 Araca bağlı
 - 3.11.4 İşleme bağlı
 - 3.11.5 Gecikmeye bağlı bir fonksiyon
 - 3.11.6 Riskli veya zararlı/zehirli duruma bağlı
- 4 Bilgi Özellikleri
 - 4.1 Bilginin evrimi
 - 4.1.1 Statik
 - 4.1.2 Kısmen dinamik
 - 4.2 Bilginin kalitesi
 - 4.2.1 Bilinen
 - 4.2.2 Sezgisel
 - 4.2.3 Tahmini
 - 4.2.4 Bilinmeyen (Gerçek zamanlı)
 - 4.3 Bilginin bulunabilirliği
 - 4.3.1 Yerel
 - 4.3.2 Küresel
 - 4.4 Bilginin işlenmesi
 - 4.4.1 Merkezi
 - 4.4.2 Merkezi olmayan
- 5 Veri Özellikleri
 - 5.1 Kullanılan veri
 - 5.1.1 Gerçek dünya verileri
 - 5.1.2 Sentetik veriler
 - 5.1.3 Hem gerçek hem sentetik veriler
 - 5.2 Veri kullanılmadan

4 ARAÇ ÇİZELGELEME PROBLEMLERİ (AÇP)

Yapılan bu çalışma bir rotalama problemi olmasına rağmen çizelgelenmiş seferlerin üzerine rotalama yaptığı için çizelgeleme problemi gibi düşünülebilir. Bu nedenle mevcut çizelgeleme problemlerine kısaca göz atmak yerinde olacaktır.

Araç çizelgeleme, hız problemleri, ekip atama, çağrı ile araç ayarlama problemleri, çevre kirliliğini azaltma, resmiyete uygun çalışma sürelerini dikkate alma gibi birçok konuyu içine alır (Laporte, 2013). Bu kısıtlar dikkate alındıktan önce veya sonra rotalama yapılır ve bu nedenle araç rotalamayı içinde barındırdığı söylenebilir.

Çağrı ile ayarlanan araç problemlerinde (dial-a-ride DARP) kalkış ve varışı belli kapasitesi sınırlı bir aracın müşteriyi aldığı noktadan gideceği noktaya götürüp, oradaki işlemlerinin süresine göre bekleyip, geri kalkış noktasına götürdüğü problemlerdir. Örneğin; araç süremeyen bir kişinin hastaneye götürülüp, tedavi gördükten sonra yeniden eve bırakılması işlemidir. Böyle durumlar için otonom araçlar ile çizelgelemenin farklı bir noktaya gidebileceğini (Pimenta *vd.*, 2017) araştırmacılar göstermiştir.

Günlük çalışma süresini aşan, resmi kısıtlar barındıran sürüşler için de çizelgeleme önemli bir işlemdir. Okul servislerinin, birden fazla okul için servis işlemi yapması nedeniyle çizelgelenmesi (Kim, Kim ve Park, 2012), araç ve sürücünün birlikte çizelgelenmesi (Stojković ve Soumis, 2001; Domínguez-Martín, Rodríguez-Martín ve Salazar-González, 2017, 2018) önemli çizelgeleme problemlerindedir.

Araç çizelgelemede en çok konu olan problemlerden biri de toplu taşıma sistemleri ile ilgilidir. Bu noktada otobüs, tren ve uçak çizelgeleme büyük önem taşımaktadır. Özellikle belediyeçilik işlemlerinde duraklardan müşterilerin alınması ve buna göre otobüs süreleri ve büyüklüklerinin ayarlanması müşteri memnuniyeti açısından önemlidir (Hassold ve Ceder, 2014; Bie, Gong ve Liu, 2015). Günün toplam talebine göre oluşturulan çizelgeleme, işe gidiş ve dönüş saatleri açısından eksik kalması nedeniyle ve bazı günlere mahsus olarak talebin arttığı zamanlar için de gerçek zamanlı çizelgeleme yöntemleri teklif edilmiştir (Wagale *vd.*, 2013). Veya daha kalabalık zamanlar için araç uğrama sıklığının artırıldığı çözüm önerileri yapılmıştır (Yue *vd.*, 2017). Müşteri talebinin kesin olmadığı ve ekibin dengeli dağıtıldığı durumlar

da dikkate alınmıştır (Cadarso ve de Celis, 2017). Bunlara ek olarak, özellikle uçaklar için inecekleri veya kalkacakları hava alanlarının yoğunluklarına göre de çizelgeleme yapılması gerektiği önerilmiştir (Pita, Barnhart ve Antunes, 2013).

Şehir içi otobüslerin ayarlanması da çok önemlidir. Başlangıç ve bitiş noktaları arasında duraklar bulunup, her durağa belli aralıklarla uğranılması gerekmektedir. Araçlara atanan sürücüler için dinlenme zamanı da verilmesi gerekmektedir. Bu durumda kaç adet sürücü ve araç gerektiği bir çizelgeleme konusudur (Shui *vd.*, 2015).

Araç kalkış sürelerinin müşteri memnuniyetini çok etkilemeden biraz esnetilmesinin çizelgelemeye pozitif katkısının olduğu gösterilmiştir (Schmid ve Ehmke, 2015).

Tren çizelgeleme problemleri de otobüs çizelgeleme problemleri gibidir. Ancak tek ray bulunmaktadır. Ayrıca başlangıcına yeniden döndüğü bir ring oluşturabilir. Müşteri talebine göre bu hat üzerinde kaç adet tren kullanılmalıdır ve trenler hangi duraklar arasında git gel yapmalıdır sorularına cevap verilmesi optimizasyon için iyi bir geliştirmedir (Rahimi Mazrae Shahi, Fallah Mehdipour ve Amiri, 2016; Wang *vd.*, 2017).

Çizelgeleme konusunu tamamlamadan önce çizelgelemenin çok geniş bir kapsamı olduğu unutulmamalıdır. Üretimde makine çizelgelemesi, üretim hattı çizelgelemesi gibi konular, araç çizelgelemeye de uygulanabilir. Bunu düşünen (Beck, Prosser ve Selensky, 2003) adlı araştırmacılar, ARP probleminin en temelinde bir Atölye Tipi Çizelgeleme Problemi (ATP) olduğu ve çizelgeleme ile de çözüleceğini göstermişlerdir. Alt kısıtları, AÇP problemini daha farklı kılabilir fakat aynı kısıtlar ATP için de eklenebilir. Bu noktada adı geçen araştırmacılar yaptıkları çalışmada araç rotalama ön planda ise ARP çözümlerinin daha iyi olduğu, müşteri açısından az bekleme gibi (minimize makespan) amaçlar olduğunda çizelgelemenin daha iyi olduğu sonucuna ulaşmışlardır.

5 ARP ÇÖZÜM YÖNTEMLERİ

Rotalama ve çizelgeleme problemlerinin tüm çeşitleri için araştırmacılar çeşitli çözüm yöntemleri önermişlerdir. Bu çözüm yöntemleri, kesin yöntemler, sezgisel, meta-sezgisel, gerçek zamanlı çözüm yöntemleri ve simülasyon şeklinde olmuştur (Braekers, Ramaekers ve Van Nieuwenhuyse, 2015). ARP problemlerinin NP-Zor olması nedeniyle bu çözümler büyük problemler için sezgisel veya meta-sezgisel yöntemler tercih edilmektedir. Kesin yöntemler için matematiksel model çözümü haricinde dinamik programlama, gevşetilmiş lagrange yöntemi, kolon oluşturma gibi yöntemler de kullanılmaktadır (El-Sherbeny, 2010).

Büyük problem kümeleri için sezgisel algoritmalar olmazsa olmaz çözüm yöntemidir. Araç rotalama problemlerinde karınca koloni, tabu arama, tavlama benzetimi, genetik algoritma (Baker ve Ayechev, 2003), memetik algoritma, parçacık sürü optimizasyonu, yapay arı kolonisi yaygın kullanılan sezgisel algoritmalar (Braekers, Ramaekers ve Van Nieuwenhuyse, 2015). Bunlar haricinde araştırmacıların denedikleri bulanık mantık, yapay sinir ağları, değişken komşu arama, kral kelebeği algoritması, modüler sezgisel algoritma gibi çok sayıda sezgisel algoritmalar kullanılmaktadır (Hemmelmayr, Doerner ve Hartl, 2009; Daneshzand, 2011; Zhang *vd.*, 2013; Dai ve Zheng, 2015; Rahimi-Vahed *vd.*, 2015; Chen, Chen ve Gao, 2017).

Bu aşamada, çalışmada kullanılacak sezgisel yöntemlerden biri olan genetik algoritma ile ilgili kısaca bilgi verilecektir.

5.1 GENETİK ALGORİTMA

ARP problemleri için genetik algoritma çok kullanılmaktadır (Potvin, 2009; Adamidis, Voliotis ve Pliatsika, 2012). Bu başlık altında kısa bir özet şeklinde genetik algoritmanın adımları anlatılacaktır.

1859 yılında Darwin tarafından ortaya atılan doğal seleksiyon teorisine göre, hayatta kalmak için en iyi olmak gereklidir. Böylece doğaya uygun olmayan bireyler yok olurken, daha iyi bireyler nüfus içinde artmaktadır. Bu teoriyi John Holland 1975 yılında genetik algoritma olarak modellemiştir. Bunun haricinde genetik programlama,

evrimsel programlama ve evrim stratejileri şeklinde farklı 3 model daha vardır. Fakat en yaygını genetik algoritmadır (Sivanandam ve Deepa, 2008).

Genetik algoritmanın avantajları,

- Paralel çözümler üretebilmesi,
- Çözüm alanının daha geniş olması,
- Karışık düzenlenebilecek uygunluk fonksiyonu,
- Genel optimuma kolay ulaşması,
- Çok amaçlı fonksiyona sahip olması,
- Sadece fonksiyon değerlendirmelerini kullanması,
- Farklı sorunlara kolaylıkla uyarlanabilmesi,
- Gürültülü fonksiyonları iyi yönetebilmesi,
- Büyük, anlaşılmayan arama alanlarını kolayca taraması,
- Çok modlu problemler için iyi çözüm paketini döndürebilmesi,
- Amaç fonksiyonunun ve çıktılarının kolay değerlendirilebilmesi,
- Yanıt alanında süreksizliklerin, genel olarak iyileştirme performansı üzerinde çok az etkiye sahip olması,
- Yerel en iyiye takılmaya karşı dirençli olması,
- Büyük ölçekli optimizasyon problemleri için çok iyi performans göstermesi,
- Çok çeşitli optimizasyon problemleri için kullanılabilirliğidir.

Genetik algoritmanın kısıtlılığı ise,

- Uygunluk fonksiyonunun belirlenmesinin zor olması,
- Problemin temsilinin tanımının yapılması,
- Prematüre yakınsama gerçekleşebilmesi,
- Populasyon büyüklüğü, mutasyon oranı, çaprazlama oranı, seçim yöntemi gibi çeşitli parametrelerin seçiminin problem olması,
- Düşüş eğilimlerinin kullanılamaması,
- Soruna özgü bilgileri kolaylıkla dahil edilememesi,
- Yerel en iyileri belirlemede iyi olmaması,
- Etkili sonlandırıcının bulunmaması,
- Basit tek amaçlı işlevler için etkili olmaması,
- Yerel bir arama tekniği ile birleştirilmesi gerekliliği,
- Genel en iyiyi bulmakta sorun yaşatabilmesi,
- Çok sayıda uygunluk fonksiyon değerlendirilmesinin gerekmesi,
- Kurulumunun basit olmamasıdır.

Genetikte özellikler kromozomlar yardımıyla yeni nesile aktarılmaktadır. İki kromozomun (genotipler) bir araya gelmesi (çaprazlama) ile yeni fenotipler oluşur. Bazen bu çaprazlanmış kromozomlar mutasyona uğrarlar ve özelliklerinde bozulma oluşur.

Evrim teorisinde doğal seleksiyon sayesinde her şey daha iyiye gider. Yeni organizmalar uyumsuz ise doğa şartlarına uyamazsa yok olmaktadır. Genetik algoritmada bunun karşılığı uygunluk değeridir.

Algoritma bir dizi kromozom ile başlar. Bu kromozomlara başlangıç kromozomları denir. Bu kromozomlardan oluşan diziye popülasyon veya nüfus denmektedir. Bu popülasyon içerisinde uygunluk değerini geçenler hayatta kalırlar ve kabul edilirler. Sırasıyla bu işlemler kromozom kodlama, çaprazlama, mutasyon, seçim yöntemi ve uygunluk testi başlığı altında beş aşamada açıklanacaktır.

5.1.1 Kromozom Kodlamaları

Kromozomlar genlerden oluşur. Kromozom yapısı basit olarak Şekil 5'te gösterilmiştir.

0	0	1	1	0	0	1	1
---	---	---	---	---	---	---	---

Şekil 5: Basit kromozom yapısı

Şekil 5'te gösterilen kromozomda 8 adet gen vardır ve genlerin değerleri 0 ve 1 olarak kodlanmıştır. Kromozom uzunluğu (yukarıdaki örnek için 8) problemdeki parametrelere göre değişmektedir. Bir araç rotalama problemi için nokta sayısı kadar kromozom uzunluğu vardır. Bazı problemler için başladığı noktaya dönüşü kaydetmek amacıyla şehir yada nokta sayısından bir fazla uzunlukta kromozom oluşturulabilmektedir. Bu genler algoritmanın kullanılacağı probleme uygun olarak kodlanacaktır. Dört çeşit kodlama türü vardır (Sivanandam ve Deepa, 2008).

- **İkili Kodlama:** Tüm genler 0 ve 1'lerden oluşurlar. Problemin tipine göre 0 ve 1 ile temsil edilebilir problemler için kullanılır. Var-yok gibi tanımlamalar ile birlikte, bilgisayar ikili kodlamaları için kolayca kullanılabilir. Şekil 5 ikili kodlamaya örnek olarak gösterilebilir.

- **Permutasyon Kodlama:** İş sıralama problemlerinde çok kullanılır. Rakamların matematiksel bir değeri yoktur. Gendeki işin bulunduğu sıra, gerçek problemde istenilen sırayı temsil etmektedir. Şekil 6'da permutasyon kodlamaya örnek bir kodlama görülmektedir. 8 adet işin sıra numaralarına göre önceden sonraya doğru kodlandığı görülebilmektedir.

1	4	2	6	8	3	5	7
---	---	---	---	---	---	---	---

Şekil 6: Permutasyon kodlamalı kromozom

• **Değer Kodlama:** Bazı problemlerde genler üzerinde daha farklı bilgiler tutmak gerekebilir. Bu durumda değer kodlama yöntemi kullanılır. Bu kodlamada çaprazlama ve mutasyon işlemleri, problemin özelliğine göre uyulması gereken kurallar bu kodlamayı kullanan geliştirici tarafından tasarlanmalıdır. Bu bakımdan kullanımı diğer kodlamalara göre daha zordur. Genler üzerinde harfler, semboller, rakamlar veya isimler bulunabilir (sağ,sol,sol,sol,ön, arka,sol,sol,sağ gibi).

• **Ağaç Kodlama:** Bu kodlama az da olsa kullanılan diğer yöntemdir. Genler üzerinde işlemler ve fonksiyonlar barındırabilir.

• **Diğer Kodlama Yöntemleri**

Burada bahsi geçen kodlamalardan başka kodlamalar da oluşturulabilir. Genler üzerinde tutulacak bilgi için herhangi bir sınır yoktur. Uygunluk fonksiyonunda hesaplanabilecek bilgiler farklı şekillerde genler üzerinde tutulabilir.

5.1.2 Seçim Yöntemi

Doğal seçim sonucu yaşayacak en iyi bireylerin diğer nesillere aktarılmasını sağlayan işlemdir. Algoritma içinde iyi sonuçların yeni nüfus içine aktarılmasını sağlar. Elitist strateji, rulet çarkı ve turnuva seçim yöntemleri çok kullanılan yöntemlerdendir. Yöntemlerin birbirlerinden genel farkı, her zaman nüfus içindeki en iyi bireyin aktarılmasının lokal en iyide kalma ihtimalini değiştirmesidir. İleride daha iyi bireylerin oluşmasını sağlayabilecek kötü bireylerin de nüfus içinde kalabilmesi bu aşamada sağlanır.

• **Rulet çarkı yöntemi:** Tüm çözümler hesaplanır ve toplam uygunluk değeri bulunur. En yüksek uygunluk değerine sahip olan birey(ler)in daha yüksek payda olacağı şekilde çark düzenlenir. Genelde maksimizasyon problemlerinde her bir bireyin uygunluk değeri toplam uygunluk değerine bölünerek elde edilen değerler ile yüzdesel oranları bulunur. Yüzdesel oranları hesaplanan sonuçlar içinden oranı daha yüksek olanların seçilme ihtimali daha yüksek olur. Bu sebeple gerçek hayatı temsile daha yakın bir kurgudur.

- **Elitist yöntem:** Kötü bireyi daha çabuk yok edebilmek için, kötü sonucu iyi sonuç ile değiştirerek popülasyonun iyileşmesi sağlanır. Çözümün yerel en iyide kalma ihtimali vardır.

- **Turnuva seçim yöntemi:** Rulet seçim yönteminin aksine sınırlı sayıda kromozomun arasından seçim yapma imkanı verir. Rastgele seçilen kromozomlar arasından en iyisi seçilerek yeni nesile aktarılırlar.

- **Diğer seçim yöntemleri:** Farklı seçim kriterleri uygulanabilir. Örneğin rastgele kromozom seçilerek seçim uygulayan problemler vardır. Bir diğer sık kullanılan seçim yöntemi rütbe seçim yöntemidir. Seçim yöntemi yerel en iyide kalmayı etkileyen en önemli parametredir.

5.1.3 Çaprazlama

Bir kromozomdaki özellikli bir parçanın (örneğin; iş sıralama için kendi içinde en az aylak süreye sahip genler parçası) diğer kromozomdaki özellikli bir parça ile birleştirilerek daha sağlıklı ve özellikli yeni bireylerin oluşturulması aşamasıdır. Hem algoritmanın doğru işlemesi, hem de daha çabuk en iyilerin bulunması için iyi tasarlanması gerekmektedir. Literatürde çok kullanılan birkaç çaprazlama örneği verilerek kısaca anlatılacaktır.

- **Tek noktadan çaprazlama:** Rastgele numara alınır ve bu sıradan iki ebeveyn kromozom bölünür. İlk kromozomun rastgele numaralı sıraya kadar olan birinci kısmı ilk çocuğa aktarılmış ise, ikinci kısmı ikinci çocuğa aktarılır. Diğer kromozomda da ilk ve ikinci kısımlar ters bir şekilde çocuklara aktarılarak yeni nesil kromozomlar üretilmiş olur.

Tablo 3: Tek noktadan çaprazlama işlemi

Kromozom 1 (Anne)	11011 00110110110
Kromozom 2 (Baba)	<u>11011</u> <u>11000011010</u>
Çocuk 1	11011 <u>11000011010</u>
Çocuk 2	<u>11011</u> 00110110110

- **Çift noktadan çaprazlama:** Bu çaprazlamada iki defa rastgele sayı belirlenir ve kromozom üçe bölünür. Ortadaki kısımları diğer kromozom ile değiştirilirler.

Tablo 4: Çift noktadan çaprazlama işlemi

Ana Kromozom 1	11011 1010011 0110
Baba Kromozom 2	11011 1100011 1110
Çocuk 1	11011 1100011 0110
Çocuk 2	11011 1010011 1110

• **Diğer çaprazlama yöntemleri:** Çaprazlamanın iyi yapılması iyi sonuçlar üretilmesi için çok önemlidir. Bu nedenle bahsedilen iki temel ve yaygın çaprazlama metodundan farklı olarak çok sayıda çaprazlama tekniği geliştirilmiştir. Çok noktadan çaprazlama, düzgün çaprazlama, lineer çaprazlama, rastgele çaprazlama, üç ebeveynli çaprazlama, karıştırmalı gibi çaprazlama yöntemleri vardır (Sivanandam ve Deepa, 2008).

Çaprazlama aşamasında yukarıdaki tablolarda kullanılan veriler öncelik ilişkilerine sahip değildirler. Bu nedenle çok kolay şekilde çaprazlama yapılabilmektedir. Ancak öncelik ilişkileri bulunan işlerin çaprazlamaları ek işlemler yapılarak düzeltilmelidir.

5.1.4 Mutasyon

Darwin'e göre daha iyi nesillerin ortaya çıkmasında mutasyon büyük önem taşımaktadır. Algoritma açısından da her ne kadar yeni nesiller iyi sonuçlar üretse de, yerel en iyide kalma ihtimali bulunmaktadır. Bu nedenle mutasyon yerel en iyiden çıkmak için bir fırsat olabilir. Fakat çok mutasyon olursa nüfus içindeki iyi bireyleri de bozabilir. Bu nedenle mutasyonun gerçekleşme ihtimali genellikle düşük tutulur.

