



**Heterojen Araç Filolu Çok Depolu Araç Rotalama
Problemi için Görsel Etkileşimli Sezgisel Çözüm Yöntemleri**

Program Kodu: 1001

Proje No: 213M438

Proje Yürütücüsü:
Gözde Yazgı TÛTÛNCÛ

Bursiyerler:

Fatih KOCATÛRK

Gökhan KAHRAMAN

Ali KANAT

ÖNSÖZ

Bu projenin ana fikri 2012 yılında İzmir Ekonomi Üniversitesinde öğrenciler için düzenlenen bir lojistik konferansında o zamanın EDAK Ecza Kooperatifi yönetim kurulu başkanı ile araç rotalama konusunda yaptığımız bir sohbetten sonra ortaya çıktı. Doktora tezimin konusu olan Araç Rotalama Optimizasyonunun gerçek hayatta ilginç varyasyonlarının karşımıza çıkmasıyla doktoram ve sonrasında yaptığım teorik çalışmaların uygulamayla bağdaştırılarak daha da genişletilip faydalı hale gelebileceğini fark ettim. Daha sonra, bu problemi doktora öğrencilerimden biri olan Fatih Kocatürk'e tez konusu olarak önerdim. Fatih doktora yeterlilik sınavına hazırlanırken deproje önerisini TÜBİTAK'a sundum. Projenin ilk yılında proje önerimizde sunduğumuz özelliklere sahip yazılımcı bir yüksek lisans öğrencisi bulamadığımızdan sadece Fatih Kocatürk ile beraber çalıştık. Projenin ikinci yılı sırasıyla Ali Kanat ve Gökhan Kahraman yazılım konularında bana yardım etmek üzere proje de bursiyer olarak ancak kısa süreli yer aldılar. Projenin ilk yılında geliştirilmeye başlanan Karar Destek Sisteminin yazılımı ile yaşanan aksaklıklar nedeniyle bir yazılım danışmanlık firmasından kısa süreli yazılım geliştirme danışmanlık desteği alındı. Alınan bu destek ve bursiyerlerimle gösterdiğimiz üstün çabalarla Ocak 2016 tarihine kadar proje önerisinde sunduğumuz Heterojen Araç Filolu ve Çok depolu Araç Rotalama problemi için akademik literatürde önerilen problemleri görüntüleyen ve görsel etkileşimli bir şekilde çözebilen Karar Destek Sistemi geliştirildi. Geliştirilen Prototip, içinde EDAK Yönetim kurulu üyelerinde bulunduğu, İzmir bölgesinde bu konuyla ilgilenen çeşitli firma çalışanlarının yer aldığı bir toplantıda tanıtıldı. Alınan geri beslemeler doğrultusunda Karar Destek Sistemi, Mart 2016 da problemin geri toplamalı versiyonunu da çözebilir hale getirildi. Alınan ek 4 aylık sürede de EDAK'ın gerçek problemini çözebilen versiyonu başarı ile geliştirildi. Geliştirilen Karar Destek Sisteminin içinde önerdiğimiz üst sezgisel ve karar destek sisteminin kullanıcı ara yüzüne eklediğimiz interaktif araçlarla literatürde verilen Heterojen Araçlı Çok Depolu Araç Rotalama problem örnekleri için çok iyi (bilinen en iyi çözümden ortalama %0,97 uzak ve %98,82 daha kısa sürede) sonuçlar elde etmeyi başardık. Ayrıca Geri-toplamalı Heterojen Araçlı Çok Depolu Araç Rotalama problemi için literatürdeki ilk örnek problemleri önerdik ve bunlar için de çok iyi (alt sınırdan ortalama %9,46, üst sınırdan %12,33 uzak ve ortalama 0,83sn sürede) sonuçlar elde etmeyi başardık. EDAK'ın gerçek hayat problemini çözüp, Karar Destek Sisteminin kullanımını son bir toplantı ile tanıttık. Bu projeye vermiş olduğu destekten dolayı TÜBİTAK'a ve bilgi ve belge paylaşımındaki desteklerinden dolayı EDAK Yönetim Kuruluna, teşekkürlerimizi sunuyoruz.

İÇİNDEKİLER

ÖNSÖZ	i
İÇİNDEKİLER	ii
Şekiller Dizini	iv
Tablolar Dizini	v
Algoritmalar Dizini	vi
ÖZET	vii
ABSTRACT	viii
1. GİRİŞ	1
2. LİTERATÜR ÖZETİ	2
2.1 HAÇARP Literatür Özeti	3
2.2 GHAÇARP Literatür Özeti	4
2.3 HAÇARP Sezgisel Yöntemler Literatür Özeti	6
2.4 GHAÇARP Sezgisel Yöntemler Literatür Özeti	8
3. GEREÇ VE YÖNTEM	10
3.1 Matematiksel Modeller	10
3.1.1 HAÇARP Matematiksel Modeli	10
3.1.2 GHAÇARP Matematiksel Modeli	15
3.2 Problem Örnekleri	21
3.2.1 HAÇARP Problem Örnekleri	21
3.2.2 GHAÇARP Problem Örnekleri	22
3.3 KDS Tanıtımı	24
3.3.1 KDS Tasarımı	24
3.4 Sezgisel Algoritmalar ve KDS Uygulamaları	32
3.4.1 Aç-gözlü Rastgele Adaptif Hafıza Programlama Arama (ARAHPA) Sezgiseli KDS Uygulaması	32
3.4.2 Yeni Aç-gözlü Rastgele Adaptif Hafıza Programlama Arama (ARAHPA) Sezgiseli	36
3.4.3 GHAÇARP için yeni ARAHPA Sezgiseli	47
3.4.4 Yeni Aç-gözlü Rastgele Adaptif Hafıza Programlama Arama 2 (ARAHPA2) Sezgiseli	48
3.4.4.1 Otomatik Başlangıç Çözümü Oluşturma (OBÇO) Sezgiseli	49
3.4.4.2 Aç-gözlü Rastgele Adaptif Hafıza Programlama Arama (ARAHPA) Sezgiseli Düzenlemeleri	51
4. BULGULAR	52
4.1 GHAÇARP Deney Sonuçları	52

4.2 HAÇARP için yeni ARAHPA Deney Sonuçları.....	55
4.3 EDAK İzmir Dağıtım Problemi Sonuçları	58
5. SONUÇ	58
KAYNAKLAR	60

Şekiller Dizini

Şekil 1. Karar destek sisteminin veri tabanışeması.....	25
Şekil 2. Karar destek sisteminin ana sayfası ve araç kısıtları ekranı.....	27
Şekil 3. Araç çubuğu	27
Şekil 4. Müşteri ekleme uyarı penceresi örneği.....	29
Şekil 5. Karar destek sisteminde yeni problem açma penceresi	30
Şekil 6. Çekirdek çözüm oluşturma	31
Şekil 7. Etkileşim paneli örneği	32
Şekil 8. Başlangıç çekirdek çözüm seçimi	33
Şekil 9. OBÇO sezgiseli parametre seçme ekranı	51
Şekil 10. EDAK İzmir ili ilaç dağıtım problemi çözümü	58

Tablolar Dizini

Tablo 1. HAÇARP Problem Örnekleri	21
Tablo 2. GHAÇARP Problem Örnekleri	23
Tablo 3. GHAÇARP için CPLEX Deney Sonuçları	53
Tablo 4. GHAÇARP için yeni ARAHPA Deney Sonuçları	54
Tablo 5. GHAÇARP için yeni ARAHPA2 Deney Sonuçları	55
Tablo 6. HAÇARP için yeni ARAHPA Deney Sonuçları	56
Tablo 7. HAÇARP için yeni ARAHPA2 Deney Sonuçları	57

Algoritmalar Dizini

Algoritma 1. Görsel etkileşimli KDS için ARAPHA algoritması	35
Algoritma 2. US1 Algoritması.....	37
Algoritma 3. US2 Algoritması.....	38
Algoritma 4. GUAA Çözüm Önerme Algoritması.....	42
Algoritma 5. KDKA Yerel Arama Algoritması	45
Algoritma 6. Otomatik Başlangıç Çözümü Oluşturma (OBÇO) Algoritması	50

ÖZET

Araç Rotalama Problemleri (ARP) her birine belli büyüklükteki ve nitelikteki ürün ya da yolcuların araç filosu ile ulaştırılması gereken birçok sayıda müşteri ya da durak içerir. ARP yöneylem araştırması ve yönetim bilimleri disiplinlerinin hem akademik araştırma hem de uygulama açısından en önemli alanlarından birisidir. Gerçek hayatta ARP problemleri temel ARP 'nin ötesinde ek kısıtlar ve zorluklar içermektedir. Bu projede gerçek hayatta çokça rastlanan ve birçok tedarik firması tarafından göz önüne alınıp önemsenen Heterojen Araç-filolu Çok-depolu Araç Rotalama Problemi (HAÇARP) ve HAÇARP 'in yaygın kullanımlı bir versiyonu olan Geri-toplamalı HAÇARP (GHAÇARP) ele alınmıştır.