Çaprazlamada bahsedilen öncüllük ilişkileri mutasyon işleminin yapılışını da değiştirmektedir. Aşağıdaki Tablo 5'de basit olarak mutasyon işlemi gösterilmiştir. İkili kodlama için basit şekilde bitler 0 ise 1'e veya tersine dönüştürülürler.

Tablo 5: İkili kodlamalı mutasyon gösterimi

Orijinal Kromozom	100111+1001101+0
Mutasyona uğrayan kromozom	1001110100110100

Fakat permutasyon, değer kodlama ve diğer kodlama türleri için rastgele alınan iki sayıdaki genler birbirleri ile yer değiştirilirler. Öncüllük ilişkileri varsa, ilgili gen en sola veya en sağa kaydırılmaya çalışılır.

Tablo 6: Permutasyon kodlamada mutasyon öncesi ve sonrası

İşlem	Kromozom								
Mutasyon Öncesi	9	5	3	4	8	7	6	2	1
Mutasyon Sonrası	9	5	3	6	8	7	4	2	1

9	5	3	4	8	7	6	2	1
---	---	---	---	---	---	---	---	---

Şekil 7: Rassal iki sayıya göre sağa veya sola kaydırma işlemi

9	5	3	4	8	7	6	2	1
---	---	---	---	---	---	---	---	---

Şekil 8: Öncüllük ilişkisine göre işlemi mümkün olan en sağa kaydırma işlemi

5.1.5 Uygunluk Fonksiyonu

Uygunluk değeri hesaplama yöntemleri amaca göre değişmektedir. Uygunluk değerinin hesaplanması için mutlaka bir matematiksel yöntem gereklidir. (Chen, ve ark., 1997) Örneğin sadece öncüllük uygunlukları denetlenebilir.

5.1.6 Sonlandırma Kriteri

Her sezgisel algoritmada olduğu gibi bu algoritmada da durdurma kriterleri belirlenmek zorundadır. Bu kriterler iterasyon sayısı, aynı sonuçları tekrarlama sayısı ve süre olabilir.

En genel anlamda genetik algoritmanın akış çizelgesi EK - 3'te verilmiştir.

6 AYLAK ZAMANI EN KÜÇÜKLEME AMAÇLI ARAÇ ROTALAMA PROBLEMİNİN TANIMI

Kalkış saatleri önceden belirli olan ve günlük olarak gerçekleştirilen seferlerin tümünün tek bir araçla yapılması durumunda, aracın başladığı noktaya geri dönmesi şartı altında tur süresi, sefer sıralamasına göre değişecektir. Bu çalışma; başlangıç konumuna dönecek olan aracın en az aylak süreyle bu turu tamamlamasını sağlayacak rotanın bulunmasını amaçlanmaktadır. Bulunan rota, rota üzerindeki araçların tümüne uygulanarak, günlük seferlerin tamamının zamanında gerçekleşmesini sağlayacak ve tüm araçlar en az aylak süreye sahip olacaklardır. Bu durum aynı zamanda günlük seferleri aksatmadan gerçekleştirebilecek en az araç sayısını da verecektir.

Problemin amacı; çalıştığı müddetçe para kazanan araçların en fazla çalıştırılarak, diğer bir ifadeyle en az aylak süreye sahip olmaları sağlanarak kârın artırılmasıdır. Aracın çalışma süresinin artması aynı zamanda onun aylak süresinin minimize edilmesi anlamını da taşır. Meseleye minimize etme veya maksimize etme çerçevesinden bakılması yaklaşımların diğer kısıtları ele alış şeklini de değiştirecektir. Bu nedenle bu çalışma ARP problemleri için yeni bir amaç fonksiyonu sunmaktadır. Amaç fonksiyonu minimum aylak süre veya maksimum araç verimliliği olarak tanımlanabilir. Araç verimliliği(V) toplam geziş süresinin (G), aylak süre(I) ile toplamına oranıdır ($V = G / (G+I)$).

Aylak süre ve toplam tur sürelerinin hesabı sırasıyla tüme varım ve tümden gelim yöntemine benzetilebilir. Aylak sürenin her nokta için belirlenmesinin başka avantajları da olabilir. Örneğin araç bekleyeceği yerde bir başka tura yönlendirilebilme için uygun süresi varsa yönlendirme yapılabilir. Kısa sürede çok sayıda alternatif rota içeren problemler için daha kolay karar alma imkanı verebilir. Zira bozulan, trafiğe takılan, hava durumuna bağlı olarak geciken (rötar) araçlar daha hızlı bir şekilde yeni rotalara yönlendirilebilirler.

Literatür taraması kısmını takip eden araştırmacıların fark edeceği üzere, bu çalışma mevcut ARP türlerinden bazı benzerlikler içermektedir. Tüm turu tek bir aracın yapması planlanırsa, tek depolu ARP problemlerine benzemektedir. Aracın her noktada

dolum ve boşaltım yapması çoklu depolu (ÇD ARP) problemleri ve Topla-Boşalt türü problemleri de (TD-ARP) kapsamına almaktadır. Kalkış ve varış saatleri yönüyle zaman pencere (ARP-ZP) problem kümesi içindedir. KARP'a benzer şekilde talep ve taşıma durumuna karşın da belli bir kapasiteye sahiptir. Fakat bu benzerliklerin yanında onu özgün kılan yönleri, yeni bir ARP problem türü yapmaktadır.

Bu bilgiler ışığında Aylak Süreyi En Küçükleyen Araç Rotalama Problemi (EKAZARP)'nin farkını birkaç madde ile özetle;

- Araç çalıştığı müddetçe para kazanmaktadır. Bu nedenle minimum aylak süre veya maksimum çalışma/aylak süresi oranı amaç fonksiyonudur.
- Sıradaki iş, sefer başlama süresi dinamik olarak öncekine bağlı olarak değişmektedir (Seferin bir sonraki güne kalma durumudur.).
- Probleme yapılan seferler dikkate alındığı için, her sefer bir kez gerçekleşmektedir. Ancak bazı düğümlere birkaç kez gidilebilir.

Problemin diğer varsayımları ise;

- Alt rota oluşumuna izin verilmemektedir.
- Tüm rota tek araç ile gerçekleşmektedir.
- Zaman çizelgesinde esneklik olmamaktadır.
- Araç kalkış ve varış sürelerini tamamen belirli şekilde gerçekleştirmektedir. Diğer tüm ihtimaller göz ardı edilmiştir.
- Zaman penceresi problemlerinde gösterilen gecikme zamanı gibi cezalar da yoktur.
- Başlanan herhangi bir düğüm depo gibi son olarak dönülen düğüm olmalıdır.
- Her sefer bir kez gerçekleştirilir.
- Her düğümden en az bir adet gidiş ve geliş olmalıdır.
- Temel problem için araçların kapasiteleri ve türleri dikkate alınmamıştır.

Normal çizelgeleme yaklaşımında araçların en az aylak süreye sahip olacağı şekilde rotalama çözümü sonrası çizelgeleme yapılmaktadır. Böyle bir yaklaşımda aylak süre çok az olmaktadır. Fakat bu çalışmanın konusu olan problemdeki çizelgeleme, gerçek hayattan gelmektedir. Yolcular(Müşteriler) varacakları şehre mesai saatleri içinde varmak istemektedirler. Veya gecenin ortasında bir şehirde olmaları, eve gidecek araç bulamama gibi mağduriyetlere sebep olabilmektedir. Bu nedenle sefer süreleri araçların

en verimli şekilde çalışabilecekleri şekilde değil müşteri taleplerine göre çizelgelenmektedir.

Gerçek hayat senaryosunda tüm seferler günlük olarak ilgili saatinde yapılmaktadır. Bu nedenle tüm noktaları bir aracın gezdiği durumda oluşturacağı halka, diğer araçlar ile takip edilerek tüm seferlerin zamanında gerçekleştirilmesi mümkün olacaktır.

Probleme stokastik yöntemler uygulanabilir. Elde müşteri beklentilerine ait veriler vardır. Bunların dağılımı ve olasılıkları çıkarılabilir. Ancak bu çalışmada, tüm araçların kapasite ve taleplerinin bilindiği ve karşılandığı varsayılmıştır.

Hazır olma süresi, tamir ve bakım gibi süreler ulaşım süresinin sonuna eklenerek problem basit şekilde çözülebilir. Problemden araç veya araçların başladıkları noktaya dönememeleri uygun olmayan bir çözümdür.

Problemin veri yapısı aşağıda Tablo 7’de gösterilmiştir. Kalkış ve varış noktaları tabloda ayrıca verilmiştir. Literatürde yer alan sorularda koordinatlar verilmiştir. Burada öklit hesabı değil, iki mesafe arasındaki geçen hesaplanmış süre üzerinden algoritma çalıştırılmıştır. Bahsedilen konu problemin verileri ile alakalı olup, koordinat sistemi ile öklit uzaklığına bağlı olarak algoritma çalıştırılabilir.

Tablo 7: Seferlere ilişkin veri yapısı, örnek 4 sefer.

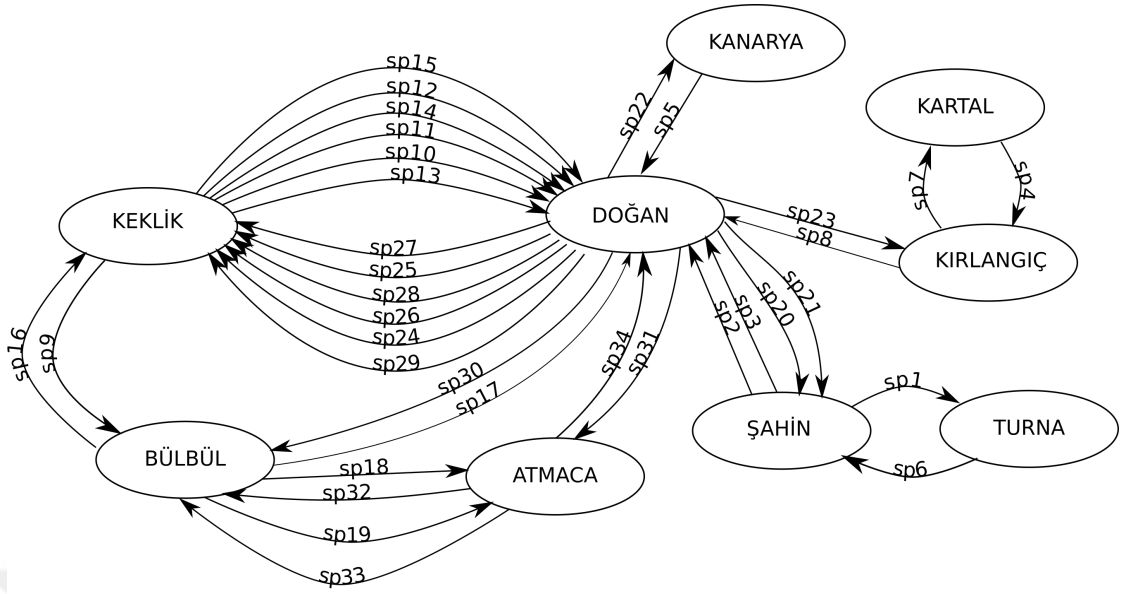
Sefer Numarası	Kalkış Noktası	Varış Noktası	Kalkış Saati	Yolculuk Süresi	Varış Saati
x1	Turna	Şahin	8:00	14:00	22:00
x2	Şahin	Turna	23:00	14:00	13:00
x3	Şahin	Turna	17:00	14:00	7:00
x4	Turna	Şahin	19:00	14:00	9:00



Resim 1: Türkiye’de büyük şehirlerden tatil noktalarına yapılan sefer rotalarının örnek gösterimi

Gerçek verilerden anket yoluyla toplanan verilerdeki şehir adları, haksız rekabet gibi sorunlara neden olmaması için, kuş adları ile kodlanmıştır. Benzer şekilde gerçek şehir adları kullanılarak elde edilen veriler harita üzerinde Resim 1’deki gibi görülebilecektir. Kuş kodlamaları ile oluşan şehir ve seferler, Yay-düğüm biçiminde Şekil 9’da gösterilmiştir. Bu seferlere ait tüm bilgiler tablo biçiminde EK - 4’te verilmiştir. Otobüs firmasına ait tüm seferler ilgili saatlerde her gün gerçekleştirilmektedir.

Şekil 9’da araştırma döneminde firmanın operasyon merkezlerinden birisinin yönettiği seferleri göstermektedir. Şekil üzerindeki her bir yay bir seferi temsil etmektedir. sp9 yayı Tablo 7’de yer alan x1 adlı kaydın (seferin) şekil üzerindeki gösterimidir ve Turna’dan saat 8:00’de hareket edip Şahin’e saat 22:00’de varan, yolculuk süresi 14 saat olan bir seferi temsil etmektedir. Şekil 9 incelendiğinde Kanarya-Doğan arasında günde 1+1 gidiş-dönüş sefer varken, Keklik- Doğan arasında 6 +6 gidiş-geliş seferi olduğu görülmektedir.



Şekil 9: Gerçekleştirilecek seferlerin gösterimi

Bu aşamada problemin optimum çözümünün önemi küçük bir örnek ile anlatılmaya çalışılmıştır. Tablo 8’de bir kayıt beş alanla tanımlanmaktadır. Seferlerden tam bir tur oluşturabilmek için, her seferin bir karşılığı (her gidiş için bir dönüş) olmalıdır ve son varış noktası, ilk hareket noktası olacaktır.

Tablo 8: Seferlere ilişkin veri yapısı (Orjinal problemin 4 seferi)

Sefer Numarası	Kalkış Noktası	Variş Noktası	Kalkış Saati	Yolculuk Süresi	Variş Saati
x1	Turna	Şahin	8:00	14:00	22:00
x4	Turna	Şahin	19:00	14:00	9:00
x3	Şahin	Turna	17:00	14:00	7:00
x2	Şahin	Turna	23:00	14:00	13:00
Toplam Süre (T)				56:00	

Anlaşılabilirlik açısından en küçüklenmek istenen aylak sürenin hesaplanması Tablo 9’da örneklenmiştir. Bu amaçla Tablo 8’deki seferlerden hızlıca bir sıralama; $x4 - x2 - x1 - x3 - x4$ şeklinde Tablo 9’da oluşturulmuştur. Araç başladığı noktaya geri dönmelidir. O nedenle son sefer ile ilk sefer aynıdır.

Tablo 9: Seferlerin sıralanmasına bağlı aylak zaman

Sıra	Sefer Numarası	Variş Saati (A)	Diğer Nokta için Kalkış Saati(D)	Aylak Zaman(D-A)
1	x4	9:00	23:00	14:00
2	x2	13:00	8:00	19:00
3	x1	22:00	17:00	19:00
4	x3	7:00	19:00	12:00
Boş Süre Toplam (B)				64:00
Verimlilik Oranı(Dakikaya çevirilerek) : T / (T+B)				56/120=0.47

Tablo üzerinde yapılan işlem sonucu 2. sıradaki sefer için variş saati 13:00 olduğu ve kalkış saati olan 08:00'ı geçtiği için araç ertesi güne kadar beklemek zorunda olduğu görülmektedir. Tüm tur için çıkan verimlilik oranı yaklaşık %47'dir. Eğer sıralama x4 – x3 – x1 – x2 – x4 şeklinde yapılırsa Tablo 10'daki değerlere ulaşılır.

Tablo 10: Seferlerin farklı sıralanmasına bağlı aylak zaman

Sıra	Sefer Numarası	Variş Saati (A)	Diğer Nokta için Kalkış Saati(D)	Aylak Zaman(D-A)
1	x4	9:00	17:00	8:00
2	x3	7:00	8:00	1:00
3	x1	22:00	23:00	1:00
4	x2	13:00	19:00	6:00
Boş Süre Toplam (B)				16:00
Verimlilik Oranı(Dakikaya çevirilerek) : T / (T+B)				56/72=0.78

Tablo 10'a ilişkin deneme sonucunda verimlilik %78'e çıkmıştır. Birinci sıralamada toplam süre 120 saat olup, ikinci sıralamada 72 saattir. 48 saat kazanılmıştır. Her sefer günlük olarak yapıldığına göre, bu 48 saat, iyi ve kötü sonuç arasında 2 araç daha az kullanılacak anlamına da gelmektedir. Dört sefer için 4 araç yerine toplam 72 saatlik rotaya uygun olarak 3 araç ile bu seferlerin aksatılmadan gerçekleştirilebileceği de görülmektedir. Buradaki problemde sefer sayıları küçük ve alternatif şehirler bulunmadığı için hesabı kolay küçük bir problemdir. Ancak daha fazla sefer sayısı ve variş noktası ile problem çok daha karmaşık hale gelmektedir.

6.1 PROBLEMİN AMAÇ FONKSİYONUNUN MATEMATİKSEL MODELİ

Amaç fonksiyonu:

$$\min \sum_{i=0}^n \sum_{j \neq i, j=0}^n c_{i,j} x_{i,j}$$

$c_{i,j}$, i'den j'ye gidiş için maliyet

$x_{i,j}$, i'den j'ye gidilmiş olduğunu gösteren tamsayı değişkendir.

ENAZARP için farklılık ise şöyledir;

$t_{i,j}$, i'den j'ye varış zamanı ile kalkış zamanı arasındaki aylak zaman olmak üzere;

$$\min \sum_{i=0}^n \sum_{j \neq i, j=0}^n t_{i,j} x_{ij}$$

olarak minimum aylak süreyi bulmayı amaçlamaktadır. Veya maksimizasyon problemi olarak da ele alınabilir. Bu durumda;

$m_{i,j}$, i'den j'ye gidiş süresi olmak üzere

$$\max \sum_{i=0}^n \sum_{j \neq i, j=0}^n \left(\frac{m_{i,j}}{m_{i,j} + t_{i,j}} \right) * x_{ij}$$

şeklinde verimlilik hesabı haline gelir.

Problemin kendine özgü ek kısıtları olacaktır. Örneğin bir düğümden gidilebilecek düğümler sınırlı olabilir. Bu yaklaşım atölye tipi üretim yöntemindeki öncüllük ilişkisi kısıtına benzetilebilir. Problem diğer ARP problemlerine de uyarlanırken, tüm kısıtları alabilir. Fakat amaç fonksiyonu yalındır.

Problemin genetik algoritma ile çözümüne geçmeden önce iki örnek yaklaşım sunulmuştur. Bu yaklaşımlara göre her düğüm bir sonraki ihtimallerin tümüne bakarak en az aylak süreyi sağlayacak çözüme yönelmiştir. Tüm yaklaşımlar için başladığı noktaya dönemeyen çözümler için algoritmada özel bir işlem yapılmıştır. Bu sayede hep kabul edilebilen sonuçlar üretilebilmektedir. Bu işleme 7.1.1 Kromozom Oluşturma başlığı altında ayrıntılı şekilde değinilecektir.

6.2 SIRADAKİ EN KÜÇÜK AYLAK SÜRE (SEKAS)

Bir başlangıç deneme çözüm elde edilmesi konusunda “Sıradaki En Küçük Aylak Süre” yaklaşımı için aşağıdaki adımlardan oluşan bir algoritma geliştirilmiş ve bilgisayar programı ile uygulanmıştır. Algoritma seferler arasındaki aylıklığın en küçüklenmesini amaçlamaktadır. Bu amaçla sıradaki muhtemel seferler arasındaki seçimde en küçük aylak zamanı olan şehri seçerek ilerlemektedir.

Adım 1. Her seferi bir liste elemanı olacak şekilde belleğe, diskten yükle.

Adım 2. Kaynak ve hedeflere ait kalkış ve yolculuk sürelerini dakikaya çevir.

Adım 3. Bir başlangıç sefer (kaynak-hedef noktası) seç. Turda ilk sıraya yerleştir.

Adım 4. Muhtemel seferler adında boş bir liste oluştur.

Adım 5. Sıralanmamış seferler arasından, mevcut noktadan gidilebilecek olan noktaları Muhtemel sefer listesine ekle.

Adım 6. Muhtemel sefer listesinden gidildiği durumdaki aylak zaman değerlerini hesapla.

Adım 7. En küçük aylak süreye sahip seferi sıraya ekle.

Adım 8. Tüm seferler gözden geçirilmediyse Adım 4'e git.

Adım 9. Tüm seferler sıralamaya dahil edildiyse DUR

Adım 10. Çözümü raporla

6.3 SIRADAKİ EN BÜYÜK AYLAK SÜRE (SEBAS)

Bir önceki algoritmadaki Adım 7, “en büyük aylak süreye sahip seferi sıraya ekle” şeklinde değiştirilerek sonuçlar elde edilmiştir. Bu aşamada, sıradaki en büyük aylak sürenin daha kötü sonuç üretebileceği önyargısı yanlış olabilir. Zira sona doğru kalan işlerde belki de en az aylak süreli işler yoğunlaşacaktır. Ve yine unutulmamalıdır ki her iki yaklaşım da sezgiseldir ve optimalliği garanti etmemektedir.