Bu projede öncelikle HAÇARP için yeni sezgisel yöntemler geliştirilmiş, geliştirilen sezgiseller literatürdeki karşılaştırmalı problem örnekleriyle test edilmiş ve gerçek hayat problemlerini çözmek üzere uygulamaya sunulmuştur. Daha sonra GHAÇARP için yeni bir matematiksel model önerilmiş, gerçek hayat problemlerini baz alan yeni problem örnekleri oluşturulmuş ve önerilen model kullanılarak bu problemlerin gevşetilmiş çözümleri elde edilmiştir. Buna ek olarak, HAÇARP için geliştirilen yeni sezgisel yöntemler göz önüne alınarak GHAÇARP için de yeni bir sezgisel geliştirilmiş ve önerilen problem örnekleriyle test edilmiş ve gerçek hayat problemlerini çözmek üzere uygulamaya sunulmuştur.

Sonuç olarak günlük hayatta çeşitli firmalarca kullanılabilir, farklı araç rotalama problemlerinin çözümü için uyarlanabilir, görsel etkileşimli ve sezgisel yöntemlere dayalı ve ele alınan problemleri iyi sonuçlarla çözebilen bir Karar Destek Sistemi geliştirilmiştir.

Anahtar Kelimeler: Çok Depolu Araç Rotalama Problemi, Heterojen Araçlar, Sezgisel Yöntemler, Görsel Etkileşimli Karar Destek Sistemi, Matematiksel Modelleme

ABSTRACT

Vehicle Routing Problems (VRP) consist of a large number of customers or stations, each requiring a specified weight of goods to be delivered by a fleet of vehicles. Vehicle routing has appeared as one of the major areas of both research and practice in the field of management science and operations research. In real life, vehicle routing problems include additional requirements and difficulties beyond the basic VRP. In this project, we consider MDVRPHVF and MDVRPHVFB which are faced usually in real life and taken into account by many supply chain companies.

In the first stage of the project, we developed a new heuristic method for the MDVRPHVF that can give good results which can compete with the results presented in the literature and can be embedded easily to be used in practice. Then we proposed a mathematical model for MDVRPHVFB, we generated new problem instances based on realistic setups and existing problem instances of MDVRPHVF and we found the relaxed solutions of these problem instances. The graphical structure of the modified DSS will be modified with respect to the structure of the defined problem. Additionally, heuristic method proposed for MDVRPHVF was modified and improved in order to solve MDVRPHVFBs. We analyzed the effects of the decisions of the proposed heuristics for the problem instances, on computational time and solution quality by preliminary tests.

In the last stage of the project, we developed a visual interactive DSS, which is applicable in real life and able to solve these problems by proposed heuristic methods, for the considered problems.

Keywords: Multi Depot Vehicle Routing Problem, Heterogeneous Vehicles, Heuristic Methods, Visual Interactive Decision Support System, Mathematical Modelling

1. GİRİŞ

Araç Rotalama Problemleri (ARP) dünyada birçok taşımacılık, lojistik ve tedarik zinciri problemleri olan firmaları yakından ilgilendirmektedir. ARP ilk olarak Dantzig ve Ramser (1959) tarafından tanımlanmış ve yaklaşık 60 sene içerisinde ARP ve türevleri için çeşitli kesin ve sezgisel çözüm yöntemleri önerilmiştir.

ARP her birine belli büyüklükteki ve nitelikteki ürün ya da yolcuların araç filosu ile ulaştırılması gereken birçok sayıda müşteri ya da durak içerir. Son 60 yılda ARP hem akademik araştırma hem de yöneylem araştırması ve yönetim bilimleri uygulamalarında en önemli alanlardan biri olmuştur. Hem akademi tarafında hem de uygulama tarafında başarı hikâyeleri yazılmıştır. Gerçek hayatta ARP problemleri temel ARP 'nin ötesinde ek kısıtlar ve zorluklar içermektedir. ARP 'nin uygulamalarıyla çalışırken problemin özelliklerini bilmek önemlidir. Çözüm yöntemine birçok durumda ek özellikler eklenebilmesine rağmen gerçek hayatta problem türünü belirlemek sonraki araştırmalara odaklanma ve zaman tasarrufu açısından önemlidir.

Bu projede, ilk olarak Heterojen Araç-filolu Çok-depolu ARP (HAÇARP) ve Geri-toplamalı Heterojen Araç-filolu Çok-depolu Araç Rotalama Problemi (GHAÇARP) için literatür taramasını tamamladık, sonrasında ise proje kapsamında ilk olarak bizim tarafımızdan tanımlanan GHAÇARP için matematiksel model geliştirdik ve bu modele bazı gevşetme kısıtları önerdik. GHAÇARP problemi için literatürde var olan Geri-toplamalı Heterojen Araç-filolu ARP ve HAÇARP problem örneklerini kullanarak yeni problem örnekleri oluşturduk. GHAÇARP için geliştirdiğimiz temel ve gevşetilmiş model için 3 saat süre sınırı kullanarak CPLEX deneylerini tamamladık ve elde edilen sonuçları aynı problem için geliştirdiğimiz sezgisel yöntemle karşılaştırma yaparken alt sınır olarak kullandık.

Yürütücünün farklı araç rotalama problemlerini çözebilmek için geliştirmiş olduğu Açgözlü Rastgele Adaptif Hafıza Programlama Arama (ARAPHA) sezgiselini HAÇARP ve GHAÇARP için düzenledik. Ayrıca ARAPHA sezgiseline ek olarak HAÇARP ve GHAÇARP için yeni sezgisel yöntemler geliştirmek için literatür taramasını tamamladık. Yapılan literatür taraması sonucunda, HAÇARP ve GHAÇARP için ARAPHA yöntemine kullanıcıya çözüm üzerinde interaktif olarak değişiklikler yapabileceği bir başlangıç çözümü sunabilen otomatik çekirdek çözüm önerme

sezgiselini ekleyerek ve algoritma içinde bazı kritik değişiklikler yapılarak daha iyi çözümler bulabilen yeni ARAHPA2 sezgiselini geliştirdik. Geliştirilen ARAHPA ve ARAHPA2 sezgisellerini HAÇARP için literatürde verilen örnek problemler üzerinde test ettik ve kısa zamanda oldukça iyi performans verdiğini gözlemledik. ARAHPA sezgiseliyle bir problemde (Problem 12) %0,12 aralıkla literatürdeki çözümlerden daha iyi çözüm bulurken ARAHPA2 sezgiseliyle üç problemde (Problemler 1,5,10) daha iyi çözümler bulduk, ortalamada ise literatürdeki en iyi çözümlere %0,97 aralıkla ortalama 3,75 saniye gibi çok kısa CPU zamanında yaklaştık. GHAÇARP için geliştirilen sezgisel yöntemleri, proje kapsamında oluşturulan problem kümesi üzerinde test ettik ve CPLEX sonuçlarıyla karşılaştırdık. Geliştirilen sezgisel yöntemin çok kısa zamanlarda uygulamada rahatlıkla kullanılabilir en iyi çözüme yakın sonuçlar verdiğini gördük. Önerilen ARAHPA ve ARAHPA2 sezgiselleri CPLEX tarafından bulunan 11 en iyi çözümü bir saniyeden daha kısa sürelerde bulmuştur ve ARAHPA sezgiseli CPLEX tarafından bulunan üst sınırlardan %12,33 daha iyiyken ARAHPA2 sezgiseli %11.87 daha iyidir.