Yapılan çözümlenelerde örnek olarak ilgilenilen operasyon merkezine ait 34 adet sefer kullanılmıştır. Örnek problemin tüm verileri EK - 4'te verilmiştir.

Tablo 11 ve Tablo 12'de SEBAS ve SEKAS için çalıştırılan algoritma çoğunlukla tüm turu tamamlayıp ilk başladığı noktaya dönememiştir. Bu nedenle kabul edilen bir sonuç üretememişlerdir. Aynı zamanda her iki çözüm yöntemi için başlangıç noktasının seçiminin ne kadar önemli olduğu anlaşılmıştır. SEKAS ile çözümü gösteren Tablo 11'deki X ile ifadesi, atanacak yeni sefer bulunmadığını ve turun tamamlanmadığını göstermektedir. Her satır başlangıç noktalarını (B.N) göstermektedir.

sp18	sp18,sp34,sp24,sp10,sp25,sp11,sp26,sp9,sp16,sp12,sp20,sp2,sp27,sp13,sp28,sp14,sp31,sp32,sp19,sp33,sp17,sp30,X,X,X,X,X,X,X,X,X,X	Sonuç Yok.
sp19	sp19,sp33,sp16,sp12,sp20,sp2,sp27,sp13,sp28,sp10,sp25,sp11,sp26,sp9,sp18,sp34,sp24,sp14,sp31,sp32,sp17,sp30,X,X,X,X,X,X,X,X,X,X	Sonuç Yok.
sp20	sp20,sp2,sp27,sp13,sp28,sp10,sp25,sp11,sp26,sp9,sp18,sp34,sp24,sp14,sp31,sp32,sp19,sp33,sp16,sp12,sp22,sp5,sp30,sp17,sp29,sp15,sp23,sp8,sp21,sp3,X,X,X,X	Sonuç Yok.
sp21	sp21,sp2,sp20,sp3,sp22,sp5,sp27,sp13,sp28,sp10,sp25,sp11,sp26,sp9,sp18,sp34,sp24,sp14,sp31,sp32,sp19,sp33,sp16,sp12,sp30,sp17,sp29,sp15,sp23,sp8,X,X,X,X	Sonuç Yok.
sp22	sp22,sp5,sp20,sp2,sp27,sp13,sp28,sp10,sp25,sp11,sp26,sp9,sp18,sp34,sp24,sp14,sp31,sp32,sp19,sp33,sp16,sp12,sp30,sp17,sp29,sp15,sp23,sp8,sp21,sp3,X,X,X,X	Sonuç Yok.
sp23	sp23,sp8,sp31,sp34,sp24,sp10,sp25,sp11,sp26,sp9,sp18,sp32,sp19,sp33,sp16,sp12,sp20,sp2,sp27,sp13,sp28,sp14,sp21,sp3,sp22,sp5,sp30,sp17,sp29,sp15,X,X,X,X	Sonuç Yok.
sp24	sp24,sp10,sp25,sp11,sp26,sp9,sp18,sp34,sp28,sp13,sp31,sp32,sp19,sp33,sp16,sp12,sp20,sp2,sp27,sp14,sp21,sp3,sp22,sp5,sp30,sp17,sp29,sp15,sp23,sp8,X,X,X,X	Sonuç Yok.
sp25	sp25,sp11,sp26,sp9,sp18,sp34,sp24,sp10,sp28,sp13,sp31,sp32,sp19,sp33,sp16,sp12,sp20,sp2,sp27,sp14,sp21,sp3,sp22,sp5,sp30,sp17,sp29,sp15,sp23,sp8,X,X,X,X	Sonuç Yok.
sp26	sp26,sp9,sp18,sp34,sp24,sp10,sp25,sp11,sp30,sp17,sp29,sp13,sp28,sp14,sp31,sp32,sp19,sp33,sp16,sp12,sp20,sp2,sp27,sp15,sp22,sp5,sp23,sp8,sp21,sp3,X,X,X,X	Sonuç Yok.
sp27	sp27,sp13,sp28,sp10,sp25,sp11,sp26,sp9,sp18,sp34,sp24,sp14,sp31,sp32,sp19,sp33,sp16,sp12,sp20,sp2,sp22,sp5,sp30,sp17,sp29,sp15,sp23,sp8,sp21,sp3,X,X,X,X	Sonuç Yok.
sp28	sp28,sp10,sp25,sp11,sp26,sp9,sp18,sp34,sp24,sp13,sp31,sp32,sp19,sp33,sp16,sp12,sp20,sp2,sp27,sp14,sp21,sp3,sp22,sp5,sp30,sp17,sp29,sp15,sp23,sp8,X,X,X,X	Sonuç Yok.
sp29	sp29,sp11,sp26,sp9,sp18,sp34,sp24,sp10,sp25,sp13,sp28,sp14,sp31,sp32,sp19,sp33,sp16,sp12,sp20,sp2,sp27,sp15,sp22,sp5,sp30,sp17,sp23,sp8,sp21,sp3,X,X,X,X	Sonuç Yok.
sp30	sp30,sp17,sp26,sp9,sp18,sp34,sp24,sp10,sp25,sp11,sp29,sp13,sp28,sp14,sp31,sp32,sp19,sp33,sp16,sp12,sp20,sp2,sp27,sp15,sp22,sp5,sp23,sp8,sp21,sp3,X,X,X,X	Sonuç Yok.
sp31	sp31,sp34,sp24,sp10,sp25,sp11,sp26,sp9,sp18,sp32,sp19,sp33,sp16,sp12,sp20,sp2,sp27,sp13,sp28,sp14,sp21,sp3,sp22,sp5,sp30,sp17,sp29,sp15,sp23,sp8,X,X,X,X	Sonuç Yok.
sp32	sp32,sp19,sp33,sp16,sp12,sp20,sp2,sp27,sp13,sp28,sp10,sp25,sp11,sp26,sp9,sp18,sp34,sp24,sp14,sp31,X,X,X,X,X,X,X,X,X,X	Sonuç Yok.
sp33	sp33,sp16,sp12,sp20,sp2,sp27,sp13,sp28,sp10,sp25,sp11,sp26,sp9,sp18,sp34,sp24,sp14,sp31,sp32,sp19,X,X,X,X,X,X,X,X,X,X	Sonuç Yok.
sp34	sp34,sp24,sp10,sp25,sp11,sp26,sp9,sp18,sp32,sp19,sp33,sp16,sp12,sp20,sp2,sp27,sp13,sp28,sp14,sp31,X,X,X,X,X,X,X,X,X,X	Sonuç Yok.

sp18	sp18,sp33,sp19,sp32,sp17,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp20,sp2,sp31,sp34,sp25,sp9,sp16,sp14,sp24,sp15,sp29,sp12,sp26,sp13,sp30	%40,42
sp19	sp19,sp32,sp17,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp20,sp2,sp31,sp33,sp18,sp34,sp25,sp9,sp16,sp14,sp24,sp15,sp29,sp12,sp26,sp13,sp30	%40,42
sp20	sp20,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp22,sp5,sp21,sp2,sp31,sp33,sp18,sp32,sp17,sp24,sp9,sp19,sp34,sp25,sp15,sp29,sp12,sp26,sp14,sp30,sp16,sp13	%40,42
sp21	sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp22,sp5,sp31,sp33,sp18,sp32,sp17,sp20,sp2,sp24,sp9,sp19,sp34,sp25,sp15,sp29,sp12,sp26,sp14,sp30,sp16,sp13	%40,42
sp22	sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp20,sp2,sp31,sp33,sp18,sp32,sp17,sp24,sp9,sp19,sp34,sp25,sp15,sp29,sp12,sp26,sp14,sp30,sp16,sp13	%40,42
sp23	sp23,sp7,sp4,sp8,sp28,sp11,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp29,sp9,sp19,sp32,sp17,sp20,sp2,sp31,sp33,sp18,sp34,sp25,sp15,sp24,sp12,sp26,sp14,sp30,sp16,sp13	%39,38
sp24	sp24,sp11,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp9,sp19,sp32,sp17,sp20,sp2,sp31,sp33,sp18,sp34,sp25,sp15,sp29,sp12,sp26,sp14,sp30,sp16,sp13	%39,38
sp25	sp25,sp9,sp19,sp32,sp17,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp20,sp2,sp31,sp33,sp18,sp34,sp29,sp15,sp24,sp12,sp26,sp14,sp30,sp16,sp13	%39,38
sp26	sp26,sp15,sp20,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp22,sp5,sp21,sp2,sp31,sp33,sp18,sp32,sp17,sp24,sp9,sp19,sp34,sp25,sp12,sp29,sp14,sp30,sp16,sp13	%39,38
sp27	sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp22,sp5,sp21,sp1,sp6,sp3,sp20,sp2,sp31,sp33,sp18,sp32,sp17,sp24,sp9,sp19,sp34,sp25,sp15,sp29,sp12,sp26,sp14,sp30,sp16,sp13	%40,42
sp28	sp28,sp11,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp24,sp9,sp19,sp32,sp17,sp20,sp2,sp31,sp33,sp18,sp34,sp25,sp15,sp29,sp12,sp26,sp14,sp30,sp16,sp13	%40,42
sp29	sp29,sp9,sp19,sp32,sp17,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp20,sp2,sp31,sp33,sp18,sp34,sp25,sp15,sp24,sp12,sp26,sp14,sp30,sp16,sp13	%39,38
sp30	sp30,sp16,sp14,sp28,sp11,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp24,sp9,sp19,sp32,sp17,sp20,sp2,sp31,sp33,sp18,sp34,sp25,sp15,sp29,sp12,sp26,sp13	%39,38
sp31	sp31,sp33,sp18,sp32,sp17,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp20,sp2,sp24,sp9,sp19,sp34,sp25,sp15,sp29,sp12,sp26,sp14,sp30,sp16,sp13	40,42
sp32	sp32,sp17,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp20,sp2,sp31,sp33,sp18,sp34,sp25,sp9,sp19,X,X,X,X,X,X,X,X	Sonuç Yok.
sp33	sp33,sp18,sp32,sp17,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp20,sp2,sp31,sp34,sp25,sp9,sp19,X,X,X,X,X,X,X,X	Sonuç Yok.
sp34	sp34,sp25,sp9,sp19,sp32,sp17,sp22,sp5,sp21,sp1,sp6,sp3,sp27,sp10,sp23,sp7,sp4,sp8,sp28,sp11,sp20,sp2,sp31,sp33,sp18,X,X,X,X,X,X,X,X	Sonuç Yok.

Her iki algoritma ile yapılan çözümlerde görüleceği üzere başlangıç noktasının seçimi sonucun kalitesini doğrudan etkilemektedir. Her çıkan araç için tur

tamamlanamamıştır. Tablo 11’deki sp4 (Kartal-Kırlangıç) satırına dikkat edilirse, araç sp4 seferini gerçekleştirdikten sonra iki alternatifi vardır. Ya yeniden Kartal (sp7) noktasına dönecek, ya da Doğan’a (sp8) gidecektir. Doğan’a gidecek araç ertesi güne kadar bekleyeceği için daha az bekleme süresi olan sp7’yi seçmiştir. Buradan başka bir sefer olmadığı için, araç orada diğer noktalara gidemeyerek, kabul edilemeyen bir sonuç üretmiştir. Fakat bu yaklaşım alt turlar oluşturulması izin verilen problemler için daha iyi yaklaşımlar önerebilir. Verimliliğin artmasına imkan verebilir.

Tablo 11’deki sp5 ile başlayan sıralamanın verileri Tablo 13’te verilmiştir. SEKAS’tan beklenildiği üzere sıradaki seferler en küçük aylak süresi olandan seçilmiştir.

Tablo 13: Tamamlanamamış SEKAS için örnek çıktı

Sefer No	Kaynak	Hedef	Kalkış Süre	Varış Süre	Aylak Süre
sp5	KANARYA	DOĞAN	690	514	TUR SONU YOK
sp20	DOĞAN	ŞAHİN	1380	650	176
sp2	ŞAHİN	DOĞAN	600	739	10
sp27	DOĞAN	KEKLİK	0	480	101
sp13	KEKLİK	DOĞAN	510	485	30
sp28	DOĞAN	KEKLİK	1080	480	85
sp10	KEKLİK	DOĞAN	300	485	180
sp25	DOĞAN	KEKLİK	840	480	55
sp11	KEKLİK	DOĞAN	1410	495	90
sp26	DOĞAN	KEKLİK	540	470	75
sp9	KEKLİK	BÜLBÜL	1020	1070	10
sp18	BÜLBÜL	ATMACA	900	950	250
sp34	ATMACA	DOĞAN	510	364	100
sp24	DOĞAN	KEKLİK	960	480	86
sp14	KEKLİK	DOĞAN	630	485	630
sp31	DOĞAN	ATMACA	1140	375	25
sp32	ATMACA	BÜLBÜL	630	950	555
sp19	BÜLBÜL	ATMACA	420	950	280
sp33	ATMACA	BÜLBÜL	0	950	70
sp16	BÜLBÜL	KEKLİK	1020	1070	70
sp12	KEKLİK	DOĞAN	750	485	100
sp22	DOĞAN	KANARYA	60	530	265
.....

SEBAS yöntemi ile tamamlanmamış bir turu açık şekilde Tablo 13’teki gibi göstermeye gerek duyulmamıştır. Tablo 12 takip edilirse, sıradaki her seçim için en büyük aylak sürelerin tercih edildiği görülecektir.

Çözüm üreten bir sonuç üretmek için yeni bir algoritma uygulanmıştır. Algoritma geri izleme yöntemindeki gibi bir işlem uygulanarak, her zaman tüm turu tamamlayan ve ilk başladığı noktaya dönmesinin mümkün kılındığı şekilde geliştirilmiştir. Bu işlem ileride 7.1.1 Kromozom Oluşturma başlığı altında ayrıntılı olarak anlatılmıştır. Elde edilen iki örnek çözüm aşağıda Geri izlemeli SEKAS ve SEBAS adında Tablo 14 ve Tablo 15’te verilmiştir. Toplam yolda geçen süre 22119 dakikadır. Tablonun tümü geri izleme yöntemiyle çözülmemiştir. Başlangıç noktasına göre kabul edilen ilk sıralama çözüm olarak kabul edilmiştir. Verimlilik hesabı Tablo 9 ve Tablo 10’da gösterildiği üzere toplam yolculuk süresinin, aylak süre ve toplam yolculuk süresine bölümü ile elde edilmektedir.

Tablo 14: Geri İzlemeli SEKAS Yaklaşımı İçin Örnek Çözüm

Sefer No	Kaynak	Hedef	Kalkış Süre	Variş Süre	Aylak Süre
sp1	ŞAHİN	TURNA	1320	890	850
sp6	TURNA	ŞAHİN	1140	890	370
sp2	ŞAHİN	DOĞAN	600	739	10
sp20	DOĞAN	ŞAHİN	1380	650	41
sp3	ŞAHİN	DOĞAN	720	739	130
sp22	DOĞAN	KANARYA	60	530	41
sp5	KANARYA	DOĞAN	690	514	100
sp27	DOĞAN	KEKLİK	0	480	236
sp13	KEKLİK	DOĞAN	510	485	30
sp28	DOĞAN	KEKLİK	1080	480	85
sp10	KEKLİK	DOĞAN	300	485	180
sp25	DOĞAN	KEKLİK	840	480	55
sp11	KEKLİK	DOĞAN	1410	495	90
sp26	DOĞAN	KEKLİK	540	470	75
sp9	KEKLİK	BÜLBÜL	1020	1070	10
sp18	BÜLBÜL	ATMACA	900	950	250
sp34	ATMACA	DOĞAN	510	364	100
sp24	DOĞAN	KEKLİK	960	480	86
sp14	KEKLİK	DOĞAN	630	485	630
sp31	DOĞAN	ATMACA	1140	375	25
sp32	ATMACA	BÜLBÜL	630	950	555
sp19	BÜLBÜL	ATMACA	420	950	280
sp33	ATMACA	BÜLBÜL	0	950	70
sp16	BÜLBÜL	KEKLİK	1020	1070	70
sp12	KEKLİK	DOĞAN	750	485	100
sp30	DOĞAN	BÜLBÜL	540	710	745
sp17	BÜLBÜL	DOĞAN	1290	679	40
sp29	DOĞAN	KEKLİK	600	480	71
sp15	KEKLİK	DOĞAN	930	485	1290
sp23	DOĞAN	KIRLANGIÇ	720	454	745
sp7	KIRLANGIÇ	KARTAL	1140	830	1406

sp4	KARTAL	KIRLANGIÇ	1290	830	760
sp8	KIRLANGIÇ	DOĞAN	600	485	1360
sp21	DOĞAN	ŞAHİN	1200	710	115

Sıradaki şehir seçiminde En Uzun Aylak Zaman temel alındığında elde edilen sonuçlar aşağıdaki gibidir.

Tablo 15: Geri İzlemeli SEBAS Yaklaşımı İçin Örnek Çözüm

Sefer No	Kaynak	Hedef	Kalkış Süre	Varış Süre	Aylak Süre
sp1	ŞAHİN	TURNA	1320	890	730
sp6	TURNA	ŞAHİN	1140	890	370
sp3	ŞAHİN	DOĞAN	720	739	130
sp27	DOĞAN	KEKLİK	0	480	1421
sp10	KEKLİK	DOĞAN	300	485	1260
sp23	DOĞAN	KIRLANGIÇ	720	454	1375
sp7	KIRLANGIÇ	KARTAL	1140	830	1406
sp4	KARTAL	KIRLANGIÇ	1290	830	760
sp8	KIRLANGIÇ	DOĞAN	600	485	1360
sp28	DOĞAN	KEKLİK	1080	480	1435
sp11	KEKLİK	DOĞAN	1410	495	1290
sp22	DOĞAN	KANARYA	60	530	1035
sp5	KANARYA	DOĞAN	690	514	100
sp21	DOĞAN	ŞAHİN	1200	710	1436
sp2	ŞAHİN	DOĞAN	600	739	130
sp31	DOĞAN	ATMACA	1140	375	1241
sp33	ATMACA	BÜLBÜL	0	950	1365
sp18	BÜLBÜL	ATMACA	900	950	1390
sp32	ATMACA	BÜLBÜL	630	950	220
sp17	BÜLBÜL	DOĞAN	1290	679	1150
sp24	DOĞAN	KEKLİK	960	480	431
sp9	KEKLİK	BÜLBÜL	1020	1070	1020
sp19	BÜLBÜL	ATMACA	420	950	1210
sp34	ATMACA	DOĞAN	510	364	580
sp25	DOĞAN	KEKLİK	840	480	1406
sp15	KEKLİK	DOĞAN	930	485	1050
sp29	DOĞAN	KEKLİK	600	480	625
sp12	KEKLİK	DOĞAN	750	485	1110
sp26	DOĞAN	KEKLİK	540	470	745
sp14	KEKLİK	DOĞAN	630	485	1060
sp30	DOĞAN	BÜLBÜL	540	710	865
sp16	BÜLBÜL	KEKLİK	1020	1070	1210
sp13	KEKLİK	DOĞAN	510	485	1300
sp20	DOĞAN	ŞAHİN	1380	650	385

Geri izlemeli SEKAS yaklaşımı için örnek çözüm Tablo 14’te verilmiştir. Sıra “sp1,sp6,sp2,sp20,sp3,sp22,sp5,sp27,sp13,sp28,sp10,sp25,sp11,sp26,sp9,sp18,sp34,sp2

4,sp14,sp31,sp32,sp19,sp33,sp16,sp12,sp30,sp17,sp29,sp15,sp23,sp7,sp4,sp8,sp21” şeklinde oluşmuştur. Toplam yolculuk süresi 22119 dakika, aylak süre ise 11001 dakikadır. Buna göre verimlilik $22119/(22119+11001) = \%66,78$ olarak hesaplanmıştır.

Geri izlemeli SEBAS yaklaşımı için, Tablo 15 için sıra, “sp1, sp6, sp3, sp27, sp10, sp23, sp7, sp4, sp8, sp28, sp11, sp22, sp5, sp21, sp2, sp31, sp33, sp18, sp32, sp17, sp24, sp9, sp19, sp34, sp25, sp15, sp29, sp12, sp26, sp14, sp30, sp16, sp13, sp20” şeklinde olup, aylak süre 32601 dakikadır. Verimlilik oranı %40,42’dir. Tablo 15 dikkatlice incelenirse, ileriye doğru çok düşük aylak süreli seferler de oluşmuştur. Buna göre iyi sonuçlar üretebilme ihtimali de bulunmaktadır.

Geri izlemeli SEBAS ve SEKAS yaklaşımlarına ait iki örnek çözüme baktığımızda (Tablo 14 ve Tablo 15), aylak zaman farkı olarak $32601 - 11001 = 21600$ dakika gibi çok büyük bir fark vardır. Bu da aylak süresi az olan turun 15 gün daha kısa sürmesi anlamına gelir ve daha uzun süren kötü sonuca göre 15 aracın daha az kullanılacağını ifade eder. 15 araç için gerekli sermayeden tasarruf edilmesi anlamı da taşır. Bu sonuç optimizasyon yapılmasının çok önemli olduğuna iyi bir örnektir ve problemin en iyi çözümünü garanti eden bir yöntemle çözümlenmesinin önemine işaret etmektedir. Ayrıca 34 sefer için 34 adet araç yerine $(22119+11001)/1440 = 23$ araç ile bu işlemin yapılabileceğini göstermektedir.

Burada ayrıca bahsetmek gereken başka bir nokta daha vardır. Doğan-Keklik arasındaki bir seferden başlanması algoritmanın uzun sürmesine sebep olabilmektedir. Örneğin Geri İzlemeli SEKAS için sp21 numaralı sefer ile başlayan tur için kabul edilen bir sonucun bulunması çok uzun sürmüştür. Bunun nedeni sıradaki sefer seçimlerinde, bir noktadan sonra diğer seferlerin gerçekleşmeyeceği düğümde sefer tamamlanmıştır. O sebeple geriye dönmeye çalışılarak ilgili seçimler yeniden denenmiştir. Fakat geri dönerken de alternatif yollar denendiği için çok fazla hesap yapmayı gerektirmektedir. Böyle bir durumun ortaya çıkması da, kabul edilebilir süreler içinde sonuçlar bulabilmek için, sezgisel algoritma kullanımının önemine işaret etmektedir.

Geri izlemeli SEBAS ve SEKAS tüm başlangıç noktaları için çalıştırıldığında üretilen sonuçlar, sp1’den sp34’e kadar şu şekildedir.

Tablo 16: Geri izlemeli SEKAS ve SEBAS tüm başlangıç noktaları için verimlilik oranları (Sefer No- Verimlilik)

Sefer No	Geri İzlemeli SEKAS Verimlilik	Geri İzlemeli SEBAS Verimlilik
sp1	%66,78	%40,42
sp2	%66,78	%40,42
sp3	%66,78	%40,42
sp4	%66,78	%39,38
sp5	%66,78	%40,42
sp6	%66,78	%40,42
sp7	%66,78	%39,38
sp8	%66,78	%39,38
sp9	%64,00	%40,42
sp10	%66,78	%40,42
sp11	%66,78	%40,42
sp12	%66,78	%40,42
sp13	%66,78	%39,38
sp14	%66,78	%39,38
sp15	%66,78	%40,42
sp16	%64,00	%39,38
sp17	%64,00	%40,42
sp18	%64,00	%40,42
sp19	%64,00	%40,42
sp20	%64,00	%40,42
sp21	Uzun Süre İptal	%40,42
sp22	%64,00	%40,42
sp23	Uzun Süre İptal	%39,38
sp24	%64,00	%39,38
sp25	%64,00	%39,38
sp26	%66,78	%39,38
sp27	%64,00	%40,42
sp28	%64,00	%40,42
sp29	%66,78	%39,38
sp30	%66,78	%39,38
sp31	%66,78	%40,42
sp32	%66,78	%40,42
sp33	%64,00	%41,51
sp34	%66,78	%40,42

Tüm sefer hesapları aynı bilgisayar ve algoritma ile 5 dakikadan az sürmüştür. Buna karşın, Tablo 16'de Geri İzlemeli SEKAS için 2 seferin (sp21 ve sp23) hesabı çok uzun sürdüğü (7 saatin üzerinde) için iptal edilmiştir. Başka problemler için çözümün her zaman kabul edilebilir süre içinde bulunması ihtimalinin olmadığını da kanıtıdır. Bu durum diğer sezgisel yaklaşımların kullanılmasının daha isabetli olacağını göstermektedir.

Tablo 16'da pek çok sıralamanın aynı sonucu vermesinin temel nedeni Şekil 9 üzerinde görülebilir. Bazı düğümler arası (Doğan,Kartal, Kırılgaç,Keklik, Turna ve Kanarya) tek gidiş geliş vardır. Bu nedenle tüm seferleri yapan araç için rota mecburi

olarak bu noktalarda aynılařmaktadır. Bu veriler gerek hayattan alındığı iin zerinde bir maniplasyon yapılmamıřtır. Eėer her nokta iin daha fazla sefer olsaydı, sonular da birbirinden ok farklı olacaktır.

Seferler arası tm aylak sreler ek EK - 5'te gsterilmiřtir.



7 GENETİK ALGORİTMA İLE ÇÖZÜMÜ

7.1 GENETİK ALGORİTMA ADIMLARI

Genetik algoritmanın ideal sonuçları üretmesi için probleme özgü doğru parametreler seçilmesi çok önemlidir. Genetik algoritmanın parametreleri, sonlandırma kriteri için aynı sonucu tekrar sayısı veya toplam iterasyon sayısı, çaprazlama ve mutasyon oranı, seçim kriteri ve nüfus sayısıdır. Seçim kriterlerinin de kendine mahsus parametreleri olabilmektedir.

Problemin ideal çözüme yaklaştığı sırada sonlandırma kriterinin gerçekleşmesi, en zayıf yönlerinden biridir. Literatürde yer alan araştırmalardan biri, soru türlerine göre ideal olan genetik algoritma parametrelerinin bulunmasıdır (Baker ve Ayechev, 2003; Potvin, 2009). Bu çalışmadaki problemin kendine özgü yapısı nedeniyle literatürde genetik algoritmaya ait optimum parametreler bulunmamaktadır. Bu nedenle genetik algoritmanın probleme özgü olarak parametrelerinin de test edilmesi gerekmektedir. Sonlandırma kriteri olan iterasyon sayısı da , problemin hızlı çözülmesi için önemlidir. Bu bakımdan parametreler örnek soru üzerinde denenerek, bazı sonuçlar elde edilmiştir.

Sonuçlara geçmeden önce kromozom oluşturma, çaprazlama ve mutasyon işlemlerinin nasıl yapıldığı hakkında bilgi verilmesi ileride araştırma yapacak araştırmacılara kolaylık açısından önem arz etmektedir.

7.1.1 Kromozom Oluşturma

Kromozom uzunluğu sefer sayısı kadardır. Örnek problemimizde 34 sefer bulunduğu için kromozom uzunluğu 34'tür. Genler üzerine değer kodlama yapılarak, sefer numaraları tutulmaktadır. Şekil 10'da 34 genden oluşan bir kromozom bulunmaktadır. Kısaca göstermek adına 26 gen nokta işaretleri ile atlanmıştır.

sp34	sp25	sp9	sp19	sp32	sp17	sp22	sp18
------	------	-----	------	------	------	------	-------	------

Şekil 10: Örnek problem için kromozom kodlama

Kromozom oluşturulurken, öncüllük ilişkileri ve başlanılan noktaya dönme zorunluluğu, bir çok kabul edilmeyen kromozom oluşmasına (tur oluşmamasına) sebep

olmaktadır. Bu yüzden geri izleme yöntemine benzer şekilde tamamı mümkün olan kromozomlar üretilmektedir. Literatürde rastgele oluşturulan kromozomlar, uygunluk fonksiyonunda elenmektedir. Bu çalışmadaki yaklaşımda tüm mümkün kromozomlar arasından en iyilerini seçme yaklaşımı yapılmıştır.

Kromozom oluşturma algoritmasını kısaca anlatmak için bazı değişkenleri tanımlamak gerekmektedir. Bu değişkenler;

V: Tüm seferlerin bulunduğu listedir.

S: Sıralanmamış seferler kümesidir. (İlk iterasyon için tüm işlerin bulunduğu listedir.).

L: Sıralanan seferler kümesidir. (İterasyon başlangıcında boş kümedir. $V = L+S$)

M: Mevcut düğümden gidilebilecek ve sıralanan seferler kümesinde bulunmayan seferler listesi, muhtemellerin kısaltması olarak M kullanılmıştır. (Sıralanmamış işler kümesinden oluşturulur, başangıçta $M=S$ 'dir. Daha sonra $sefer \notin L, M \subseteq S$)

K: Kromozom

duzey: 0 ile kromozom uzunluğu aralığında bulunur. Yerleştirilen iş kadar artırılır. L listesi uzunluğu kadardır.

musait_duzey: Algoritmanın geri izleme yöntemine benzer şekilde, başlangıç noktasından ileriye doğru kesinlikle bir kromozom üretilmesini sağlayan en önemli değişkendir. Her adım için ayrı ayrı tutulur. musait_duzey_{duzey} şeklinde gösterilir.

Adım 1: S kümesini, tüm sefer numaraları ile doldur. duzey=0 ayarla, musait_duzey listesini, S kümesi uzunluğunda 0 ile doldur. musait_duzey=[0,0,...,0]

Adım 2: M kümesini doldur. (Mevcut noktadan gidilebilecek seferler, L listesinde olmayan.)

Adım 3: M kümesinin uzunluğu > 0 ise;

Adım 3.1: L listesine, M[musait_duzey_{duzey}] elemanını yerleştir.

Adım 3.2: duzey ve musait_duzey_{duzey} 1 artır.

Adım 3.3: Adım 5'e geç.

Adım 4: M kümesi uzunluğu = 0 veya musait_duzey_{duzey} > M kümesi uzunluğu ise;

Adım 4.1: duzey 1 eksilt, mevcut düzey sonrası tüm listedeki musait_duzey_{duzey} parametresini yeniden 0 yap. (Örneğin 3. düzey sonrasını musait_duzey=[1,2,1,0,...,0])

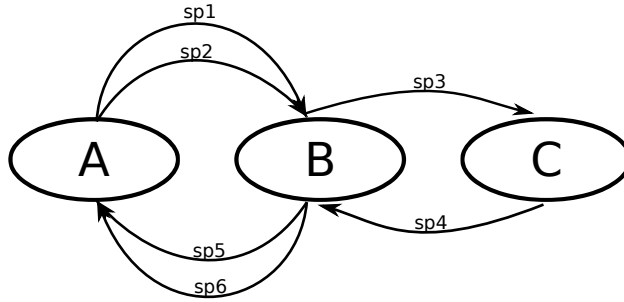
Adım 4.2: duzey seviyesinde atanan L elemanı varsa sil.

Adım 4.3: Aşama 2'e dön.

Adım 5: Eğer düzey değişkeni, S listesi uzunluğunda ise, kromozom oluşturulmuştur, DUR. Değil ise aşama 2'e dön.

Kromozom oluřturma ařaması ok nemlidir. Oluřturulma teknięi bu alıřmaya zgüdür. Arařtırmacılara kolaylık olması iin, kromozom oluřturma prosedürü kk bir rnek zerinden de anlatılacaktır.

Altı adet seferi bulunan bir veride, kromozom uzunluęu da 6 olacaktır. Őekil 11'de yaylar zerinde seferler grlmektedir.



Őekil 11: 3 Őehirli 6 seferli rnek soru

$i = 0$, iterasyon sayısı.

$musait_duzey = [0 , 0 , 0 , 0 , 0 , 0]$

$duzey=0$;

$S: \{ sp1, sp2, sp3, sp4, sp5, sp6 \}$

$M: \{ sp1, sp2, sp3, sp4, sp5, sp6 \}$

$musait_duzey_{duzey} = 0$ olduęu iin, M listesinden 0'ıncı eleman alınır.¹

$L: \{ sp1 \}$

$musait_duzey_{duzey} += 1$

$duzey += 1$

$i=1$

$musait_duzey = [1 , 0 , 0 , 0 , 0 , 0]$

$duzey=1$;

$S: \{ sp1, sp2, sp3, sp4, sp5, sp6 \}$

$L: \{ sp1 \}$, sp1 listesindeki son elemana gre Msaitler listesi

$M: \{ sp3, sp5, sp6 \}$

$musait_duzey_{duzey} = 0$ olduęu iin, M listesinden 0'ıncı eleman alınır.

1 Bilgisayar programlamada ilk terim 0 olarak alındıęı unutulmamalıdır.

L: {sp1,sp3}
musait_duzey_{duzey} += 1
duzey +=1

i=2
musait_duzey = [1 , 1, 0, 0, 0, 0]
duzey=2;
S: { sp1,sp2,sp3,sp4,sp5,sp6}
L: {sp1,sp3} , sp3 listesindeki son elemana göre Müsaitler listesi
M: {sp4}
musait_duzey_{duzey} = 0 olduğu için, M listesinden 0'inci eleman alınır.
L: {sp1,sp3,sp4}
musait_duzey_{duzey} += 1
duzey +=1

i=3
musait_duzey = [1 , 1, 1, 0, 0, 0]
duzey=3;
S: { sp1,sp2,sp3,sp4,sp5,sp6}
L: {sp1,sp3,sp4} , sp4 listesindeki son elemana göre Müsaitler listesi
M: {sp5,sp6}
musait_duzey_{duzey} = 0 olduğu için, M listesinden 0'inci eleman alınır.
L: {sp1,sp3,sp4,sp5}
musait_duzey_{duzey} += 1
duzey +=1

i=4
musait_duzey = [1 , 1, 1, 1, 0, 0]
duzey=4;
S: { sp1,sp2,sp3,sp4,sp5,sp6}
L: {sp1,sp3,sp4,sp5} , sp5 listesindeki son elemana göre Müsaitler listesi

$M: \{sp2\}$

$musait_duzey_{duzey} = 0$ olduğu için, M listesinden 0'ncı eleman alınır.

$L: \{sp1, sp3, sp4, sp5, sp2\}$

$musait_duzey_{duzey} += 1$

$duzey += 1$

$i=5$

$musait_duzey = [1, 1, 1, 1, 1, 0]$

$duzey=5;$

$S: \{ sp1, sp2, sp3, sp4, sp5, sp6 \}$

$L: \{sp1, sp3, sp4, sp5, sp2\}$, sp2 listesindeki son elemana göre Müsaitler listesi

$M: \{sp6\}$

$musait_duzey_{duzey} = 0$ olduğu için, M listesinden 0'ncı eleman alınır.

$L: \{sp1, sp3, sp4, sp5, sp2, sp6\}$

$musait_duzey_{duzey} += 1$

$duzey += 1$

$i=6$

$musait_duzey = [1, 1, 1, 1, 1, 1]$

$duzey=6;$

$duzey$ ve S küme uzunluğu eşit olduğuna göre işlem durdurulur. Kromozom oluşturulmuştur.

K:

Sp1	Sp3	Sp4	Sp5	Sp2	sp6
-----	-----	-----	-----	-----	-----

Burada tüm işlemler sorunsuz olarak ilerlemiştir ve ilk başlanılan noktaya geri dönmüştür. Bu nedenle $musait_duzey$ değişkeninin faydası görülmemiştir. İlerleyen iterasyonda bir çıkmaza girildiği durumda 3. $duzey$ e geri döndüğü varsayılınsın. Bu durumda iterasyon şu şekilde ilerleyecektir;

3. $duzey$ e dönüş için öncelikle $musait_duzey$ için geri kalan değerler 0 olarak ayarlanması gerektiği bahsedilmiştir.

$i=8$

$musait_duzey = [1, 1, 1, 1, 0, 0]$

$duzey=3;$

$S: \{ sp1, sp2, sp3, sp4, sp5, sp6 \}$

$L: \{ sp1, sp3, sp4 \}$, $sp4$ listesindeki son elemana göre Müsaitler listesi

$M: \{ sp5, sp6 \}$

$musait_duzey_{duzey} = 1$ olduğu için, M listesinden 1'inci eleman alınır.

$L: \{ sp1, sp3, sp4, sp6 \}$

$musait_duzey_{duzey} += 1$

$duzey += 1$

Böylece iterasyon ilerletilerek önceki çıkmaza girdiği yere bir daha uğramaz.

Daha sonraki aşama üretilen kromozomların çaprazlanması işlemidir.

7.1.2 Çaprazlama İşlemi

Çaprazlamanın hangi bireyler arasında olduğu, yeni üretilen nesil için çok önemlidir. Genel olarak her birey nüfus içindeki diğer bireylerle eşleştirilerek çaprazlama yapılır. Buradaki hedef, giderek daha iyi olan nesil içinde, iyi yönlerin yeni nesillere aktarılmasının sağlanmasıdır. Diğer bir yaklaşım ise nüfus içindeki en iyi birey ile diğerleri eşleştirilirler. Lokal en iyide kalmaya sebep olabilecek bu işlem, daha hızlı ve optimum sonuç bulmada iyidir. Bu çalışmada daha çok alternatif üreteceği düşüncesi ile her birey diğerleri ile çaprazlandığı yöntem seçilmiştir.

Çaprazlama işlemi için bu çalışmada tek noktadan çaprazlama tekniği kullanılmıştır. Fakat öncüllük ilişkileri nedeniyle başta anlatıldığı gibi iki kromozomdaki genler kolaylıkla değiştirilemezler. Algoritma aşağıdaki adımlardan oluşmaktadır:

Adım 1: Kromozom uzunluğunu aşmayan rassal bir sayı üret. Birinci çaprazlama noktasına kadar ilk üretilecek kromozomun birinci kısmına koy.

Adım 2: İkinci kromozomun çaprazlama noktasından sonraki ikinci kısmını sırayla ilk üretilecek kromozoma yerleştirmek için bir değişkene ata (M).

Adım 3: M kümesinde olup, üretilen kromozomun birinci kısmında olanlar varsa, M listesinden sil.

Adım 4: Oluşan kromozomun birinci kısmının sonunda yer alan seferi al ve sıradakileri (E) muhtemel listesine ata.

Adım 5: Eğer E listesinde, M listesinden birden fazla eleman varsa, M listesindeki önceliğe göre koy. M listesinden atanamı çıkar.

Adım 6: Oluşan kromozomun uzunluğu sefer uzunluğuna erişti ise algoritmayı durdur. Aksi halde Adım 4'e dön.

Kısa örnek ile yukarıda verilen adımlar örnek üzerinde gösterilerek, çaprazlama prosedürü anlatılacaktır. Çaprazlama sonrası aktarılan kısımların görülebilmesi için K_1 koyu verilmiştir.

$K_1: \{\text{sp1,sp3,sp4,sp5,sp2,sp6}\}$

$K_2: \{\text{sp1,sp3,sp4,sp6,sp2,sp5}\}$

r (0-5) arasında rassal bir sayı olsun.

r=2 ise;

$\mathcal{C}_1: \{\text{sp1,sp3,sp4, sp6,sp2,sp5}\}$

$\mathcal{C}_2: \{\text{sp1,sp3,sp4, sp5,sp2,sp6}\}$

şeklinde çocuk kromozomlar üretilirler.

r=3 seçilse idi;

$K_1: \{\text{sp1,sp3,sp4,sp5,sp2,sp6}\}$

$K_2: \{\text{sp1,sp3,sp4,sp6,sp2,sp5}\}$

$\mathcal{C}_1: \{\text{sp1,sp3,sp4,sp5,sp2,sp6}\}$

$\mathcal{C}_2: \{\text{sp1,sp3,sp4,sp6, sp2,sp5}\}$

görüldüğü üzere, çaprazlanan noktadan öncesinde yeni gelen genlerdeki seferler zaten bulunmaktaydı, bu seferler çaprazlanan kromozom parçasının önceliğine göre yerleştirilmişlerdir.

7.1.3 Mutasyon

Mutasyonda rastgele alınan iki rassal sayı üzerindeki iki genin değiştirilmesi işlemi yapılır. Fakat bu işlem, bir noktadan herhangi bir diğerine gidiş olmaması ihtimali nedeniyle her zaman kabul edilebilir sonuç vermeyecektir. Literatürde sonuç kabul edilebilir değil ise, işlem eski genlerle devam etmek şeklinde ilerlemektedir. Fakat bu da mutasyonun zaten az olan oranının daha da az bir oranda (bilinmeyen bir oranda) gerçekleşmesine sebep olacaktır. Bu nedenle bu çalışmada her şart altında mutasyonu

mümkün kılacak şekilde, rastgele alınan sayıya denk gelen genin mümkün olan en sona (kromozomda en sağa) yerleştirilmesi yöntemini kullanır.

Mutasyon için, kromozom uzunluğu ile 1 arasında rastgele bir rakam alınır. Bu rakam ile mutasyon yapılacak gen bulunur. Mutasyon yapılacak gen üzerindeki sefer numarasına bakılır. Bu noktadan sonraki kısım bir değişkene atanır. (M) Kromozom oluşturma aşamasındaki gibi sırayla kromozom yeniden doldurulur. Fakat bu aşamada M listesinde bulunan mutasyona uğrayacak gen, listenin her zaman sonunda tutulur.