Projede ele alınan problemleri çözmek için geliştirilen sezgisel yöntemlerle çalışabilen görsel etkileşimli Karar destek Sistemi (KDS) geliştirdik. Geliştirdiğimiz KDS ile kullanıcıların problemin çözüm sürecine dahil olmasına imkan sağladık. KDS içerisindeki interaktif araçları kullanarak problemlerin çözümünde belirgin iyileştirmeler elde ettik. Sektörde ortalama problemleri olan firmalara hem KDS'yi tanıtmak hem de firma yetkililerinden geri bildirimler almak amacıyla bir tanım toplantısı ve geri besleme toplantısı düzenledik. Ayrıca, KDS üzerinde projemize veri desteğinde bulunan EDAK Ecza Koop. firmasının problemini önerdiğimiz yöntemle çözdük ve firmanın kullanımına sunduk. Bunlarla birlikte, projemizin bilimsel çıktılarını bir uluslararası konferans (Verolog 2015), iki ulusal konferans (YAEM 2015 ve YAEM 2016) ve bir doktora kolokyumunda (YAEM 2016 Doktora Kolokyumu) meslektaşlarımızla paylaştık.

Projemizin sonuç raporunun ikinci bölümünde literatür özetini sunduk. Kullandığımız yöntemleri Bölüm 3'de, deney sonuçlarını Bölüm 4'de aktardık. Beşinci Bölüm'de ise projenin sonuçlarını ve önerilerimizi tartıştık.

2. LİTERATÜR ÖZETİ

Raporun bu bölümünde proje kapsamında çalışılan problem ve kullanılan yöntemler ile ilgili proje dönemi içerisinde yer alan literatür taraması özetlenmiştir. Bu bölüm, sırasıyla HAÇARPliteratür

özeti, GHAÇARP literatür özeti, HAÇARP sezgisel yöntemler literatür özeti ve HAÇARP sezgisel yöntemler literatür özeti alt bölümlerinden oluşmaktadır.

2.1 HAÇARP Literatür Özeti

ARP ilk olarak Dantzig ve Ramser (1959) tarafından tanımlanmıştır ve lojistik dağıtım problemleri için önemli bir metottur. ARP hem kamu hem özel sektörde geniş uygulama alanı olan bir problemdir. ARP 'nin amacı bir depoda bulunan araçlar için müşterilerin taleplerini karşılayacak şekilde en az maliyeti veren rotaları bulmaktır. Bu amacı sağlarken her problem ARP türüne özgü kısıtlar (araç kapasitesi, depo sayısı, geri toplama izni vb.) içerir. ARP ve türevlerinin modelleri ve çözüm yöntemleri hakkında çok geniş bilgiye Toth ve Vigo (2002)' den ulaşılabilir.

Çok-depolu ARP (ÇARP) tek depolu ARP 'ye göre daha gelişmiş ve daha zordur, ancak birçok gerçek hayat problemini ÇARP olarak modellemek ve incelemek daha uygundur. Lenstra ve Rinnooy Kan (1981) ÇARP 'nin NP-Zor bir Kombinatorik Optimizasyon (KO) problemi olduğunu göstermişlerdir. ÇARP 'yi çözmek için genellikle iki aşama kullanılır: Kümeleme aşaması, müşteriler depolara atanır; rotalama aşaması, her küme için en az maliyeti veren rota bulunur. Problem bu aşamaları iki farklı şekilde kullanarak çözülebilir: Önce Rotala Sonra Kümele (ÖRSK), Önce Kümele Sonra Rotala (ÖKSR) (Geetha vd., 2012).

ARP'ye kıyasla ÇARP alanında yapılan çalışmalar çok azdır. Laporte vd. (1984) ve Laporte vd. (1988) ilk olarak simetrik ve asimetric durumlar için kesin çözüm algoritmaları geliştirmişlerdir. ÇARP için geliştirilen sezgisel yöntemler 2.3 bölümünde detaylı şekilde anlatılmıştır.

Lojistik temelli şirketler genellikle birden fazla dağıtım merkezinde kullanmak üzere çeşitli türlerde (farklı kapasitelerde) araçlar kullanırlar. Bu problem Heterojen Araç-filolu ÇARP (HAÇARP) olarak bilinir. Literatürde bütün araç türlerinin aynı olduğu (kapasiteleri aynı olan) standart ÇARP ile ilgili birçok çalışma olmasına rağmen, bildiğimiz kadarıyla bu problemle ilgili literatürde sadece iki çalışma vardır (Salhi ve Sari, 1997; Salhi vd., 2014). Salhi vd., (2014) HAÇARP için matematiksel model geliştirmişler ve CPLEX kullanarak problemin alt ve üst sınırlarını bulmuşlardır. Bu çalışmada ayrıca DKA sezgiselini probleme etkili bir şekilde uygulanmışlar ve literatürdeki 26 veri örneğinin (Salhi ve Sari, 1997) 23 tanesi için yeni en iyi sonuçları bulmuşlardır.

2.2 GHAÇARP Literatür Özeti

Toplamalı Dağıtım Araç Rotalama Problemi (TDARP) klasik Araç Rotalama Problemi (ARP)'nin bir uzantısıdır. Bu problemde, araçlar müşterilere ürün dağıtımının yanında müşterilerden ürün toplarlar ve bu ürünleri depoya geri getirirler. TDARP klasik ARP'nin geliştirilmiş hali olduğundan NP-Zor bir problemdir (Nagy ve Salhi, 2005).

Çoğu araştırmacı araçların bütün dağıtım müşterilerini (linehaul) gezdikten sonra toplama yapacakları müşterilere (backhaul) uğrayabileceklerini varsayımlardır. Bunun sebebinin araçtaki toplanan ve dağıtılacak malların düzenlenmesindeki zorluk ve bu uğraştan doğacak zaman kaybı olarak göstermişlerdir. Bu varsayımın yapılmadığı durumda iki farklı metot vardır: aynı anda toplama dağıtım (simultaneous pickup and delivery), karışık toplama dağıtım (mixed pickup and delivery). Aynı anda toplama durumunda, müşteriler aynı anda mal gönderebilir ve alabilirler. Karışık toplama durumunda müşteriler ya dağıtım noktasıdır ya da toplama noktasıdır, aynı anda hem toplama hem dağıtım noktası olamazlar. Böyle bir durumda, bu müşteri hem dağıtım hem toplama müşterisi olarak düşünülür ve iki kez ziyaret edilir. Karışık toplama olarak isimlendirilmesindeki amaç rotadaki toplama ve dağıtım ziyaretlerinin istenilen sırada olabilmesidir. Özetle, TDARP literatürü üç temel kategoriye ayrılabilir: (i) Aynı-anda Toplama Dağıtım (ATD); (ii) Karışık Toplama Dağıtım (KTD); (iii) Önce Dağıtım Sonra Toplama (ÖDST) (Salhi ve Nagy, 1999). Önce dağıtım sonra toplama ve karışık toplama dağıtım yöntemleri birleştirildiğinde problem Geri-toplamalı ARP (GARP) olarak adlandırılır.