Kısa örnek üzerinden açıklamak gerekirse;

K: {sp1,sp5,sp2,sp3,sp4,sp6}

r=1 olsun, sp5 seferi en sağa kaydırılmak üzere;

K: {sp1,,,,,} kalan kısım L: { sp2,sp3,sp4,sp6 }, alternatifler L listesine göre tercih edilecektir.

sp1 sonrası M tablosu oluşsun > M={sp3,sp5,sp6}

M tablosu sıralanır ve M={sp3,sp6,sp5} sona eklenir.

K: {sp1,sp3,,,,} kromozomu oluşur. İşlem hızlıca gösterilirse;

M: {sp4}

K: {sp1,sp3,sp4}

M: {sp5,sp6} => sıralanır M: {sp6,sp5}

K: {sp1,sp3,sp4,sp6}

M: {sp2}

K: {sp1,sp3,sp4,sp6,sp2}

M: {sp5}

K: {sp1,sp3,sp4,sp6,sp2,sp5} mutasyona uğramış kromozom oluşturulur.

Önceki hali ise; K: {sp1,sp5,sp2,sp3,sp4,sp6}

7.1.4 Genetik Algoritma Parametreleri Belirleme

Genetik algoritmanın parametrelerinin belirlenmesi için kesin bir yöntem yoktur. Ancak bazı tavsiyeler yapılmaktadır (Eiben ve Smit, 2011). Genetik algoritmanın parametreleri;

Nüfus sayısı: Kromozom havuzunun büyüklüğüdür.

İterasyon sayısı: Daha çok ihtimalin denenmesine imkan tanımakla birlikte sonlandırma kriteri olarak da kullanılabilir.

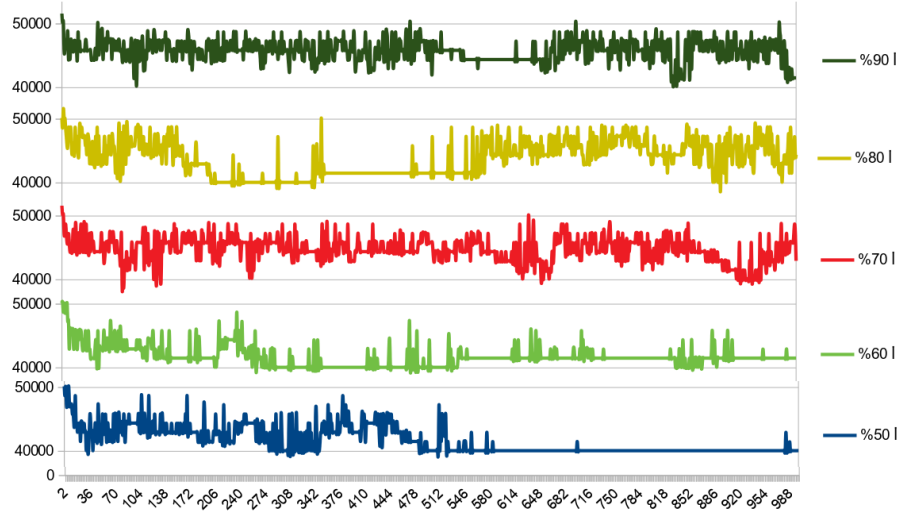
Çaprazlama oranı: Nüfusun daha iyi bireyler üretmesi için yüksek seçilmelidir. Fakat iyi olan nüfus içindeki bireylerin de yok olmasına neden olabilir.

Mutasyon oranı: Nüfus içinde zamanla aynılaştan kromozom kümelerini alternatifleri oluşturarak yerel en iyiden uzaklaştırmaya yarayan en önemli parametredir. Yüksek oranda uygulanması nüfusun daha iyiye gitme ihtimalini de bozabilir.

Seçim kriteri: Çaprazlanan kromozomların hangi kritere göre yeni nesile aktarılmasının seçimidir. Seçim kriterinin de kendine ait parametreleri olabilir. Örneğin turnuva seçim yöntemine göre nüfus içinden kaç adet rastgele kromozom çekileceği bir başka parametredir.

Bütün bu parametreler, birbirleriyle ilişkilidir. O nedenle hepsinin değiştirildiği birkaç kombinasyon denenecektir. Her kombinasyon 10'ar kez denenecektir. Bu parametrelerin genellenmesi mümkün değildir ve bu problem türüne uygun olacağı bilinmelidir. Daha önce bahsedildiği üzere, tek geliş-gidiş olan düğümler için çözümdeki sonuçlar çok farklı olmamaktaydılar. Bu nedenle parametreleri denerken, çok fazla alternatif oluşması açısından orjinal soruda her düğüm için en az 3 adet gidiş geliş eklenmiş ve her eklenen sefer bir önceki seferden 2 saat daha erken kalkış saatine ayarlanarak veri kümesi çok sayıda alternatif barındırmıştır. Toplam sefer sayısı 82 adettir. EK - 6'da tüm veriler bulunmaktadır. Buradan elde edilen parametreler, hem bu konuda çalışacaklar için başlangıç değerleri olacak, hem de orijinal, gerçek hayat verilerinden oluşan problemi (EK - 4) çözmek için yol gösterecektir.

Grafik 1: Çaprazlama oranı parametresi etkisi, (%50-%90)



Parametrelerin algoritma üzerinde nasıl etki yaptığının görülmesi için çaprazlama oranı/iterasyon sayısı grafik olarak verilmiştir. Grafiği oluşturmak için EK - 4'te yer alan 34 seferlik problem kullanılmıştır. Grafik 1'de genetik algoritma ile 1000 iterasyon sonucunda nüfus içindeki en iyi bireylerin farklı çaprazlama oranlarına karşı, zamanla değişimini göstermektedir. Aylak süre dahil toplam tur süresi 38000 ile 56000 dakika arasında dağılmaktadır. Toplam tur süresinin en az olması, en az aylak süreyi de sağlamaktadır. Buna göre grafik incelendiğinde tüm işlemlerde bir iyileşme görülmektedir. Yüksek orandaki çaprazlama oranı çok farklı bireyler üretmekte yardımcı olurken, nüfusun daha iyiye gitmesine de engel olabilmektedir. %50'lik çaprazlama oranı nüfusu hızlı şekilde iyiye doğru götürmüştür fakat en iyinin bulunmasını sağlayacak çeşitliliği %70 oranındaki sonuçlara kıyasla üretmemiş olduğu görülmektedir.

Literatürde yapılan araştırmalara bakıldığında, tam anlamıyla her problem türü için geçerli parametre ve belirleme yöntemi yoktur. Bu durumda tüm kombinasyonların denenmesi gerekmektedir. Bunun için tam faktoriyel deney tasarımı yapmak gerekmektedir. Tablo 17'de test edilecek algoritma parametreleri görülmektedir. Buna göre 3 seviye ve 5 faktör bulunmaktadır. Tam faktoriyel analiz yapılırsa toplam 243 adet deney yapılması gerekmektedir. Bunun yerine daha kısa ve kolay bir deney tasarımı açısından Taghuchi yöntemi kullanılmıştır. Böylece 27 deney yapılması yeterlidir.

Daha önce yapılan bir başka çalışmada tam faktoriyel deney tasarımı ile benzer sonuçlar ürettiği de gösterilmiştir (Sun, 2007).

Tablo 17: Denenecek genetik algoritma parametreleri

Yöntem	Seviye 1	Seviye 2	Seviye 3
Nüfus Sayısı (N)	50	100	150
Çaprazlama Oranı (Ç)	%50	%70	%90
Mutasyon Oranı (M)	%5	%10	%15
Seçim Kriteri (Turnuva) (T)	10	20	30
İterasyon Sayısı (İ)	500	1000	1500

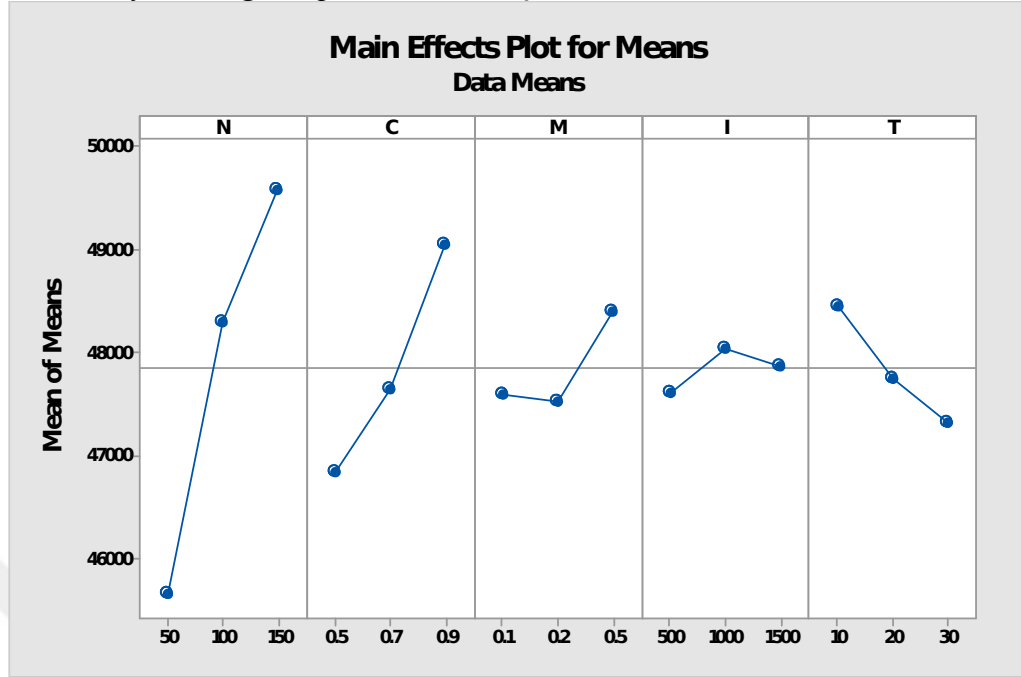
Parametrede bulunan seçim kriteri için alternatifler yerine Turnuva yöntemi tek olarak seçilmiştir. Turnuva yöntemi, nüfus içinde bulunan diğer ihtimalleri alması bakımından diğer yöntemlere göre daha iyidir. Turnuva seçim için rassal seçilen 10, 20 veya 30 adet kromozom arasından en iyisini gelecek nüfusa aktarmaktadır. Doğal olarak turnuva sayısı ne kadar yüksek ise, daha iyi olanları seçme ihtimali de o kadar yüksek olacaktır.

Deney tasarımında Taguchi yöntemi için nüfus ortalamasının en küçük olması hedeflenmiştir. Çünkü yüksek işlem yapmak yerine daha az işlem ile sonuca ulaşmak, işlem gücü ve zamanının ekonomik kullanılmasını mümkün kılar. Sonuçlar minitab istatistik programı ile oluşturulmuştur. Sinyal gürültü oranları yerine, ortalamaların tepki tablosu verilmiştir.

Tablo 18: Taguchi tepki tablosu, parametrelerin önem derecesi

Seviye	N	Ç	M	I	T
1	45649	46839	47601	47610	48448
2	48301	47639	47520	48043	47755
3	49579	49051	48408	47876	47326
Delta	3930	2211	888	433	1122
Önem	1	2	4	5	3

Grafik 2: Taguchi parametre sonuçları



Deney tasarımına ait tüm sonuçlar ek EK - 7'de bulunabilir. Parametrelerin önemine bakıldığında en büyük etken nüfus sayısı olarak bulunmuştur. Parametrelerin doğrusal modele uyumu ve varyans analizine bakılarak, etkileri hakkında yorum yapılabilir. Anlamlılık düzeyi olarak %10 seçimi (minitab, 2018) bir parametrenin etkisinin olup olmadığı hakkında karar vermemizi kolaylaştıracaktır. Bu sayede analiz sonucu aynı zamanda daha az parametre ile çalışmaya da imkan vermektedir.

Tablo 19: Parametrelerin ANOVA sonuçları

Source	DF	Seq SS	Adj SS	Adj MS	F	P
N	2	2.44679	2.44679	1.2234	69.95	0
C	2	0.76905	0.76905	0.38453	21.99	0
M	2	0.15951	0.15951	0.07975	4.56	0.027
I	2	0.03381	0.03381	0.01691	0.97	0.401
T	2	0.19088	0.19088	0.09544	5.46	0.016
Residual Error	16	0.27982	0.27982	0.01749		
Total	26	3.87987				

Sonuçlara bakılırsa iterasyon sayısının nüfus ortalamasının düşmesinde katkısı olmadığı görülmektedir. Bunun birkaç nedeni olabilir. Şüphesiz daha uzun bir iterasyon

ile genetik algoritma, nüfusu daha iyi sonuçlara götürecektir. Bu etki Grafik 2’de de görülebilir. Bu durumda hesap zamanı da dikkate alınması nedeniyle, iterasyon sayısının en küçük rakam olan 500 olanı seçmek doğru ve kolay bir yaklaşım olacaktır.

Nüfus sayısının etkisinin birinci etken olarak çıkması çok tesadüf değildir. Çünkü nüfusun kalabalık olup olmaması, farklı çözümleri içinde bulundurması açısından önemlidir. Ayrıca iyileşmesi için daha çok iterasyona ihtiyaç vardır. En düşük işlem süresi istemesi ve hızlı sonuç üretebilmesi için nüfus sayısı 50 olarak seçilecektir. İkinci önem sırasında olan çaprazlama oranı nüfus içinde farklılıkların oluşmasına imkan vermesi açısından etkilidir. Parametre değeri olarak 0.5 oranı seçilecektir. Mutasyon için sonuçlar daha kolay görülmektedir. 0.2 oranı ideal görünmektedir. Turnuva sayısı önceden bahsedildiği üzere büyük olması, beklenen sonucu vermiştir. Bu çalışmada da 30 seçilecektir.

Özet olarak genetik algoritma parametreleri, Tablo 20’teki gibi seçilmiştir.

Tablo 20: Seçilen parametreler

Nüfus Sayısı (N)	50
Çaprazlama Oranı (Ç)	%50
Mutasyon Oranı (M)	%20
Seçim Kriteri (Turnuva) (T)	30
İterasyon Sayısı (İ)	500

7.2 PROBLEMİN ÇÖZÜMÜ

Algoritma Python dili ile yazılmıştır. Bilgisayar konfigürasyonu, 4 çekirdekli ve i5,i6300hq olup, herhangi bir komut seti veya gpu desteği kullanılmamıştır. Grafik arayüz için QT kütüphanesi seçilmiştir. Programın arayüzü ve rapor bilgileri EK - 8’de incelenebilir.

Parametreler belirlenirken nüfus oluşturma aşaması hepsi için aynı değerler ile yapılmıştır. Genetik algoritmanın iyi sonuçlar üretmesi nüfusun zenginliği/çeşitliliği ile doğru orantılıdır. Bu nedenle parametreler artık belirlendiğine göre, daha zengin nüfus oluşturulması için, sefer başlangıç noktaları ve sıradaki seçimleri rastgele olarak yapılacaktır.

34 seferlik gerçek hayat verilerinden alınan problem genetik algoritma uygulanarak çözülmüştür. 7.1.1 Kromozom Oluşturma başlığında da anlatıldığı üzere algoritmanın ürettiği tüm çözümler tek araçla, başladığı noktaya dönecek şekilde mümkün olan turlardan oluşmaktadır. Bu nedenle bu algoritmanın kullanımı, rastgele seçilmeden her zaman daha iyi sonuçlar verecektir. Ayrıca diğer bahsedilen (SEBAS-SEKAS) sezgisel metotlara göre de, ne olursa olsun mümkün sonuç üretebilecektir.

Çözüm algoritması 10 kez denenmiştir. Sonuçlar Tablo 21'de gösterilmektedir. Tüm sıralamalar EK - 9'da görülebilir.

Tablo 21: 10 deneyin sonuçları

Deneme No	Ortalama	Standart Sapma	Verimlilik Oranı	En Düşük Aylak Süre
1	17221.8	2738.8784858040112	61.44166666666667	13881
2	18892.2	2413.70805193999	61.44166666666667	13881
3	18690.6	2368.0886469893885	61.44166666666667	13881
4	17884.2	1959.4585374536332	61.44166666666667	13881
5	17509.8	1826.928175928124	61.44166666666667	13881
6	18489.0	2285.929132759806	61.44166666666667	13881
7	17682.6	1815.999295154064	59.078525641025635	15321
8	17711.4	2553.4736810862128	64.00173611111111	13881
9	18460.2	2742.6613644414715	61.44166666666667	13881
10	17970.6	2597.753614182845	61.44166666666667	13881

EK - 9'daki sıralamalara bakan araştırmacılar fark edeceklerdir ki, verimlilikleri aynı olmasına rağmen sıralamalar farklıdır.

Algoritma çalışma süresi 12 saniye olmakla birlikte, bilgisayardaki çekirdek sayısına bağlı olarak çoklu işlemciyi desteklemektedir. Bu nedenle 10 deney, $10 \times 12 = 120$ saniye değil, örneğin 4 çekirdekli bir bilgisayar için $120/4 = 30$ saniye sürmektedir.

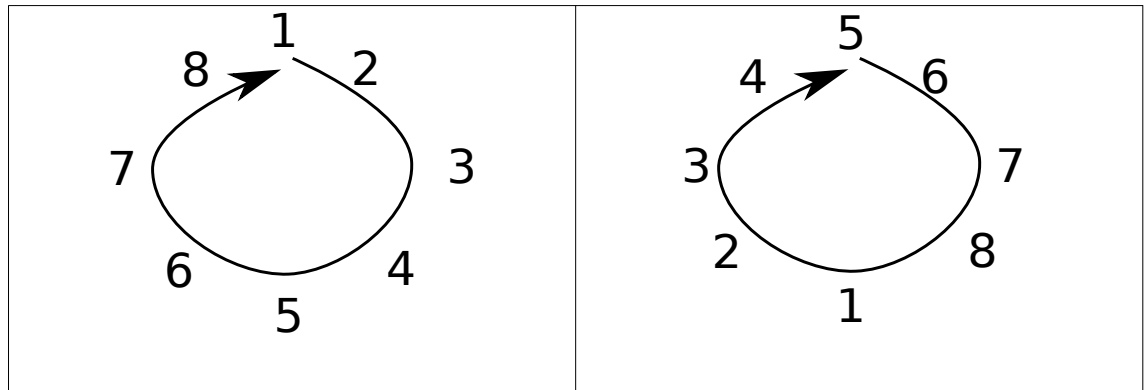
Daha önce geri izlemeli SEKAS yöntemi ile aylak süre 11001 dakika olmak üzere, verimlilik %66.78 olarak bulunmuştu. Fakat algoritmanın bu sonucu bulması 12 saniyeden çok daha uzun sürmüştü. Genetik algoritma için de çok fazla iterasyon sayısının verilmesi, daha iyi sonuçlara götürme ihtimalini içinde barındırmaktadır. Ayrıca başlangıç nüfus oluşturmasının sonuçlar üzerine etkili olduğundan bahsedilmişti. Bu nedenle aynı parametrelerle algoritma 100 kez tekrar edilmiş ve daha iyi sonuçlar

elde edilmiştir. Algoritmanın 10 yerine 100 kez tekrar edilmesinin bize verdiği avantaj, başlangıç kromozomlarının daha farklı olmasını sağlamasıdır.

Tablo 22: Çözüm sonucu elde edilen daha iyi sonuçlar

Verimlilik	Aylak Süre(dakika)	Ortalama (dakika)	Standart Sapma (dakika)
%64.00173611111111	12441	14255.4	1794.1729682502787
Sıra	sp2, sp20, sp3, sp22, sp5, sp30, sp17, sp23, sp7, sp4, sp8, sp31, sp32, sp19, sp33, sp16, sp9, sp18, sp34, sp24, sp10, sp25, sp11, sp26, sp12, sp27, sp13, sp28, sp15, sp29, sp14, sp21, sp1, sp6		
%66.78442028985508	11001	14485.8	3396.5689982686954
Sıra	sp24, sp10, sp25, sp11, sp29, sp15, sp30, sp17, sp26, sp9, sp18, sp32, sp19, sp33, sp16, sp12, sp20, sp1, sp6, sp2, sp27, sp13, sp28, sp14, sp21, sp3, sp22, sp5, sp23, sp7, sp4, sp8, sp31, sp34		

Bir başka fark edilen ilginç durum ise aynı verimliliğe sahip çok sayıda farklı sıralama oluşmasıdır. Bu sıralamalar incelendiğinde aslında bulunan sıraların farklı olmadığı görülmektedir. Çözüm sonucu istenen ilk noktaya geri dönüş şartı, bir daire oluşturmaktadır. Bu dairenin istenilen noktasından bir başka başlangıç noktası seçilebilir. Aslında sıralama aynı olmasına rağmen, başlangıç noktaları farklı görülecektir. Bu fark Şekil 12’de gösterilmiştir.