ATD yöntemi son zamanlarda çok ilgi çeken bir ARP problemi olmuştur. Min (1989) taşımacılık kitaplarındaki çok yaygın olan bir problemi (bir depo, iki araç ve 22 müşteri) ATD yöntemine göre çözmüştür. Müşteriler ilk olarak iki sınıfa ayrılmış, daha sonra iki adet Gezgin Satıcı Problemi (GSP) çözülmüştür. Mümkün olmayan (infeasible) çözümler aşırı yüklemenin olduğu arklara ceza vererek ve mümkün olmayan GSP'ni tekrar çözerek mümkün bir çözüm olması sağlanmıştır. Dethloff (2001) ATD yöntemine dayanan TDARP'nin geri lojistikte (reverse logistic) uygulamasını tanımlamış ve araçta kalan yük fikrini kullanan ekleme temelli sezgisel önermiştir. Crispim ve Brandao (2005), Chen ve Wu (2006) ve Tang ve Galvao (2006) ATD yöntemine dayanan TDARP için üst-sezgiseller (genellikle TA) önermişler, DellAmico v.d. (2006) dal fiyat metoduna dayanan kesin çözüm yöntemini önermişlerdir. Bianchessi ve Righini (2007) ATD yöntemine dayanan TDARP için karmaşık ve değişken komşuluk yaklaşımını kullanan Tabu Arama (TA) algoritmasını

önermişlerdir. Ganesh ve Narendran (2007) TDARP için müşterileri yakınlıklarına (proximity) göre kümeleyen sonrada küçültme-sarma algoritması (sink-wrap algorithm) kullanarak rotalara atayan ve genelleştirilmiş atama yaklaşımı (generalized assignment procedure) kullanarak araçları yerleştiren çok aşamalı yapısal bir sezgisel önermişlerdir. Bu sezgisel yöntemle iyi bir başlangıç çözümü elde etmişler ve Genetik Algoritmada (GA) kullanarak en iyi çözümü aramışlardır. Bu çalışmada ayrıca ilk olarak ATD ve ÖDST yöntemi aynı anda kullanılmıştır. Gajpal ve Abad (2009) ATD yöntemine dayanan TDARP için Karınca Kolonisi Sistemi (KKS) algoritmasını önermişler; değiştirme, ekleme ve alt-yolları (sub-paths) değiştirme operasyonlarına dayanan iki adet çok rotalı yerel arama yöntemini kullanmışlardır.

Literatürde KTD yöntemini kullanan çok az sayıda çalışma bulunmaktadır. Deif ve Bodin (1984) KTD ve önce ÖDST yönteminin uygulandığı TDARP' ni çözmek için Clarke and Wright tasarruf sezgiselini uyarlamışlardır. Toplama (backhaul) müşterisinin rotaya eklenmesini ertelemek için ceza katsayısı ekleyerek dağıtım (linehaul) ve toplama müşterileri arasındaki tasarrufu hesaplamışlardır. Golden v.d. (1985) dağıtım müşterilerinin oluşturduğu rotalara toplama müşterisi ekleme yöntemini geliştirmişlerdir. Cosco v.d. (1988) tasarruf metodu ve yüke bağlı ekleme metodunu birleştirerek TDARP'yi ÖDST ve KTD yöntemlerini kullanarak çözmüşlerdir. Bu yöntemde toplama müşterileri için araçta kalan yüke bağlı ekleme maliyeti hesaplamışlardır. Golden v.d. (1985), Cosco v.d. (1988) ve Salhi ve Nagy (1999) Clarke and Wright tasarruf sezgiselini ekleme temelli bir sezgiselle birleştirmişlerdir. Bu çalışmalarda ilk olarak sadece dağıtım müşterilerinden oluşan Kapasiteli ARP (KARP) çözülmüş, daha sonra rotaya toplama müşterileri eklenmiştir. Bu çalışmalar var olan rotaya toplama müşterisi ekleme yöntemi bakımından farklılıklar göstermektedir. Golden v.d. (1985) toplama müşterisi eklendikten sonra kalan dağıtım müşterilerinin sayısını dikkate alırken, Cosco v.d. (1988) toplama müşterisi eklendikten sonra dağıtılması gereken kalan toplam yükü dikkate alır. Salhi ve Nagy (1999) benzer bir yöntemi uygulamışlar, fakat rotaya bir seferde iki toplama müşterisi eklemişlerdir.

Toth ve Vigo (1996) ÖDST yöntemini TDARP' ye uygulamışlar ve problemi çözmek için Önce Kümele Sonra Rotala (ÖKSR) sezgisel algoritmasını geliştirmişlerdir. Wade ve Salhi (2004) çalışmalarında ÖDST ve KTD yöntemine dayanan TDARP için Karınca Kolonisi algoritmasını tasarlamışlardır.

Toth ve Vigo (1997) GARP için Tamsayı Doğrusal Programlama (TDP) modeli için Lagrangian gevşetmesi yöntemiyle alt sınırı hesaplayan dal-sınır algoritmasını önermişler, daha sonra bu alt sınırı uzay kesme (cutting plane) yöntemiyle güçlendirmişlerdir. Mingozzi v.d. (1999) aynı problemin 100 müşteriye kadar olan problem örneklerinin kesin çözümlerini dal-sınır algoritması kullanarak bulmuşlardır. GARP için son zamanlarda birçok sezgisel yöntem önerilmiştir. Halse (1992) çalışmasında GARP ve TDARP' yi içeren çok sayıda ARP türünü incelemiştir. TDARP problemini çözmek için ÖKSR yöntemini kullanmış ve rotalama aşamasında 3-opt algoritmasını kullanmıştır. TDARP için 100 müşteriye kadar, GARP için 150 müşteriye kadar çözümleri raporlamıştır.

2.3 HAÇARP Sezgisel Yöntemler Literatür Özeti

Bu bölümde Çok-depolu ARP (ÇARP) problemi ile Heterojen Araç-filolu ARP (HAARP)' nin gerçek hayatta daha sık karşılaşılan ama daha karmaşık bir hali olan Heterojen Araç-filolu Çok-depolu ARP (HAÇARP) için geliştirilen sezgisel yöntemlerin literatür taraması özetlenmiştir.

Çok-depolu ARP (ÇARP) için literatürde geliştirilen algoritmaların çoğu sezgisel algoritmalar. Tillman (1969) ÇARP için tasarruf kısıdını (Clarke ve Wright, 1964) kullanan ilk sezgiseli önermiştir. Wren ve Holliday (1972) süpürme algoritması kullanan bir sezgisel önermiştir. Gillet ve Johnson (1976) müşterileri en yakın depo etrafında ayırık kümeler oluşturacak şekilde gruplamışlar ve sonra her depoya süpürme algoritması uygulamışlardır.

Golden vd. (1977) ÇARP için iki sezgisel metot önermişlerdir. İlk metotta Yellow (1970) tarafından önerilen tasarruf kısıdı, ikinci metotta ise büyük boyutlu problemleri çözmek için Önce Kümele Sonra Rotala (ÖKSR) yöntemi kullanılmıştır. Raft (1982) ve Ball vd. (1983) Önce Rotala Sonra Kümele (ÖRSK) yöntemini kullanmışlardır. Raft (1982) aynı zamanda problemi daha küçük alt problemlere ayırıştıran bir modüler yaklaşım önermiştir. Müşteriler ilk olarak araç rotalarına atanmış, rotalar daha sonra en yakın depoya atanmıştır.

Chao vd. (1993) çok aşamalı sezgisel tanımlamışlardır. Bu sezgisel ilk olarak müşterileri en yakın depoya atar, sonra tasarruf kısıdı kullanarak her depoda araç rotalama yapar ve son olarak müşterilerin rotalarını değiştirerek çözümü iyileştirir. Potvin ve Rousseau (1993) müşterileri depolara atamak için Chao vd. (1993) 'nin önerdiği sezgisel birkaç yeni fikir eklemiştir.

Renaud vd. (1996) rota ve kapasite kısıtları olan ÇARP 'yi çözmek için Tabu Arama (TA) kullanan yeni bir sezgisel önermişlerdir. Cordeau vd. (1997) aynı zamanda ÇARP için TA sezgiselini kullanmışlardır. Salhi ve Sari (1997) üç aşamalı bir sezgisel önermişlerdir. İlk aşamada mümkün bir başlangıç çözümü oluşturulmuş, ikinci aşamada her bir depodaki rotalar iyileştirilmiş ve son aşamada bütün depolardaki rotalar iyileştirilmiştir. Thangiah ve Salhi (2001) genetik kümeleme sezgiseli geliştirmişlerdir.

Geetha vd. (2012) ÇARP'yi çözmek için ÖKSR metodunu uygulamışlar ve Genetik Algoritma (GA), Parçacık Sürüsü Optimizasyonu (PSO) gibi akıllı sezgiseller kullanmışlardır. Bu çalışmada aynı zamanda Melez PSO (MPSO) önerilmiştir. Ho vd. (2008) ÇARP için başlangıç çözümleri farklı olan iki tane Melez GA (MGA) önermişlerdir. Önerilen MGA sezgisellerinin başlangıç çözümleri rastsal olarak ve Clarke ve Wright tasarruf kısıdını en yakın komşu sezgiseli ile birleştirerek oluşturulmuştur. Mirabi vd. (2010) ÇARP 'yi çözmek için üç tane melez sezgisel önermişlerdir. Bu sezgiseller sırasıyla belirleyici, stokastik ve Tavlama Benzetimi (TB) geliştirme metotlarını kullanırlar. Liu vd. (2010) uygulanabilir büyük boyutlu ÇARP 'yi çözmek için boş araç değişimlerini en küçükleyen iki aşamalı Açgözlü Algoritma (AA) önermişlerdir. Önerilen algoritmanın ilk aşamasında AA kullanılarak başlangıç rotaları oluşturulmuş, ikinci aşamada ise yerel arama metotları kullanılarak başlangıç çözümü iyileştirilmiştir. Montoya-Torres vd. (2015) ÇARP' nin zaman pencereli, heterojen filolu, atlamalı toplamalı ve periyodik toplamalı gibi birçok türevi için detaylı bir literatür taraması yapmışlardır.

Aras vd. (2011) ürün kurtarma amacı olan bir firmanın günü geçmiş, bozuk ürünleri toplama problemini ÇARP olarak tanımlamışlardır. Bu problem için iki tane Karışık Tamsayılı Doğrusal Programlama (KTDP) modeli tanımlamışlardır. Orta ve büyük boyutlu problemleri çözmek için TA sezgiseli önermişlerdir. Gulczynski vd. (2011) iki ayrı ARP 'yi birleştirerek yeni bir ARP problemi olan Çok-depolu Bölünmüş Dağıtımli ARP'yi tanımlamışlardır. Bu yeni problem için tamsayılı programlamaya dayalı bir sezgisel önermişlerdir.

Kuo ve Wang (2012) Çok-depolu Yükleme Maliyetli ARP için Değişken Komşuluk Arama (DKA) sezgiselini önermişlerdir. Önerilen DKA sezgiseli üç aşamadan oluşmaktadır: İlk aşamada başlangıç çözümü oluşturmak için stokastik metot kullanılmaktadır, ikinci aşamada komşu çözümleri aramak için dört tane operatör rastgele seçilmektedir, üçüncü aşamada komşu

çözümün seçimi için TB sezgiseline benzer bir kriter kullanılmaktadır. Yücenur ve Demirel (2011) ÇARP için geometrik şekle dayalı yeni bir genetik kümeleme algoritması önermişlerdir. Önerilen kümeleme tekniğine dayanarak problem çözümünde kullanılmak üzere GA geliştirmişlerdir.

Bettinelli vd. (2011) zaman pencereli HAÇARP için ilk defa bir kesin çözüm yöntemi önermişlerdir. Çalışmada önerilen yöntem dal-kesme-fiyat (branch-cut-price), yöntemidir ve farklı fiyatlandırma kesme tekniklerini uygulamışlardır. Xu vd. (2012) çalışmalarında zaman pencereli HAÇARP' ı çözmek için DKA sezgiselini önermişlerdir. Xu ve Jiang (2014) bir önceki çalışmalarında önerdikleri yöntemi geliştirerek (improved) DKA sezgiselini önermişlerdir. İyi sonuçlar aldıkları geliştirilmiş DKA sezgiselini Çin' deki büyük su projesinde uygulamışlardır. Adelzadeh vd. (2014) çalışmalarında bulanık zaman pencereli HAÇARP problemi için iki amaçlı matematiksel model önermişler, toplam kat edilen mesafeyi azaltarak toplam maliyeti en küçükmeyi ve müşteri memnuniyetini arttırmayı amaçlamışlardır. Tanımladıkları problem için tavlama benzetimi sezgiselini kullanan çok aşamalı bir yöntem geliştirmişlerdir. Benslimane ve Benadada (2013) HAÇARP' yi çok miktarda tek tip ürünün müşterilere dağıtıldığı dağıtım problemine uygulamışlardır. Bu problem için Karınca Kolonisi Algoritması sezgiselini önermişlerdir. Mancini (2015) çalışmasında çok periyotlu HAÇARP problemini tanımlamış ve modellemiştir. Problem için Adaptif Geniş Komşuluk Arama Yöntemine dayanan matematiksel-sezgisel yaklaşımını önermiştir.

2.4 GHAÇARP Sezgisel Yöntemler Literatür Özeti

Bu bölümde, toplama ve dağıtma müşterileri olmak üzere iki tip müşterinin olduğu, çok depo ve heterojen araç filosunun bulunduğu GHAÇARP için yeni sezgisel yöntemler geliştirmeye temel olması amacıyla detaylı bir literatür taraması yapılmıştır.

Goetschalckx ve Jacobs-Blecha (1989) Geri-toplamalı ARP (GARP) için uzay doldurma (space-filling curves) yöntemine dayanan iki aşamalı sezgisel geliştirmişlerdir. Mosheiov (1998) tur parçalama yaklaşımına dayanan bir sezgisel önermiştir. Brandao (2006) çok aşamalı Tabu Arama (TA) algoritmasını geliştirmiş ve Wassan (2007) adaptif hafıza programlama ile geliştirilen reaktif TA algoritmasını önermiştir. Osman ve Wassan (2002) tabu süresi arama süresince güncellenen reaktif TA algoritmasını önermişlerdir. Zachariadis ve Kiranoudis (2012) arama uzayını genişleten bir yerel arama sezgiseli önermişlerdir. Cuervo vd. (2014) tekrarlı yerel arama algoritmasını

önermişlerdir. Yerel arama algoritması iki temel bileşenden oluşur: birinci bileşende algoritma bulunan çözümleri hafızada tutarak her tekrarda daha geniş bir uzayı tarar, ikinci bileşende ise mümkün olmayan çözümlere ceza atayarak düzenli olarak çözüm uzayının mümkün olan ve mümkün olmayan kısımlarında dolaşır. Bu çalışmada ayrıca iki karşılaştırmalı test örneğinin yeni en iyi çözümleri bulunmuştur.

Tütüncü, Carreto ve Baker (2009) yaptıkları çalışmada (Tütüncü vd., 2009) GARP ve türevleri için kullanıcıya (karar verici) çözüm sürecinin içinde yer alma olanağı sağlayan ve Açgözlü Rastgele Adaptif Hafıza Programlama Arama (ARAHPA) metoduna dayanan görsel etkileşimli bir yaklaşım önermişlerdir. Bu çalışmada ayrıca GARP ve türevlerini çözmek için önerilen görsel etkileşimli yaklaşımı kullanan bir Karar Destek Sistemi (KDS) önerilmiştir.

Taillard (1996) gerçek hayat dağıtım yönetiminde temel ARP 'ye göre daha yaygın olan Heterojen Sabit-filolu ARP (HSARP) 'yi tanımlamıştır. Bu problem Heterojen Araç-filolu ARP (HAARP) 'nin özel bir durumu olarak tanımlanabilir. HAARP için her araç türü sınırsız sayıda olurken HSARP için her araç türünden sabit sayıda bulunmaktadır. HSARP verilen araçlar için en iyi rotayı bulmayı amaçlarken HAARP en iyi araç filosu kombinasyonunu bulmayı amaçlar. Tütüncü (2010) HSARP ve bu problemin bir türevi olan tanımladığı Geri-toplamalı HSARP (GHSARP) 'yi çözmek için ARAHPA metoduna dayanan görsel etkileşimli metodu önermiştir. Tütüncü (2010) önerdiği bu yaklaşımı görsel bir KDS ile uygulayarak kullanıcıya HSARP hakkındaki bilgi ve deneyimlerini kullanarak alternatif kararlar üretme ve kararlar üzerinde değişiklikler yapabilme olanağı sağlamıştır.