Şekil 12: Başladığı noktaya geri dönen, halka dizilişine sahip sıralama

İlk sıralama 1-2-3-4-5-6-7-8 şeklinde görülürken, ikinci sıralama 5-6-7-8-1-2-3-4 şeklinde görülmektedir. Bu bakımdan algoritma, çok sayıda başlangıç noktaları farklı ama aynı halka dizilişine sahip sıralamalar üretmektedir.

8 TARTIŞMA VE SONUÇ

Bu çalışmada literatürde olmayan yeni bir problem tanımı yapılmış ve çözümün önemi gösterilmiştir. Klasik yaklaşımda araçlar çalıştıkça, fazla yol kat ettikçe maliyet oluşturduğu düşüncesiyle, en kısa yol ve en düşük maliyet amaç fonksiyonu olmuştur. Bu çalışmadaki yaklaşım ise araçların çalıştığı müddetçe para kazandığı varsayımıyla araçların en çok çalışması veya en az aylak süresinin bulunmasını amaçlamaktadır. Bu nedenle araçların bazen daha uzun yola yönlendirilmeleri istenmektedir.

Çözüm oluşturmada genetik algoritma kullanılmıştır. Genetik algoritmanın ne olursa olsun kabul edilebilir rota oluşturmasını mümkün kılan sıralama oluşturması, yazılan algoritmanın güçlü yanlarından biridir. Sıralamayı tutan kromozom oluşturma prosedürü diğer araştırmacıların da çalışmalarında kullanabilmeleri için açıkça anlatılmıştır. Çaprazlama ve mutasyon işlemleri öncelik ilişkileri bulunan işlere benzer olarak kullanılmıştır.

Anket yoluyla toplanan 34 seferden oluşan gerçek hayat verilerine genetik algoritma uygulanarak rotalama yapılmıştır. Sonuçlar incelendiğinde bulunan en kötü sıralama değeri ve en iyi sıralama değeri arasında 21600 dakika gibi büyük bir aylak süre farkı bulunmaktadır. Bu fark 15 güne eşittir. Bunun sonucu iyi ve kötü sonuç arasında 15 araç tasarrufu anlamına gelmektedir. Çıkan en iyi sonuç ile araçların %66,78 oranında çalışabileceği gösterilmiştir. %66,78'lik verimlilik elde edilen tur 23 günde tamamlanmaktadır. 34 seferin yer aldığı bu problem için tüm seferlerin zamanında gerçekleşmesini sağlayacak en az araç sayısının 23 olduğu da sonuç olarak elde edilmektedir.

Gerek ülkemizde, gerekse dünyadaki tüm sefer saatleri önceden belirlenmiş araçlar için yapılan bu çalışma büyük katkı sağlayacaktır. Çalışmanın hedefi olan en çok kar yanında, daha az araç kullanılması gibi nedenlerle, daha az kirlilik hedefleri de dolaylı olarak sağlanmaktadır. Ayrıca işletmelerin ellerinde bulundurdukları araçların sigortaları ve bakım masrafları da düşecektir.

Çalışmada verilmek istenen yaklaşım insansız araçların yönetimi için daha çok uygundur. Çünkü araç çizelgeleme problemlerinde, araç ve filo, sürücü, nöbet gibi

durumlar da düşünölmelidir. Fakat sürücüsüz bir araç için, en verimli kullanım ön plana çıkacaktır. Ayrıca otonom araçlar kişilerin randevu saatine uymak zorundadır. Bu nedenle çalışmaya konu olan problemin ve sunulan çözüm yönteminin gelecek yıllarda önemi daha da iyi anlaşılacaktır.

Son olarak gelecekte yapılacak çalışmalar için bazı önerilerde bulunmak uygun olacaktır. Genetik algoritmanın daha etkili parametrelerinin bulunması için başka testler yapılabilir. Burada Taguchi'nin sunduğu deney tasarımı çalışmasında elde edilen parametreler bize bir fikir vermekle beraber kesin en iyi sonuçtur demek yanlış olacaktır.

Yeni bir problem türü olması nedeniyle, ARP problemlerinde yer alan kısıtlar bu problem türüne de uygulanarak, diğer gerçek hayat senaryoları içeren problemler türetilir. Örneğin; kapasitelerin dikkate alınması, müşteri taleplerinin değişkenliği, varış sürelerini etkileyebilecek yol durumlarının dikkate alınması ve stokastik verilerin kullanılması gibi durumlar dikkate alınarak problem yeniden çözülebilir. Bunun yanında birden fazla araç ile çözüm denenebilir. Alt turların oluşmasına izin verilmesi sonucu muhtemelen daha verimli rotalama mümkün olacaktır.

EKLER

EK - 1 Cordeau problem türü

Cordeau'nun problem türü için veriler ve çözümler aşağıdaki gibi tanımlanmıştır.

İlk satır sorunun tipini belirtmektedir.

type m n t

- type:
 - 0 (ARP)
 - 1 (P ARP)
 - 2 (ÇD ARP)
 - 3 (TD ARP)
 - 4 (ARPZP)
 - 5 (P ARPZP)
 - 6 (ÇD ARPZP)
 - 7 (TD ARPZP)
- m: araç sayısı
- n: müşteri sayısı
- t: gün sayısı (P ARP), depolar (ÇD ARP) veya araç tipleri (TD ARP)

Sonraki t kadar satır her güne ait şu bilgileri içerir;

D Q

D: Rotanın maksimum süresi

Q: Aracın maksimum yükü

Sıradaki satırlar, her müşteri için;

i x y d q f a list e l

- i: müşteri numarası
- x: x koordinatı
- y: y koordinatı

- d: servis süresi
- q: talep
- f: uğrama sıklığı
- a: muhtemel uğrama kombinasyon sayısı
- list: tüm muhtemel uğrama kombinasyon sayısı
- e: zaman penceresinin başlangıcı
- l: zaman penceresinin sonu

Tüm uğrama kombinasyonları periyoda göre ikili sistemle gösterilmiştir. Örneğin 5 günlük bir periyodu varsa kod üzerinde 01010 olarak gösterilir ve soldan sağa doğru 2. ve 4. günde uğradığını gösterir.

Not: ÇDARP soruları için satırlar 1 den $n+t$ 'ye kadar gitmektedir. Ve en sondaki t girdisi t deposuna aittir. Diğer ARP, PARP ve ÇDARP soruları için satırlar 0'dan n'ye kadar gitmektedir ve ilk girdi tek depoyu göstermektedir.

EK - 2 Solomon problem türü

Solomon'un problem türleri

İlk satır problemin adını belirtir.

K Q

K: araç sayısı

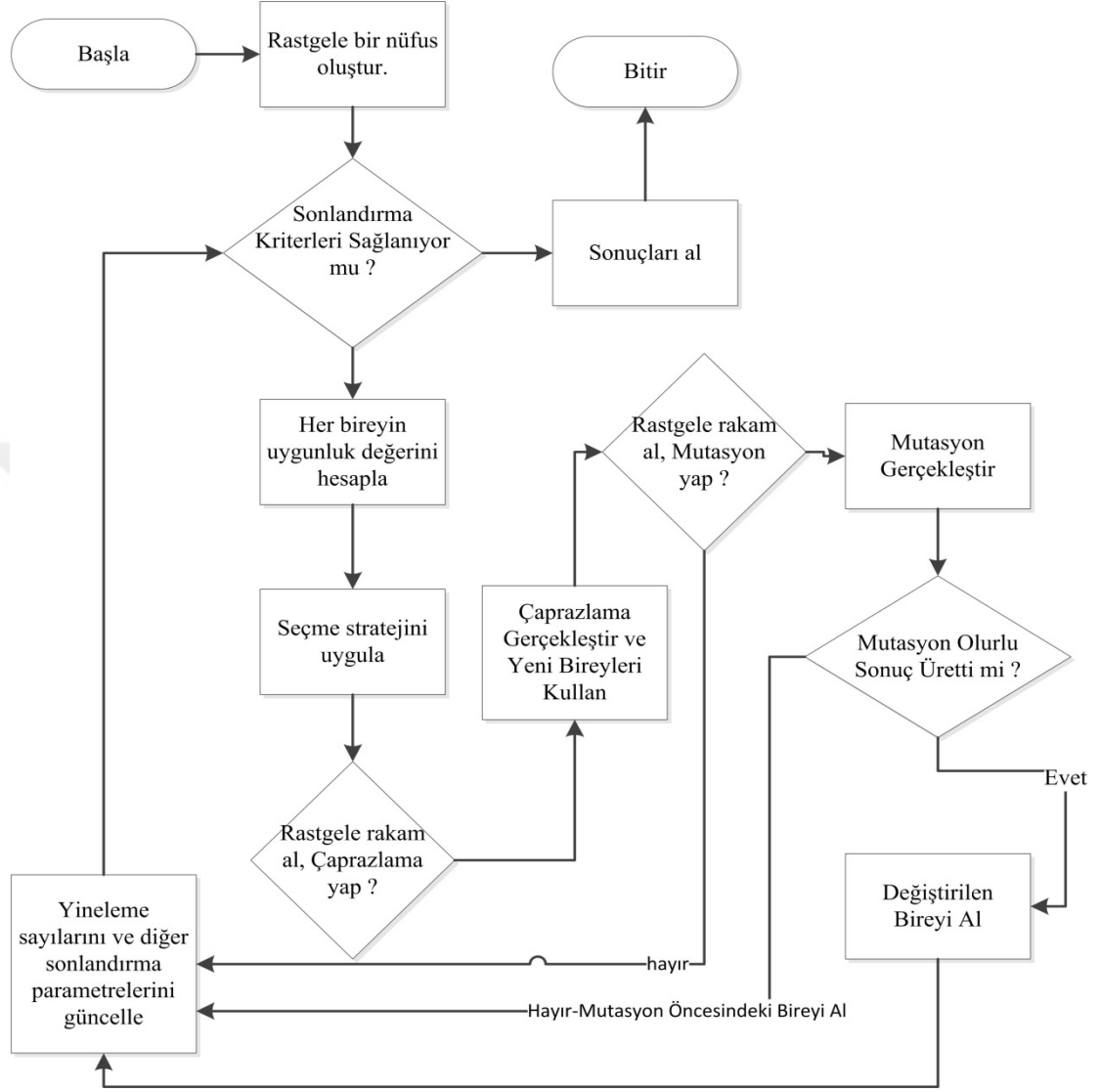
Q: kapasitesi

Kalan satırlar şu şekilde bilgiler verir;

i x y q e l d

- i: müşteri numarası
- x: x kordinatı
- y: y kordinatı
- q: talep
- e: hazır olma zamanı
- l: bitiş zamanı
- d: servis süresi

EK - 3 Temel Genetik Algoritma Çizelgesi



EK - 4 34 Sefere ait problem verileri

Sefer Numarası	Kalkış Noktası	Variş Noktası	Kalkış Saati	Variş Saati
sp1	ŞAHİN	TURNA	22:00	14:50
sp2	ŞAHİN	DOĞAN	10:00	12:19
sp3	ŞAHİN	DOĞAN	12:00	12:19
sp4	KARTAL	KIRLANGIÇ	21:30	13:50
sp5	KANARYA	DOĞAN	11:30	08:34
sp6	TURNA	ŞAHİN	19:00	14:50
sp7	KIRLANGIÇ	KARTAL	19:00	13:50
sp8	KIRLANGIÇ	DOĞAN	10:00	08:05
sp9	KEKLİK	BÜLBÜL	17:00	17:50
sp10	KEKLİK	DOĞAN	05:00	08:05
sp11	KEKLİK	DOĞAN	23:30	08:15
sp12	KEKLİK	DOĞAN	12:30	08:05
sp13	KEKLİK	DOĞAN	08:30	08:05
sp14	KEKLİK	DOĞAN	10:30	08:05
sp15	KEKLİK	DOĞAN	15:30	08:05
sp16	BÜLBÜL	KEKLİK	17:00	17:50
sp17	BÜLBÜL	DOĞAN	21:30	11:19
sp18	BÜLBÜL	ATMACA	15:00	15:50
sp19	BÜLBÜL	ATMACA	07:00	15:50
sp20	DOĞAN	ŞAHİN	23:00	10:50
sp21	DOĞAN	ŞAHİN	20:00	11:50
sp22	DOĞAN	KANARYA	01:00	08:50
sp23	DOĞAN	KIRLANGIÇ	12:00	07:34
sp24	DOĞAN	KEKLİK	16:00	08:00
sp25	DOĞAN	KEKLİK	14:00	08:00
sp26	DOĞAN	KEKLİK	09:00	07:50
sp27	DOĞAN	KEKLİK	00:00	08:00
sp28	DOĞAN	KEKLİK	18:00	08:00
sp29	DOĞAN	KEKLİK	10:00	08:00
sp30	DOĞAN	BÜLBÜL	09:00	11:50
sp31	DOĞAN	ATMACA	19:00	06:15
sp32	ATMACA	BÜLBÜL	10:30	15:50
sp33	ATMACA	BÜLBÜL	00:00	15:50
sp34	ATMACA	DOĞAN	08:30	06:04

EK - 5 Seferler Arası Aylak Sürelerin Dağılımı (Dakika)

Sefer No	sp1	sp2	sp3	sp4	sp5	sp6	sp7	sp8	sp9	sp10	sp11	sp12	sp13	sp14	sp15	sp16	sp17	sp18	sp19	sp20	sp21	sp22	sp23	sp24	sp25	sp26	sp27	sp28	sp29	sp30	sp31	sp32	sp33	sp34	
sp1	X	-	-	-	-	370	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
sp2	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	41	1301	161	821	1061	941	641	101	1181	701	641	1241	-	-	-	
sp3	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1361	1181	41	701	941	821	521	1421	1061	581	521	1121	-	-	-	
sp4	-	-	-	X	-	-	460	1360	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
sp5	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	176	1436	296	956	1196	1076	776	236	1316	836	776	1376	-	-	-	
sp6	730	10	130	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
sp7	-	-	-	760	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
sp8	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	295	115	415	1075	1315	1195	895	355	1435	955	895	55	-	-	-	
sp9	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	370	640	250	1210	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
sp10	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	595	415	715	1375	175	55	1195	655	295	1255	1195	355	-	-	-	
sp11	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	915	735	1035	255	495	375	75	975	615	135	75	675	-	-	-	
sp12	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	145	1405	265	925	1165	1045	745	205	1285	805	745	1345	-	-	-	
sp13	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	385	205	505	1165	1405	1285	985	445	85	1045	985	145	-	-	-	
sp14	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	265	85	385	1045	1285	1165	865	325	1405	925	865	25	-	-	-	
sp15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	1405	1225	85	745	985	865	565	25	1105	625	565	1165	-	-	-	
sp16	-	-	-	-	-	-	-	-	370	1090	760	100	1300	1420	280	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
sp17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	851	671	971	191	431	311	11	911	551	71	11	611	-	-	-	
sp18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	220	1030	100	
sp19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	700	70	580	
sp20	730	10	130	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
sp21	850	130	250	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	-	
sp22	-	-	-	-	100	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	-	
sp23	-	-	-	-	-	-	1406	866	-	-	-	-	-	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	-	
sp24	-	-	-	-	-	-	-	-	1020	300	1410	750	510	630	930	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	-	
sp25	-	-	-	-	-	-	-	-	1140	420	90	870	630	750	1050	-	-	-	-	-	-	-	-	-	X	-	-	-	-	-	-	-	-	-	

EK - 6 82 Sefere ait problem verileri

Sefer Numarası	Kalkış Noktası	Variş Noktası	Kalkış Saati	Variş Saati
sp1	ŞAHİN	TURNA	22:00	14:50
sp2	ŞAHİN	DOĞAN	10:00	12:20
sp3	ŞAHİN	DOĞAN	12:00	12:20
sp4	KARTAL	KIRLANGIÇ	21:30	13:50
sp5	KANARYA	DOĞAN	11:30	08:00
sp6	TURNA	ŞAHİN	19:00	14:50
sp7	KIRLANGIÇ	KARTAL	19:00	13:50
sp8	KIRLANGIÇ	DOĞAN	10:00	08:00
sp9	KEKLİK	BÜLBÜL	17:00	17:50
sp10	KEKLİK	DOĞAN	05:00	08:00
sp11	KEKLİK	DOĞAN	23:30	08:00
sp12	KEKLİK	DOĞAN	12:30	08:00
sp13	KEKLİK	DOĞAN	08:30	08:00
sp14	KEKLİK	DOĞAN	10:30	08:00
sp15	KEKLİK	DOĞAN	15:30	08:00
sp16	BÜLBÜL	KEKLİK	17:00	17:50
sp17	BÜLBÜL	DOĞAN	21:30	11:20
sp18	BÜLBÜL	ATMACA	15:00	15:50
sp19	BÜLBÜL	ATMACA	07:00	15:50
sp20	DOĞAN	ŞAHİN	23:00	10:50
sp21	DOĞAN	ŞAHİN	20:00	11:50
sp22	DOĞAN	KANARYA	01:00	08:00
sp23	DOĞAN	KIRLANGIÇ	12:00	07:35
sp24	DOĞAN	KEKLİK	16:00	08:00
sp25	DOĞAN	KEKLİK	14:00	08:00
sp26	DOĞAN	KEKLİK	09:00	07:50
sp27	DOĞAN	KEKLİK	00:00	08:00
sp28	DOĞAN	KEKLİK	18:00	08:00
sp29	DOĞAN	KEKLİK	10:00	08:00
sp30	DOĞAN	BÜLBÜL	09:00	11:50
sp31	DOĞAN	ATMACA	19:00	06:15
sp32	ATMACA	BÜLBÜL	10:30	15:50
sp33	ATMACA	BÜLBÜL	00:00	15:50
sp34	ATMACA	DOĞAN	08:30	06:05
sp35	ŞAHİN	TURNA	20:00	14:50
sp36	TURNA	ŞAHİN	17:00	14:50
sp37	KIRLANGIÇ	KARTAL	17:00	13:50
sp38	KARTAL	KIRLANGIÇ	19:30	13:50
sp39	KANARYA	DOĞAN	09:30	08:00
sp40	DOĞAN	KANARYA	23:00	08:00
sp41	DOĞAN	KIRLANGIÇ	10:00	07:35
sp42	KIRLANGIÇ	DOĞAN	08:00	08:00
sp43	BÜLBÜL	KEKLİK	15:00	17:50
sp44	KEKLİK	BÜLBÜL	15:00	17:50
sp45	DOĞAN	BÜLBÜL	07:00	11:50