Geri-toplamalı Çok-depolu ARP (GÇARP) için çok az sayıda çalışma bulunmaktadır. Salhi ve Nagy (1999) küme ekleme (cluster-insertion) kullanan ekleme temelli sezgisel GARP için geliştirmişler ve bu sezgisel çok-depolu probleme uyarlamışlardır. Çalışmalarında, Aynı-anda Toplama Dağıtma (ATD) ve Karışık Toplama Dağıtma (KTD) yöntemlerini de incelemişlerdir. Irnich (2000) çalışmasında Çok-depolu Toplamalı ve Dağıtılabilir ARP (ÇTDARP) 'nin özel bir türü olan Tek-merkezli Heterojen-araç-filolu Çok-depolu Toplama Dağıtma ARP' yi (THÇTDARP) tanımlamıştır. Bu problem için ayrıştırma (decomposition) algoritmasını tanımlamış ve büyük boyutlu problem örneklerini çözmüştür. Nagy ve Salhi (2005) ATD ve KTD yöntemlerine dayanan TDARP için etkili bir bileşik sezgisel yaklaşım geliştirmişler ve bu sezgisel çok-depolu probleme de uygulamışlardır. Önerilen sezgiseller TDARP için 1-5 depolu ve 50-249 müşteri problemler

için birkaç saniyede çok iyi kalitede sonuçlar vermiştir. Li v.d. (2015) çalışmalarında ilk defa Çok-depolu Aynı-anda Toplama Dağıtma ARP için yerel arama yöntemine dayanan bir üst-sezgisel geliştirmişlerdir. Yaptıkları deneylerde aynı problem için önerilen daha önceki çalışmaları geçmişlerdir. Ayrıca, önerdikleri üst-sezgiseli tek-depolu probleme de uygulamışlar ve literatürdeki birçok üst-sezgiselden daha iyi sonuçlar elde etmişlerdir.

3. GEREÇ VE YÖNTEM

3.1 Matematiksel Modeller

3.1.1 HAÇARP Matematiksel Modeli

Salhi v.d. (2014) HAÇARP için aşağıda detayları verilen Karışık Tamsayılı Doğrusal Programlama (KTDP) modelini önermişlerdir. Önerilen modelde düğüm temelli (node-based) modelleme yerine diğer araç rotalama problemlerini çözmede etkili olan (Yaman, 2006) akış temelli (flow-based) bir model önerilmiştir.

Verilen bir arkta ilerleyen ve bir depodan hareket eden aracın tipini tanımlayan dört indeksli bir ikili değişken kullanılmıştır. Bu değişken bir aracın çıktığı depoya geri dönmesini garantilemektedir. Bu kısıtlama üç indeksli bir değişken kullanılarak gevşetilebilmektedir.

Parametreler:

n : müşteri sayısı,

m : depo sayısı,

Bütün müşteriler ve depolar düğüm olarak düşünülmüştür $(1, \dots, n + m)$, burada m tane depo $n + 1, \dots, n + m$ olarak gösterilir.

q_i : i düğümünün talebi ($i = 1, \dots, n + m$) ve $i = n + 1, \dots, n + m$ için $q_i = 0$ 'dır,

K : araç türü sayısı,

Q_k : k araç türünün kapasitesi ($k = 1, \dots, K$),

f_k : k araç türünün sabit maliyeti ($k = 1, \dots, K$),

α_k : k araç türünün çalışma maliyeti ($k = 1, \dots, K$),

D_{ij} : i ve j düğümleri arasındaki uzaklık ($i, j = 1, \dots, n + m$).

Karar Değişkenleri:

$$x_{ijkd} = \begin{cases} 1, & \text{eğer } d \text{ deposundan } k \text{ kanve } (i, j) \text{ arkından geçen araç türü seçilirse,} \\ 0, & \text{diğer durumlarda.} \end{cases}$$

Burada $i, j = 1, \dots, n + m$; $k = 1, \dots, K$; $d = n + 1, \dots, n + m$ 'dir.

$y_{ij} = (i, j)$ arkından geçen aracın j düğümüne varmadan önce kalan toplam yük miktarı ($i, j = 1, \dots, n + m$).

Model:

$$\text{Min } Z = \sum_{d=n+1}^{n+m} \sum_{k=1}^K f_k \sum_{i=n+1}^{n+m} \sum_{j=1}^n x_{ijkd} + \sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} \alpha_k D_{ij} x_{ijkd} \quad (1)$$

$$\sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{i=1}^{n+m} x_{ijkd} = 1, j = 1, \dots, n \quad (2)$$

$$\sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{i=1}^{n+m} x_{ijkd} = 1, i = 1, \dots, n \quad (3)$$

$$\sum_{i=1}^{n+m} x_{ijkd} = \sum_{i=1}^{n+m} x_{jikd}, k = 1, \dots, K; j = 1, \dots, n + m; d = n + 1, \dots, n + m \quad (4)$$

$$\sum_{i=n+1}^{n+m} \sum_{j=1}^n y_{ij} = \sum_{j=1}^n q_j \quad (5)$$

$$\sum_{i=1}^{n+m} y_{ij} - \sum_{i=1}^{n+m} y_{ji} = q_j, j = 1, \dots, n \quad (6)$$

$$y_{ij} \leq \sum_{d=n+1}^{n+m} \sum_{k=1}^K Q_k x_{ijkd}, i = 1, \dots, n+m; j = 1, \dots, n \quad (7)$$

$$x_{d_1 i k d_2} = 0, i = 1, \dots, n; k = 1, \dots, K; d_1 \neq d_2 = n+1, \dots, n+m \quad (8)$$

$$x_{i d_1 k d_2} = 0, i = 1, \dots, n; k = 1, \dots, K; d_1 \neq d_2 = n+1, \dots, n+m \quad (9)$$

$$x_{ijkd} \in \{0, 1\}, i, j = 1, \dots, n+m; k = 1, \dots, K; d = n+1, \dots, n+m \quad (10)$$

$$y_{ij} \geq 0, i, j = 1, \dots, n+m \quad (11)$$

Amaç fonksiyonu (1) sabit araç maliyetini ve seyahat maliyetini içeren toplam maliyeti göstermektedir. Kısıt (2) her müşteriye sadece bir kez uğranacağını gösterir. Kısıt (3) akışın korunmasını sağlar yani bir müşteriye uğrayan araç oradan ayrılmak zorundadır. Kısıt (4) herhangi bir depodan hareket eden en fazla bir aracın $i - j$ arkından geçmesini garantiler. Kısıt (5) bütün depolardan çıkan toplam yük miktarının bütün müşterilerin talepleri toplamına eşit olduğunu gösterir ve (6) j müşterisine uğradıktan sonra araçta kalan yük miktarının, aracın bu müşteriye ziyaret etmeden önceki yük miktarı eksi müşterinin talebine eşit olmasını garantiler. Kısıt (7) herhangi araç türü için araç kapasitesinin aşılmamasını sağlar. (8) ve (9) kısıtları bir depodan ayrılan ya da bir depoya dönmekte olan bir aracın farklı bir depoya gitmemesini garantiler. (10) ve (11) kısıtları ise sırasıyla ikili karar değişkenini ve negatif olmayan karar değişkenini gösterirler.