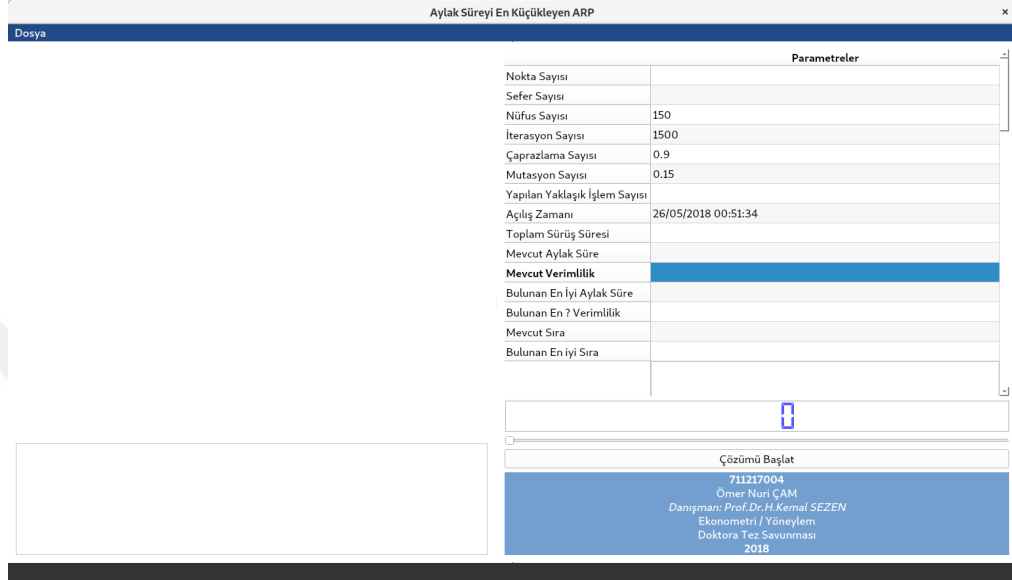
sp46	BÜLBÜL	DOĞAN	19:30	11:20
sp47	ŞAHİN	TURNA	18:00	14:50
sp48	TURNA	ŞAHİN	15:00	14:50
sp49	KIRLANGIÇ	KARTAL	15:00	13:50
sp50	KARTAL	KIRLANGIÇ	17:30	13:50
sp51	KIRLANGIÇ	KARTAL	13:00	13:50
sp52	KARTAL	KIRLANGIÇ	15:30	13:50
sp53	KANARYA	DOĞAN	07:30	08:00
sp54	DOĞAN	KANARYA	21:00	08:00
sp55	KANARYA	DOĞAN	05:30	08:00
sp56	DOĞAN	KANARYA	19:00	08:00
sp57	BÜLBÜL	KEKLİK	13:00	17:50
sp58	KEKLİK	BÜLBÜL	13:00	17:50
sp59	DOĞAN	BÜLBÜL	05:00	11:50
sp60	BÜLBÜL	DOĞAN	17:30	11:20
sp61	BÜLBÜL	KEKLİK	11:00	17:50
sp62	KEKLİK	BÜLBÜL	11:00	17:50
sp63	DOĞAN	BÜLBÜL	03:00	11:50
sp64	BÜLBÜL	DOĞAN	15:30	11:20
sp65	ATMACA	BÜLBÜL	08:30	15:50
sp66	BÜLBÜL	ATMACA	05:00	15:50
sp67	ATMACA	BÜLBÜL	06:30	15:50
sp68	BÜLBÜL	ATMACA	03:00	15:50
sp69	ATMACA	DOĞAN	06:30	06:05
sp70	DOĞAN	ATMACA	17:00	06:15
sp71	ATMACA	DOĞAN	04:30	06:05
sp72	DOĞAN	ATMACA	15:00	06:15
sp73	ATMACA	DOĞAN	02:30	06:05
sp74	DOĞAN	ATMACA	13:00	06:15
sp75	DOĞAN	KIRLANGIÇ	08:00	07:35
sp76	KIRLANGIÇ	DOĞAN	06:00	08:00
sp77	DOĞAN	KIRLANGIÇ	06:00	07:35
sp78	KIRLANGIÇ	DOĞAN	06:00	08:00
sp79	ŞAHİN	TURNA	16:00	14:50
sp80	TURNA	ŞAHİN	13:00	14:50
sp81	DOĞAN	ŞAHİN	21:00	10:50
sp82	ŞAHİN	DOĞAN	10:00	12:20

EK - 7 Taguchi Deney Tasarımı İçin Sonuçlar

N	C	M	I	T	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10	Ortalama
50	0.5	0.1	500	10	41193.4	39148.6	47155	46982.2	41827	44764.6	46175.8	38601.4	46175.8	45340.6	43736.44
50	0.5	0.1	500	20	46838.2	45110.2	46406.2	48652.6	40819	46060.6	46118.2	42921.4	45916.6	46463.8	45530.68
50	0.5	0.1	500	30	41049.4	45254.2	41711.8	44995	44966.2	39263.8	44505.4	44073.4	46665.4	39955	43243.96
50	0.7	0.2	1000	10	45052.6	45599.8	45945.4	47731	44563	46204.6	48537.4	48595	45628.6	45513.4	46337.08
50	0.7	0.2	1000	20	42979	46463.8	44937.4	46723	41366.2	45801.4	46521.4	43094.2	48191.8	43555	44963.32
50	0.7	0.2	1000	30	39897.4	44015.8	46406.2	47587	43555	44735.8	45801.4	44303.8	44102.2	46233.4	44663.8
50	0.9	0.5	1500	10	45254.2	48335.8	47068.6	46809.4	48422.2	48681.4	44822.2	48508.6	48278.2	48739	47491.96
50	0.9	0.5	1500	20	48623.8	48739	46492.6	48451	47529.4	47212.6	47702.2	48134.2	48220.6	47961.4	47906.68
50	0.9	0.5	1500	30	47817.4	47731	46953.4	46003	45772.6	48307	46147	48508.6	47183.8	45225.4	46964.92
100	0.5	0.2	1500	10	48983.8	47889.4	48033.4	49531	47399.8	49171	49214.2	48955	48407.8	49099	48668.44
100	0.5	0.2	1500	20	44894.2	46809.4	47126.2	42417.4	48134.2	46780.6	48407.8	47716.6	49027	46478.2	46779.16
100	0.5	0.2	1500	30	44649.4	47299	45023.8	42331	44001.4	47601.4	46967.8	45974.2	44188.6	47687.8	45572.44
100	0.7	0.5	500	10	49545.4	49862.2	48566.2	50452.6	48249.4	49646.2	49991.8	48537.4	49315	48911.8	49307.8
100	0.7	0.5	500	20	46924.6	46939	47846.2	47183.8	47731	48710.2	49415.8	46737.4	48868.6	46377.4	47673.4
100	0.7	0.5	500	30	48969.4	46795	47831.8	49343.8	49315	49156.6	49372.6	47169.4	46751.8	48350.2	48305.56
100	0.9	0.1	1000	10	49847.8	49905.4	50265.4	49876.6	49804.6	49905.4	49991.8	49300.6	49905.4	49718.2	49852.12
100	0.9	0.1	1000	20	47644.6	49761.4	49948.6	49747	47399.8	48292.6	49833.4	50078.2	48523	49631.8	49086.04
100	0.9	0.1	1000	30	48551.8	48983.8	49833.4	50107	49487.8	48739	49747	50063.8	50150.2	48955	49461.88
150	0.5	0.5	1000	10	50371	49612.6	50169.4	50371	50054.2	50294.2	50063.8	49852.6	50476.6	50150.2	50141.56
150	0.5	0.5	1000	20	49411	48537.4	50015.8	47596.6	50303.8	49823.8	48067	47366.2	47740.6	50063.8	48892.6
150	0.5	0.5	1000	30	48355	48460.6	48998.2	49103.8	49987	47961.4	48220.6	49737.4	50035	49007.8	48986.68
150	0.7	0.1	1500	10	49775.8	50620.6	50246.2	49939	50006.2	50015.8	49574.2	50207.8	49939	50035	50035.96
150	0.7	0.1	1500	20	48489.4	48835	49151.8	48643	50169.4	49958.2	47404.6	47779	47337.4	48892.6	48666.04
150	0.7	0.1	1500	30	48134.2	49958.2	48326.2	47788.6	47049.4	48758.2	49987	49036.6	49679.8	49267	48798.52
150	0.9	0.2	500	10	50495.8	50966.2	50553.4	50505.4	50054.2	50313.4	50486.2	50361.4	50553.4	50303.8	50459.32
150	0.9	0.2	500	20	49737.4	50217.4	50390.2	50227	50275	50467	50217.4	50495.8	50543.8	50380.6	50295.16
150	0.9	0.2	500	30	49055.8	49507	48873.4	50543.8	50284.6	49324.6	50198.2	50803	50495.8	50284.6	49937.08

EK - 8 ENAZARP Programı ve Kullanımı

Program python dilinde yazılmıştır. Standart kütüphaneler haricinde kütüphane kullanılmamıştır. Arayüz tasarımı için QT platformu tercih edilmiştir.

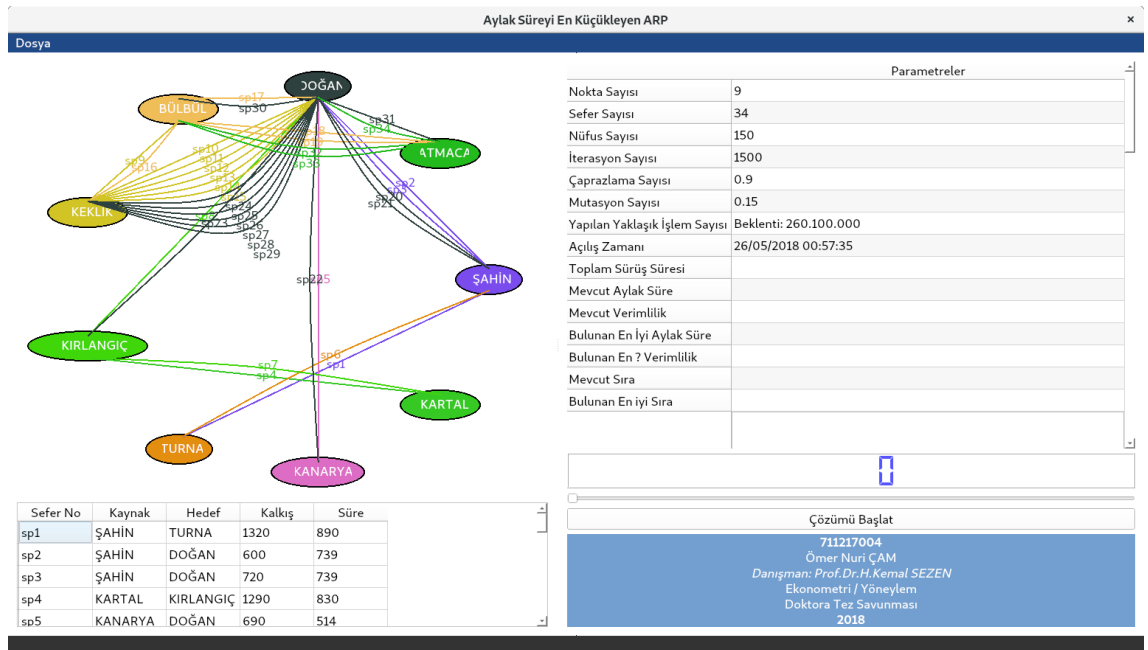


Parametreler	
Nokta Sayısı	
Sefer Sayısı	
Nüfus Sayısı	150
İterasyon Sayısı	1500
Çaprazlama Sayısı	0.9
Mutasyon Sayısı	0.15
Yapılan Yaklaşık İşlem Sayısı	
Açılış Zamanı	26/05/2018 00:51:34
Toplam Sürüş Süresi	
Mevcut Aylak Süre	
Mevcut Verimlilik	
Bulunan En İyi Aylak Süre	
Bulunan En ? Verimlilik	
Mevcut Sıra	
Bulunan En İyi Sıra	

Çözümü Başlat

711217004
Ömer Nuri ÇAM
Danışman: Prof.Dr.H.Kemal SEZEN
Ekonometri / Yöneylem
Doktora Tez Savunması
2018

Program açıldıktan sonra dosya menüsünden problem yüklenir. Problem veri biçimi csv olup, boşluklarla ayrılan verileri hafızaya yüklemektedir. Veriler yüklendikten sonra çözüme hazırlamak için kalkış ve varış saatleri dakika biçimine çevrilir. (Örneğin; saat 8:00 , 480 olarak , 22:10, 1330 olarak gösterilir.) Aynı zamanda yüklenen veriler çizim görselinin altında tablo şeklinde yer alır.



Sefer No	Kaynak	Hedef	Kalkış	Süre
sp1	ŞAHİN	TURNA	1320	890
sp2	ŞAHİN	DOĞAN	600	739
sp3	ŞAHİN	DOĞAN	720	739
sp4	KARTAL	KIRLANGIÇ	1290	830
sp5	KANARYA	DOĞAN	690	514

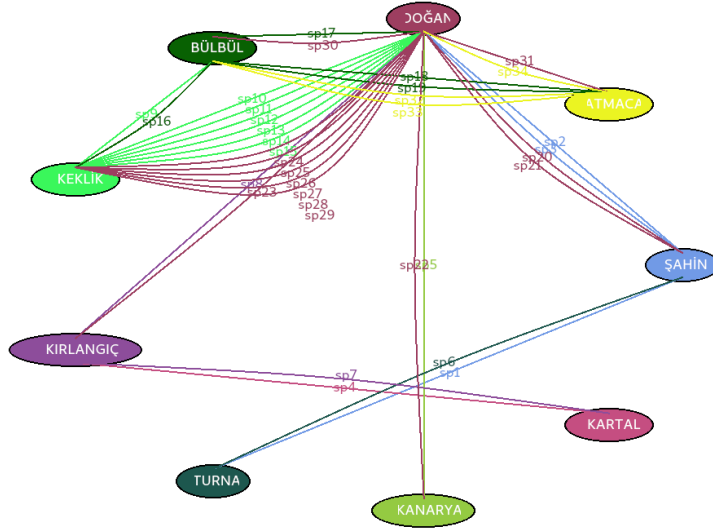
Çözümü Başlat

711217004
Ömer Nuri ÇAM
Danışman: Prof.Dr.H.Kemal SEZEN
Ekonometri / Yöneylem
Doktora Tez Savunması
2018

Sefer No	Kaynak	Hedef	Kalkış	Süre
sp1	ŞAHİN	TURNA	1320	890
sp2	ŞAHİN	DOĞAN	600	739
sp3	ŞAHİN	DOĞAN	720	739
sp4	KARTAL	KIRLANGIÇ	1290	830
sp5	KANARYA	DOĞAN	690	514
sp6	TURNA	ŞAHİN	1140	890

Veriler yüklendi.

Problemin yüklenmesi ile birlikte verilere ait istatistikler sağ tarafta görüntülenir. Yay ve düğüm biçiminde görsel oluşturulur. Her veri yüklenişinde düğümlerin renkleri rastgele seçilir. Eğer yay ve düğüm aynı renkte ise sefer yönünün aynı renkten diğer noktaya olduğunu gösterir. Böylece sefer yönlerini gösterecek ok uçları kullanılmayarak daha sade bir görsel elde edilmiştir.



Sağ tarafta tabloda genetik algoritmanın parametreleri değiştirilebilir. Çözüm başlatılması için başlat düğmesine basılır. İstenildiği zaman çalışma durdurulabilir. Çözüm başladıktan sonra butonun hemen üstünde iterasyon sayısını gösteren bir gösterge bulunur. Buradan kaçınıcı iterasyonda olduğu takip edilebilir. Çözüm devam ederken mevcut sıra ve en iyi sıra takip edilebilir.

Mevcut Sıra	sp10, sp25, sp11, sp26, sp12, sp27, sp13, sp28, sp14, sp21, sp2, sp20, sp1, sp6, sp3, sp22, sp5, sp29, sp9, sp18, sp32, sp19, sp33, sp16, sp15, sp30, sp17, sp23, sp7, sp4, sp8, sp31, sp34, sp24
Bulunan En iyi Sıra	sp10, sp25, sp11, sp26, sp12, sp27, sp13, sp28, sp14, sp21, sp2, sp20, sp1, sp6, sp3, sp22, sp5, sp29, sp9, sp18, sp32, sp19, sp33, sp16, sp15, sp30, sp17, sp23, sp7, sp4, sp8, sp31, sp34, sp24

Algoritma sonlandıktan sonra bulunan en iyi sıraya ait tablo yazdırılır.

	Sefer No	Kaynak	Hedef	Kalkış	Varış	Aylak Süre
Sonuç	sp10	KEKLİK	DOĞAN	300	785	300
	sp25	DOĞAN	KEKLİK	840	1320	55
	sp11	KEKLİK	DOĞAN	1410	465	90
	sp26	DOĞAN	KEKLİK	540	1010	75
	sp12	KEKLİK	DOĞAN	750	1235	1180
	sp27	DOĞAN	KEKLİK	0	480	205
	sp13	KEKLİK	DOĞAN	510	995	30
	sp28	DOĞAN	KEKLİK	1080	120	85
	sp14	KEKLİK	DOĞAN	630	1115	510
	sp21	DOĞAN	ŞAHİN	1200	470	85
	sp2	ŞAHİN	DOĞAN	600	1339	130
	sp20	DOĞAN	ŞAHİN	1380	590	41
	sp1	ŞAHİN	TURNA	1320	770	730
	sp6	TURNA	ŞAHİN	1140	590	370
	sp3	ŞAHİN	DOĞAN	720	19	130
	sp22	DOĞAN	KANARYA	60	590	41
	sp5	KANARYA	DOĞAN	690	1204	100

180

Diğer bilgiler istatistik bilgilerinin yer aldığı satırlarda bulunmaktadır.

Parametreler	
Nokta Sayısı	9
Sefer Sayısı	34
Nüfus Sayısı	150
İterasyon Sayısı	1500
Çaprazlama Sayısı	0.9
Mutasyon Sayısı	0.15
Yapılan Yaklaşık İşlem Sayısı	606.070.563
Açılış Zamanı	26/05/2018 00:58:12
Toplam Sürüş Süresi	22119 dakika = 15 days, 8:39:00
Mevcut Aylak Süre	12441 dakika = 8 days, 15:21:00
Mevcut Verimlilik	iyi: %64.00, kötü: %64.00
Bulunan En İyi Aylak Süre	12441 dakika = 8 days, 15:21:00
Bulunan En ? Verimlilik	iyi: %64.00, kötü: %51.20

Programın işleminin ne kadar sürdüğü ve yorum tablonun en altında yer alır.

Hesap Süresi	0:00:33
Yorum	Toplam 34 adet araç yerine, bu işlem 25 araç ile yapılabilmektedir. En iyi ile kötü tur arasında 4 araç tasarrufu söz konusudur.



EK - 9 Problem çözümünde 10 deneme ve oluşan sıralar

No	Sıra
1	sp26, sp9, sp16, sp12, sp20, sp1, sp6, sp2, sp27, sp13, sp28, sp14, sp31, sp34, sp21, sp3, sp22, sp5, sp23, sp7, sp4, sp8, sp24, sp10, sp25, sp11, sp29, sp15, sp30, sp18, sp32, sp19, sp33, sp17
2	sp31, sp32, sp19, sp33, sp16, sp9, sp18, sp34, sp24, sp10, sp25, sp12, sp20, sp1, sp6, sp2, sp22, sp5, sp23, sp7, sp4, sp8, sp21, sp3, sp26, sp11, sp27, sp13, sp28, sp15, sp29, sp14, sp30, sp17
3	sp9, sp18, sp34, sp21, sp1, sp6, sp2, sp20, sp3, sp22, sp5, sp23, sp7, sp4, sp8, sp24, sp10, sp25, sp11, sp26, sp12, sp27, sp13, sp28, sp14, sp29, sp15, sp30, sp17, sp31, sp32, sp19, sp33, sp16
4	sp16, sp9, sp18, sp34, sp21, sp1, sp6, sp2, sp20, sp3, sp22, sp5, sp23, sp7, sp4, sp8, sp24, sp10, sp25, sp11, sp26, sp12, sp27, sp13, sp28, sp14, sp29, sp15, sp30, sp17, sp31, sp32, sp19, sp33
5	sp31, sp32, sp19, sp33, sp16, sp9, sp18, sp34, sp21, sp1, sp6, sp2, sp20, sp3, sp23, sp7, sp4, sp8, sp22, sp5, sp24, sp10, sp25, sp11, sp26, sp12, sp27, sp13, sp28, sp14, sp29, sp15, sp30, sp17
6	sp25, sp11, sp26, sp12, sp27, sp13, sp28, sp14, sp29, sp15, sp30, sp17, sp31, sp32, sp19, sp33, sp16, sp9, sp18, sp34, sp21, sp1, sp6, sp2, sp20, sp3, sp22, sp5, sp23, sp7, sp4, sp8, sp24, sp10
7	sp34, sp21, sp2, sp22, sp5, sp20, sp1, sp6, sp3, sp23, sp7, sp4, sp8, sp24, sp10, sp25, sp11, sp26, sp12, sp27, sp13, sp28, sp14, sp29, sp15, sp30, sp17, sp31, sp32, sp19, sp33, sp16, sp9, sp18
8	sp1, sp6, sp2, sp27, sp13, sp28, sp14, sp21, sp3, sp22, sp5, sp23, sp7, sp4, sp8, sp29, sp15, sp30, sp17, sp31, sp32, sp19, sp33, sp16, sp9, sp18, sp34, sp24, sp10, sp25, sp11, sp26, sp12, sp20
9	sp7, sp4, sp8, sp31, sp32, sp19, sp33, sp16, sp9, sp18, sp34, sp24, sp10, sp25, sp11, sp26, sp12, sp20, sp2, sp21, sp1, sp6, sp3, sp22, sp5, sp27, sp13, sp28, sp14, sp29, sp15, sp30, sp17, sp23
10	sp20, sp3, sp22, sp5, sp24, sp10, sp25, sp11, sp26, sp9, sp16, sp12, sp27, sp13, sp28, sp14, sp29, sp15, sp30, sp18, sp34, sp31, sp32, sp19, sp33, sp17, sp23, sp7, sp4, sp8, sp21, sp1, sp6, sp2

KAYNAKLAR

Adamidis, P., Voliotis, C. ve Pliatsika, E. (2012) “An Evolutionary Algorithm for a Real Vehicle Routing Problem”, *Logistics.Teithe.Gr*, 7(3). Available at: http://www.logistics.teithe.gr/icsc2012/fullabstracts/session_1/1_7_ICSC_12_ADAMIDIS.pdf.