Arama uzayını azaltarak yukarıda verilen modeli daraltan kısıtlar aşağıda verilmiştir. Arama uzayı geçerli eşitlikler tanımlanarak (örneğin, (7) yerine (12) ve (13) tanımlanmıştır) ve problemin yapısından dolayı bulunmasına gerek duyulmayan bazı karar değişkenlerinin en iyi değerleri (14)-(18) kısıtlarında sifıra eşitlenmiştir.

$$y_{ij} \leq \sum_{d=n+1}^{n+m} \sum_{k=1}^K (Q_k - q_i) x_{ijkd}, i = 1, \dots, n+m; j = 1, \dots, n \quad (12)$$

$$y_{ij} \geq \sum_{d=n+1}^{n+m} \sum_{k=1}^K q_j x_{ijkd}, i, j = 1, \dots, n \quad (13)$$

$$y_{ij} = 0, i = 1, \dots, n; j = n + 1, \dots, n + m \quad (14)$$

$$y_{ij} = 0, i = n + 1, \dots, n + m; j = n + 1, \dots, n + m \quad (15)$$

$$y_{ii} = 0, i = 1, \dots, n \quad (16)$$

$$x_{ijkd} = 0, d = n + 1, \dots, n + m; k = 1, \dots, K; i, j = n + 1, \dots, n + m \quad (17)$$

$$x_{iikd} = 0, d = n + 1, \dots, n + m; k = 1, \dots, K; i = 1, \dots, n + m \quad (18)$$

Kısıt (12) herhangi (i, j) arki üzerindeki yük miktarının i müşterisine dağıtım yapmış ve herhangi bir depodan hareket etmiş bir aracın kalan yük miktarından fazla yük alamayacağını ifade etmektedir. Fark ederseniz modeli basitleştirmek için her deponun talebi sıfıra eşitlenmiştir. Kısıt (13) (i, j) arki üzerinde j düğümüne doğru hareket ederken en azından j düğümüne yetecek kadar yük kalmasını garantiler. Kısıt (14) bütün araçların depolara boş olarak dönmesini garantilerken (15) ve (16) numaralı kısıtlar sırasıyla depodan depoya ve müşterinin kendisine yük taşınmasını engeller. Kısıt (17) ve (18) sırasıyla depolar arasında ve müşterilerin kendilerine doğru herhangi bir araç türünü kullanarak ark oluşmasını engeller.

Problemimize ayrıca i düğümünden ayrıldıktan sonra kalan toplam talebi ifade eden bir tam sayılı değişken (u_i) kullanan alternatif düğüm tabanlı modeller uyarlanabilir. Bunlar alt tur eleme kısıtları ve kapasite kısıtlarını sağlamak için kullanılır.

Proje kapsamında geliştirilen Karar Destek Sistemi (KDS) birçok HAÇARP türünü de çözebilmek üzere tasarlanmıştır. Verilen model bazı uygulanabilir problem türleri için aşağıdaki şekillerde düzenlenebilir:

Eğer bir araç türündeki araç sayısı, N_k , biliniyorsa aşağıdaki kısıt modele eklenebilir ve KDS sisteminde kullanıcıdan bu bilgi girdi olarak istenebilir.

$$\sum_{d=n+1}^{n+m} \sum_{i=n+1}^{n+m} \sum_{j=1}^n x_{ijkd} \leq N_k, k = 1, \dots, K \quad (19)$$

Eğer bir depodaki toplam araç sayısı, M_i , biliniyorsa aşağıdaki kısıt modele eklenebilir ve KDS için bir girdi olabilir.

$$\sum_{k=1}^K \sum_{j=1}^n x_{ijki} \leq M_i, i = n + 1, \dots, n + m \quad (20)$$

Eğer i deposunda bulunan k araç türü, M_{ik} , sınırlı sayıdaysa aşağıdaki kısıt modele eklenebilir ve KDS ile birleştirilebilir.

$$\sum_{j=1}^n x_{ijki} \leq M_{ik}, i = n + 1, \dots, n + m; k = 1, \dots, K \quad (21)$$

Eğer verilen bir d_1 deposunda, $d = n + d_1$, bazı araç türleri kümesinin, $V(d)$, bulunması istenmiyorsa aşağıdaki kısıt modele eklenebilir. KDS sisteminde her depodaki her bir araç için başlangıç müşterileri seçilerek rotalar oluşturulur. Bundan dolayı, $V(d)$ içinde araçlar için rota oluşturulmazsa bu kısıt sağlanmış olur.

$$x_{djkd} = 0, k \in V(d) \subset \{1, \dots, K\}; j = 1, \dots, n \quad (22)$$

Eğer bir aracın dağıtımına başladığı depoya geri dönme zorunluluğu olmazsa problem çözülmesi daha kolay bir hal alır. Verilen modeldeki (1)-(17) formülleri aynı kalmak üzere 3 indisli bir karar değişkeni, x_{ijk} , kullanılarak bu koşul sağlanmış olur.

Eğer probleme rota uzunluğu (TD) kısıdı eklenirse aşağıdaki kısıtlar modele sürekli, negatif olmayan yeni bir karar değişkeni t_{ijd} ile eklenir. Bu değişken d deposundan j müşterisine $i - j$ arkını kullanarak yapılan en kısa seyahat uzunluğunu ifade eder. Bu kısıtta KDS sistemine kısıt olarak kolaylıkla eklenebilir.

$$\sum_{j \neq i}^{n+m} t_{ijd} - \sum_{j \neq i}^{n+m} t_{jid} = \sum_{j=1}^{n+m} D_{ij} \sum_{k=1}^K x_{ijkd}, \forall i = 1, \dots, n; d = n + 1, \dots, n + m \quad (23)$$

$$t_{ijd} \leq (TD - D_{jd}) \sum_{k=1}^K x_{ijkd}, i \neq j = 1, \dots, n; d = n + 1, \dots, n + m \quad (24)$$

$$t_{idd} \leq TD \sum_{k=1}^K x_{idkd}, i = 1, \dots, n; d = n + 1, \dots, n + m \quad (25)$$

$$t_{ijd} \geq (D_{ij} + D_{di}) \sum_{k=1}^K x_{ijkd}, i \neq j = 1, \dots, n; d = n + 1, \dots, n + m \quad (26)$$

$$t_{did} = D_{di} \sum_{k=1}^K x_{dikd}, i = 1, \dots, n; d = n + 1, \dots, n + m \quad (27)$$

$$t_{ijd} = 0, i, j = n + 1, \dots, n + m; d = n + 1, \dots, n + m \quad (28)$$

$$t_{ijd} \geq 0, i, j = 1, \dots, n + m; d = n + 1, \dots, n + m \quad (29)$$

3.1.2 GHAÇARP Matematiksel Modeli

Bu bölümde ARP' nin yeni bir türü olan GHAÇARP tanımlanmıştır. Klasik ARP' de tek bir depoda talepleri bilinen (sadece dağıtım) bir müşteri kümesi, homojen araçların (sınırsız sayıda) olduğu bir araç filosu tanımlanır. Problemin amacı, en az maliyeti veren aynı depodan çıkan ve aynı depoda sonlanan araç rotalarını araçların kapasite kısıdını ve eğer varsa rota uzunluğu kısıdını dikkate alarak bulmaktır. GHAÇARP ise Klasik ARP'nin daha karmaşık bir versiyonudur ve ilk defa bu proje kapsamında tanımlanarak ele alınacaktır. Literatüre bakıldığında GHAÇARP Salhi v.d. (2013) tanımladığı Geri-toplamalı Heterojen Araç-filolu ARP (GHAARP/Fleet Size and Mix VRP with Backhauls) ve Salhi ve Sari (1997) tanımladığı HAÇARP (Multi-depot VRP with Heterogeneous Vehicle Fleet) modelleri birleştirilerek oluşturulmuştur. Oluşturulan modelin özellikleri özetle aşağıdaki gibi sıralanabilir:

- Müşteriler iki gruba ayrılmıştır: dağıtım (linehaul) ve toplama (backhaul) müşterileri.
- Birden fazla depo vardır ve her depoda sabit maliyetleri kapasitesine göre değişen, değişken maliyetleri olan heterojen bir araç filosu (her araç tipinden sınırsız sayıda) vardır.

- Dağıtım müşterileri tamamlanmadan toplama müşterilerine gidilmez.
- Sadece toplama müşterilerinin olduğu bir rotaya izin verilmezken, sadece dağıtım müşterilerinin olduğu rotalar olabilir.
- Araç kapasite kısıdı uygulanmıştır. Rota uzunluğu ve her bir araç tipi için araç sayısı kısıdı eklenebilir, fakat bu çalışmada bu kısıtlar kullanılmamıştır.