Archetti, C., Savelsbergh, M. W. P. ve Speranza, M. G. (2006) “Worst-Case Analysis for Split Delivery Vehicle Routing Problems”, *Transportation Science*, 40(2), ss. 226–234. doi: 10.1287/trsc.1050.0117.

Baker, B. M. ve Ayechev, M. A. (2003) “A genetic algorithm for the vehicle routing problem”, *Computers & Operations Research*, 30(5), ss. 787–800. doi: 10.1016/S0305-0548(02)00051-5.

Baldacci, R. vd. (2011) “An Exact Algorithm for the Period Routing Problem”, *Operations Research*, 59(1), ss. 228–241. doi: 10.1287/opre.1100.0875.

Baldacci, R., Toth, P. ve Vigo, D. (2010) “Exact algorithms for routing problems under vehicle capacity constraints”, *Annals of Operations Research*, 175(1), ss. 213–245. doi: 10.1007/s10479-009-0650-0.

Barış Keçeci, Fulya Altıparmak, İ. K. (2015) “Heterojen eş-zamanlı topla-dağıt araç rotalama problemi: matematiksel modeller ve sezgisel bir algoritma”, *Journal of the Faculty of Engineering and Architecture of Gazi University*, 30(2), ss. 185–195.

Bayrak, A. ve Özyörük, B. (2017) “Bölünmüş Dağıtımli Eş Zamanlı Topla Dağıt Araç Rotalama Problemi İçin Karşılaştırmalı Matematiksel Modeller”, *Gazi Üniversitesi Mühendislik-Mimarlık Fakültesi Dergisi*, 32(2), ss. 469–479. doi: 10.17341/gazimmfd.322172.

Beck, J. C., Prosser, P. ve Selensky, E. (2003) “Vehicle Routing and Job Shop Scheduling: What’s the difference?”, *13th International Conference on Artificial Intelligence Planning and Scheduling (ICAPS)*, ss. 267–276. Available at: <http://www.aaai.org/Papers/ICAPS/2003/ICAPS03-027.pdf>.

Berghida, M. ve Boukra, A. (2015) “EBBO: an enhanced biogeography-based optimization algorithm for a vehicle routing problem with heterogeneous fleet, mixed backhauls, and time windows”, *International Journal of Advanced Manufacturing Technology*, 77(9–12), ss. 1711–1725. doi: 10.1007/s00170-014-6512-1.

Bie, Y., Gong, X. ve Liu, Z. (2015) “Time of day intervals partition for bus schedule using GPS data”, *Transportation Research Part C: Emerging Technologies*. Elsevier Ltd, 60, ss. 443–456. doi: 10.1016/j.trc.2015.09.016.

Braekers, K., Ramaekers, K. ve Van Nieuwenhuyse, I. (2015) “The vehicle routing problem: State of the art classification and review”, *Computers and Industrial Engineering*. Elsevier Ltd, 99, ss. 300–313. doi: 10.1016/j.cie.2015.12.007.

Cadarso, L. ve de Celis, R. (2017) “Integrated airline planning: Robust update of scheduling and fleet balancing under demand uncertainty”, *Transportation Research*

Part C: Emerging Technologies. Elsevier Ltd, 81, ss. 227–245. doi: 10.1016/j.trc.2017.06.003.

Campbell, A. M. ve Wilson, J. H. (2014) “Forty years of periodic vehicle routing”, *Networks*, 63(1), ss. 2–15. doi: 10.1002/net.21527.

Chen, S., Chen, R. ve Gao, J. (2017) “A Monarch Butterfly Optimization for the Dynamic Vehicle Routing Problem”, *Algorithms*, 10(3), s. 107. doi: 10.3390/a10030107.

Cordeau, J.-F. vd. (2007) “Chapter 6 Vehicle Routing”, içinde *Transportation*, ss. 367–428. doi: [http://dx.doi.org/10.1016/S0927-0507\(06\)14006-2](http://dx.doi.org/10.1016/S0927-0507(06)14006-2).

Dai, Z. ve Zheng, X. (2015) “Design of close-loop supply chain network under uncertainty using hybrid genetic algorithm: a fuzzy and chance-constrained programming model”, *Computers & Industrial Engineering*. Elsevier Ltd, 88, ss. 444–457. doi: 10.1016/j.cie.2015.08.004.

Daneshzand, F. (2011) *The Vehicle-Routing Problem, Logistics Operations and Management*. Elsevier Inc. doi: 10.1016/B978-0-12-385202-1.00008-6.

Dantzig, G. B. ve Ramser, J. H. (1959) “The Truck Dispatching Problem”, *Management Science*, 6(1), ss. 80–91. doi: 10.1287/mnsc.6.1.80.

Dantzig, G., Fulkerson, R. ve Johnson, S. (1954) “Solution of a Large-Scale Traveling-Salesman Problem”, *Journal of the Operations Research Society of America*, 2(4), ss. 393–410. doi: 10.1287/opre.2.4.393.

Domínguez-Martín, B., Rodríguez-Martín, I. ve Salazar-González, J.-J. (2018) “The driver and vehicle routing problem”, *Computers & Operations Research*, 92, ss. 56–64. doi: 10.1016/j.cor.2017.12.010.

Domínguez-Martín, B., Rodríguez-Martín, I. ve Salazar-González, J. J. (2017) “An exact algorithm for a Vehicle-and-Driver Scheduling Problem”, *Computers and Operations Research*, 81, ss. 247–256. doi: 10.1016/j.cor.2016.12.022.

Dror, M., Laporte, G. ve Trudeau, P. (1994) “Vehicle routing with split deliveries”, *Discrete Applied Mathematics*, 50(3), ss. 239–254. doi: 10.1016/0166-218X(92)00172-I.

Eiben, A. E. ve Smit, S. K. (2011) “Evolutionary Algorithm Parameters and Methods to Tune Them”, içinde *Autonomous Search*. Berlin, Heidelberg: Springer Berlin Heidelberg, ss. 15–36. doi: 10.1007/978-3-642-21434-9_2.

Ekşioglu, B., Vural, A. V. ve Reisman, A. (2009) “The vehicle routing problem: A taxonomic review”, *Computers and Industrial Engineering*. Elsevier Ltd, 57(4), ss. 1472–1483. doi: 10.1016/j.cie.2009.05.009.

El-Sherbeny, N. A. (2010) “Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods”, *Journal of King Saud University - Science*. Japanese Association for Dental Science, 22(3), ss. 123–131. doi: 10.1016/j.jksus.2010.03.002.

García-Nájera, A., Bullinaria, J. A. ve Gutiérrez-Andrade, M. A. (2015) “An evolutionary approach for multi-objective vehicle routing problems with backhauls”, *Computers & Industrial Engineering*, 81, ss. 90–108. doi: 10.1016/j.cie.2014.12.029.

Gendreau, M., Ghiani, G. ve Guerriero, E. (2015) “Time-dependent routing problems: A review”, *Computers & Operations Research*, 64, ss. 189–197. doi: 10.1016/j.cor.2015.06.001.

Hassold, S. ve Ceder, A. A. (2014) “Public transport vehicle scheduling featuring multiple vehicle types”, *Transportation Research Part B: Methodological*. Elsevier Ltd, 67, ss. 129–143. doi: 10.1016/j.trb.2014.04.009.

Hemmelmayr, V. C., Doerner, K. F. ve Hartl, R. F. (2009) “A variable neighborhood search heuristic for periodic routing problems”, *European Journal of Operational Research*. Elsevier B.V., 195(3), ss. 791–802. doi: 10.1016/j.ejor.2007.08.048.

Jing-Quan Li, P. B. M. and D. B. (2006) “The k -Degree Cayley Graph and its Topological Properties”, *Networks*, 47(1), ss. 26–36. doi: 10.1002/net.

Kim, B.-I., Kim, S. ve Park, J. (2012) “A school bus scheduling problem”, *European Journal of Operational Research*. Elsevier B.V., 218(2), ss. 577–585. doi: 10.1016/j.ejor.2011.11.035.

Koç, Ç. ve Karaoglan, I. (2016) “The green vehicle routing problem: A heuristic based exact solution approach”, *Applied Soft Computing Journal*, 39, ss. 154–164. doi: 10.1016/j.asoc.2015.10.064.

Kuo, Y. ve Wang, C.-C. (2011) “Optimizing the VRP by minimizing fuel consumption”, *Management of Environmental Quality: An International Journal*, 22(6), ss. 440–450. doi: 10.1108/14777831111136054.

Laporte, G. (2009) “Fifty Years of Vehicle Routing”, *Transportation Science*, 43(4), ss. 408–416. doi: 10.1287/trsc.1090.0301.

Laporte, G. (2013) “Scheduling issues in vehicle routing”, *Annals of Operations Research*. Springer Science+Business Media New York, (July 2013), ss. 1–12. doi: 10.1007/s10479-013-1423-3.

Lin, C. vd. (2014) “Survey of Green Vehicle Routing Problem: Past and future trends”, *Expert Systems with Applications*, 41(4 PART 1), ss. 1118–1138. doi: 10.1016/j.eswa.2013.07.107.

Matai, R., Singh, S. P., & Mittal, M. L. (2010) “Traveling Salesman Problem: An Overview of Applications, Formulations, and Solution Approaches”.

minitab (2018) *Interpret the key results for Analyze Taguchi Design*. Available at: <https://support.minitab.com/en-us/minitab/18/help-and-how-to/modeling-statistics/doe/how-to/taguchi/analyze-taguchi-design/interpret-the-results/key-results/> (Erişim: 02 Ocak 2018).

Montoya-Torres, J. R. *vd.* (2015) “A literature review on the vehicle routing problem with multiple depots”, *Computers & Industrial Engineering*. Elsevier Ltd, 79, ss. 115–129. doi: 10.1016/j.cie.2014.10.029.

Pimenta, V. *vd.* (2017) “Models and algorithms for reliability-oriented Dial-a-Ride with autonomous electric vehicles”, *European Journal of Operational Research*. Elsevier B.V., 257(2), ss. 601–613. doi: 10.1016/j.ejor.2016.07.037.

Pita, J. P., Barnhart, C. ve Antunes, A. P. (2013) “Integrated Flight Scheduling and Fleet Assignment Under Airport Congestion”, *Transportation Science*, 47(4), ss. 477–492. doi: 10.1287/trsc.1120.0442.

Potvin, J.-Y. (2009) “**State-of-the Art Review** —Evolutionary Algorithms for Vehicle Routing”, *INFORMS Journal on Computing*, 21(4), ss. 518–548. doi: 10.1287/ijoc.1080.0312.

Rahimi-Vahed, A. *vd.* (2015) “Fleet-sizing for multi-depot and periodic vehicle routing problems using a modular heuristic algorithm”, *Computers & Operations Research*. Elsevier, 53, ss. 9–23. doi: 10.1016/j.cor.2014.07.004.

Rahimi Mazrae Shahi, M., Fallah Mehdipour, E. ve Amiri, M. (2016) “Optimization using simulation and response surface methodology with an application on subway train scheduling”, *International Transactions in Operational Research*, 23(4), ss. 797–811. doi: 10.1111/itor.12150.

Reinelt, G. (tarih yok) “TSPLIB 95”. Universitat Heidelberg.

Şahin, M. *vd.* (2013) “An efficient heuristic for the Multi-vehicle One-to-one Pickup and Delivery Problem with Split Loads”, *Transportation Research Part C: Emerging Technologies*, 27, ss. 169–188. doi: 10.1016/j.trc.2012.04.014.

Schmid, V. ve Ehmke, J. F. (2015) “Integrated timetabling and vehicle scheduling with balanced departure times”, *OR Spectrum*, 37(4), ss. 903–928. doi: 10.1007/s00291-015-0398-7.

Shui, X. *vd.* (2015) “A clonal selection algorithm for urban bus vehicle scheduling”, *Applied Soft Computing Journal*. Elsevier B.V., 36, ss. 36–44. doi: 10.1016/j.asoc.2015.07.001.

Sivanandam, S. N. ve Deepa, S. N. (2008) *Introduction to genetic algorithms, Introduction to Genetic Algorithms*. doi: 10.1007/978-3-540-73190-0.

Stojković, M. ve Soumis, F. (2001) “An Optimization Model for the Simultaneous Operational Flight and Pilot Scheduling Problem”, *Management Science*, 47(9), ss. 1290–1305. doi: 10.1287/mnsc.47.9.1290.9780.

Sun, J. U. (2007) “A Taguchi Approach to Parameter Setting in a Genetic Algorithm for General Job Shop Scheduling Problem”, *Industrial Engineering and Management Systems*, 6(2), ss. 119–124.

Tonci caric, H. G. (ed.) (2008) *Vehicle Routing Problem*. Faculty of, Optimization. Faculty of. Zagreb, Croatia: Uniaversity of Zagreb.

Toth, P. ve Vigo, D. (1999) “A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls”, *European Journal of Operational Research*, 113(3), ss. 528–543. doi: 10.1016/S0377-2217(98)00086-1.

Vehicle Routing Problem | NEO Research Group (2013). Available at: <http://neo.lcc.uma.es/vrp/> (Erişim: 27 Ekim 2017).

Wagale, M. vd. (2013) “Real-time Optimal Bus Scheduling for a City Using A DTR Model”, *Procedia - Social and Behavioral Sciences*. Elsevier B.V., 104, ss. 845–854. doi: 10.1016/j.sbspro.2013.11.179.

Wang, Y. vd. (2017) “Train scheduling and circulation planning in urban rail transit lines”, *Control Engineering Practice*. Elsevier Ltd, 61(February), ss. 112–123. doi: 10.1016/j.conengprac.2017.02.006.

Yue, Y. vd. (2017) “Integrated Train Timetabling and Rolling Stock Scheduling Model Based on Time-Dependent Demand for Urban Rail Transit”, *Computer-Aided Civil and Infrastructure Engineering*, 32(10), ss. 856–873. doi: 10.1111/mice.12300.

Zhang, Z. vd. (2013) “A memetic algorithm for the multiperiod vehicle routing problem with profit”, *European Journal of Operational Research*. Elsevier B.V., 229(3), ss. 573–584. doi: 10.1016/j.ejor.2012.11.059.

ÖZGEÇMİŞ

Mail Adresi: omernuricam@gmail.com, onc@uludag.edu.tr

Uzmanlık Alanları: Yöneylem Araştırması, Algoritma Geliştirme, Bilgisayar (Donanım / Programlama) , Parametrik – Parametrik olmayan 3B Modelleme - Yazdırma, Grafik Tasarım

Yabancı Dil: İngilizce

Öğrenim Durumu

Üniversite : Ege Üniversitesi Deri Mühendisliği : 03.08.2005

Yüksek Lisans: Endüstri Mühendisliği : 12.10.2009

İdari Görevler		
2009-2013	Bilgi İşlem Daire Başkanlığı	
<ul style="list-style-type: none">• Web Sunucusu Yönetimi• Bulut Sistemleri Yönetimi• Linux Sunucu Yönetimi• Web Programlama• Arayüz Programlama		
Akademik Çalışmalar		
Tezler		
Yüksek Lisans: Atanacak Ekip Özelliklerine Göre Değişken Süreye Sahip Projelerin Planlanması İçin Genetik Algoritma Bazlı Bir Yöntemin Geliştirilmesi		
Kongreler		
<ul style="list-style-type: none">• Gülseren D.A., Çam Ö.N., Anlık Yazışma Programlarının Kültürlerarası Diyaloga Etkileri ve Fırsatlar Bursa 2009• Gülseren D.A., Çam Ö.N., Uluslararası Boyutta Alışveriş Yapma Oranı, Tercih Sebeplerinin Değerlendirilmesi ve İnternetin Rolü Bursa 2009• Gülseren D.A., Çam Ö.N., Teknik Bilimler MYO Grafik Programının Uzaktan Eğitime Geçiş Çalışmaları ve Uzaktan Eğitimin Gelecekte Yaratacağı Fırsatlar Konya 2009• Uludağ Üniversitesi Web Sayfası Kullanılabilirlik Analizi International Conference On New Horizons İn Education (Poster) Viyana 2016• "BİLGİSAYAR Destekli Grafik Tasarımının U.Ü Tbmyo Grafik Öğrencilerinin Tasarımlarına Etkilerinin Değerlendirilmesi"• Uav/Drone Usage İn Agriculture, (Poster) Bursa 2016• "ÖĞRENCİ Oryantasyonunda Artırılmış Gerçeklikten Yararlanma – Uludağ Üniversitesi Modeli",International Conference On Quality İn Higher Education (ICQH), Sakarya 2016• Eğitimin Web 3.0 Teknolojisi İle Desteklenmesi,International Conference On Quality İn Higher Education (ICQH), Sakarya 2016• "GRAFİK Tasarımcılarının Önlisans Eğitiminde Aldıkları• Bilgisayar Tabanlı Programların İş Hayatında Sürdürülebilirliği, Uü Tbmyo Grafik Programı Mezun Anketi", 14th International Jtefs/Bbcc conference Sustainable development, Culture,Education, Konya 2016• Açık Kaynaklı/Ticari Yazılımların Mesleki Eğitimde Sürdürülebilirliğe Katkılarının		

Tartışılması,14th International Jtefs/Bbcc conference Sustainable development, Culture,Education, Konya 2016

- Uludağ Infografik, Arge Günleri, (Poster) Bursa, 2017
- "Kendin Yap" Projelerinin Teknik Eğitime Katkısı Ve Yetersiz Türkçe İçerik Problemi, Ordu 2017
- Avrupa Birliği Projeleri ile Kazanımlar ve bir hareketlilik projesi örneği, Cloud Computing, Ordu 2017
- Ülkemizde Paralel programlama/Çoklu işleme teknolojilerine olan ilginin incelenmesi, GPU, CUDA, OPENCL , ICITS, Malatya 2017
- Uludağ Üniversitesi Teknik Bilimler Meslek Yüksekokulu Grafik Tasarımı Programında Ters Yüz Sınıf (Flipped Classroom) Modelinin Uygulanması ve Etkilerinin Araştırılması, ICITS, Malatya 2017
- Vehicle Routing Problem Which Is Minimized Idle Time, ICBED, London 2017

Araştırma-Eğitim Projeleri

- Gülseren D.A., Çam Ö.N., Cloud Computing 2011
- Alper E., Gülseren D.A., Çam Ö.N., Uludağ Üniversitesi Online Anket Sistemi 2011
- Gülseren D.A., Çam Ö.N., 3dsmax ile Modelleme 2011
- Gülseren D.A., Çam Ö.N., Autodesk Maya ile Karakter Modelleme ve Animasyon 2012
- Gülseren D.A., Çam Ö.N., Oyun Programlama 2017

Ödüller

- Bologna Süreci Çalışması Özel Ödülü - 2011
- Geri Bildirim Grubu Özel Ödülü – 2013

Eğitimler

- Ofis Eğitimi, UÜ Personel, Bursa 2010
- Linux Eğitimi, UÜ Bilgi İşlem Dairesi Bursa 2017
- BOSH GBE Sensörük Eğitimi 2014-2016

Diğer

- CorelDraw, FreeHand, Photoshop, Flash Eğitimi, Dokuz Eylül Sürekli Eğitim Merkezi İzmir,2004-2005
- Bursaspor ve TV web hizmeti 2015 (programlama + tasarım)

Bildiği Yazılım Dilleri

- C / C++
- C#
- Python
- Php
- Javascript / Typescript / EcmaScript
- HTML

ULUDAĞ ÜNİVERSİTESİ

TEZ ÇOĞALTMA VE ELEKTRONİK YAYIMLAMA İZİN FORMU

Yazar Adı Soyadı	Ömer Nuri ÇAM
Tez Adı	Aylak Zamanı En Küçükleme Amaçlı Araç Rotalama Probleminin Genetik Algoritma İle Çözümü
Enstitü	Uludağ Üniversitesi Sosyal Bilimler Enstitüsü
Anabilim Dalı	Ekonometri
Tez Türü	Doktora
Tez Danışman(lar)ı	Prof.Dr.H.Kemal SEZEN
Çoğaltma (Fotokopi Çekim) izni	<input checked="" type="checkbox"/> Tezimden fotokopi çekilmesine izin veriyorum <input type="checkbox"/> Tezimin sadece içindekiler, özet, kaynakça ve içeriğinin % 10 bölümünün fotokopi çekilmesine izin veriyorum <input type="checkbox"/> Tezimden fotokopi çekilmesine izin vermiyorum
Yayımlama izni	<input checked="" type="checkbox"/> Tezimin elektronik ortamda yayımlanmasına izin Veriyorum

Hazırlamış olduğum tezimin belirttiğim hususlar dikkate alınarak, fikri mülkiyet haklarım saklı kalmak üzere Uludağ Üniversitesi Kütüphane ve Dokümantasyon Daire Başkanlığı tarafından hizmete sunulmasına izin verdiğimi beyan ederim.

Tarih 17.05.2018

İmza :