Bu bölümde, projemizin ilk döneminde oluşturulan GHAÇARP modeli verilecektir. Söz konusu matematiksel programlama modeli Salhi v.d. (2014) ve Salhi v.d. (2013) çalışmalarındaki matematiksel modeller dikkate alınarak oluşturulmuştur. Modelimizin, parametreleri ve detayları aşağıdaki gibidir.

Parametreler:

n : müşteri sayısı, $(1, \dots, n + m)$

m : depo sayısı, $(n + 1, \dots, n + m)$

l : dağıtım müşteri sayısı, $(1, \dots, l)$

b : toplama müşteri sayısı, $(l + 1, \dots, n)$

Bütün müşteriler ve depolar düğüm olarak düşünülmüştür $(1, \dots, n + m)$, burada m tane depo $n + 1, \dots, n + m$ olarak, l tane dağıtım müşterisi $1, \dots, l$ olarak ve $b = n - l$ tane toplama müşterisi $l + 1, \dots, n$ olarak gösterilir.

q_i : i düğümünün talebi ($i = 1, \dots, n + m$) ve $i = n + 1, \dots, n + m$ için $q_i = 0$ 'dır,

p_i : i düğümünün arzı ($i = 1, \dots, n + m$) ve $i = n + 1, \dots, n + m$ için $p_i = 0$ 'dır,

K : araç tipi sayısı,

Q_k : k tipi araçların kapasitesi ($k = 1, \dots, K$),

f_k : k tipi araçların sabit maliyeti ($k = 1, \dots, K$),

α_k : k tipi araçların çalışma maliyeti ($k = 1, \dots, K$),

D_{ij} : i ve j düğümleri arasındaki uzaklık ($i, j = 1, \dots, n + m$).

Karar Değişkenleri:

$$x_{ijkd} = \begin{cases} 1, & \text{eğerdeposundankalkanve}(i, j) \text{ arkındangeçenkti paraç seçilirse,} \\ 0, & \text{diğerdurumlarda.} \end{cases}$$

Burada $i, j = 1, \dots, n + m$; $k = 1, \dots, K$; $d = n + 1, \dots, n + m$ 'dir.

$y_{ij} = (i, j)$ arkından geçen aracın j düğümüne varmadan önce kalan toplam yük miktarı ($i, j = 1, \dots, n + m$).

Model:

$$\text{Min}Z = \sum_{d=n+1}^{n+m} \sum_{k=1}^K f_k \sum_{i=n+1}^{n+m} \sum_{j=1}^l x_{ijkd} + \sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{i=1}^{n+m} \sum_{j=1}^{n+m} \alpha_k D_{ij} x_{ijkd} \quad (1)$$

$$\sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{i=1}^{n+m} x_{ijkd} = 1, \quad j = 1, \dots, n \quad (2)$$

$$\sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{j=1}^{n+m} x_{ijkd} = 1, \quad i = 1, \dots, n \quad (3)$$

$$\sum_{i=1}^{n+m} x_{ijkd} = \sum_{i=1}^{n+m} x_{jikd}, \quad k = 1, \dots, K; j = 1, \dots, n + m; d = n + 1, \dots, n + m \quad (4)$$

$$\sum_{i=n+1}^{n+m} \sum_{j=1}^l y_{ij} = \sum_{j=1}^l q_j \quad (5)$$

$$\sum_{i=l+1}^n \sum_{j=n+1}^{n+m} y_{ij} = \sum_{i=l+1}^n p_i \quad (6)$$

$$\sum_{i=1}^l y_{ij} + \sum_{i=n+1}^{n+m} y_{ij} = \sum_{i=1}^{n+m} y_{ji} + q_j, \quad j = 1, \dots, l \quad (7)$$

$$\sum_{i=l+1}^n y_{ji} + \sum_{i=n+1}^{n+m} y_{ji} = \sum_{i=1}^n y_{ij} + p_j, \quad j = l+1, \dots, n \quad (8)$$

$$y_{ij} \leq \sum_{d=n+1}^{n+m} \sum_{k=1}^K Q_k x_{ijkd}, \quad i \neq j = 1, \dots, n+m \quad (9)$$

$$y_{ij} = 0, \quad i = 1, \dots, l; i = n+1, \dots, n+m; j = l+1, \dots, n+m \quad (10)$$

$$y_{ij} = 0, \quad i = n+1, \dots, n+m; j = n+1, \dots, n+m \quad (11)$$

$$y_{ii} = 0, \quad i = 1, \dots, n \quad (12)$$

$$x_{d_1 i k d_2} = 0, i = 1, \dots, n; k = 1, \dots, K; d_1 \neq d_2 = n+1, \dots, n+m \quad (13)$$

$$x_{i d_1 k d_2} = 0, i = 1, \dots, n; k = 1, \dots, K; d_1 \neq d_2 = n+1, \dots, n+m \quad (14)$$

$$x_{d j k d} = 0, \quad j = l+1, \dots, n; k = 1, \dots, K; d = n+1, \dots, n+m \quad (15)$$

$$x_{i j k d} = 0, \quad i = l+1, \dots, n; j = 1, \dots, l; k = 1, \dots, K; d = n+1, \dots, n+m \quad (16)$$

$$x_{i j k d} \in \{0, 1\}, i, j = 1, \dots, n+m; k = 1, \dots, K; d = n+1, \dots, n+m \quad (17)$$

$$y_{ij} \geq 0, i, j = 1, \dots, n+m \quad (18)$$

Modelin amaç fonksiyonu olan (1) eşitliği toplam maliyeti en küçüklemeyi amaçlamaktadır. Eşitliğin birinci kısmında kullanılan her bir araç türü için sabit maliyeti eklemekte, ikinci kısmındaysa araçların toplam kat ettiği mesafeyi değişken maliyetle çarparak toplam maliyete eklemektedir. Kısıtlar (2) ve (3), her müşterinin sadece bir araçla ve sadece bir kez ziyaret edilmesi gerektiğini gösterirken, kısıt (4) her bir rotanın sürekliliğini ve sadece bir araç türüyle tamamlanmasını garantilemektedir. Kısıt (5) depolardan dağıtım müşterilerine giden toplam yük miktarının dağıtım müşterilerinin toplam talebine eşit olmasını sağlarken, kısıt (6) dağıtım müşterilerinden depolara gelen toplam yük miktarının dağıtım müşterilerinin toplam arzına eşit

olmasını sağlar. (7) ve (8) kısıtları sırasıyla dağıtım ve toplam müşterilerine giren ve çıkan yük miktarı akışını kontrol eder. Her bir arktaki taşınan yük miktarının üst sınırı kısıt (9)' da o arktan geçen aracın kapasitesine eşitlenmiştir. Kısıt (10)' da dağıtım müşterileri ve depolardan toplama müşterilerine taşınan yük miktarı, kısıt (11)' de depolar arasında taşınan yük miktarı ve kısıt (12)' de müşterilerin kendi aralarında taşınan yük miktarları sıfıra eşitlenmiş ve böylelikle bu ihtimaller ortadan kaldırılmıştır. Kısıtlar (13) ve (14) araçların kalkış ve varış depolarının farklı olması durumunu engeller. Kısıt (15) araçların depolardan toplama müşterilerini gitmesini engellerken, kısıt (16) araçların toplam müşterilerinden dağıtım müşterilerine gitmesini engeller. Kısıt (17) ikili karar değişkenini, kısıt (18) ise sürekli negatif olmayan karar değişkenini tanımlar.

Salhi v.d. (2013) yaptıkları çalışmada GARP probleminin bazı kısıtlarını toplama ve dağıtım müşterileri için ayırarak modeli gevşetmeyi önemli bulmuşlar ve yaptıkları deneylerde gevşetilmiş modelle genellikle daha az CPU zamanında daha iyi Üst Sınır (ÜS) ve Alt Sınır (AS) değerlerini elde etmişlerdir. Bizde aynı gevşetme yöntemini elde ettiğimiz modele uyguladık. (2)-(4) kısıtlarındaki karar değişkenlerinin her zaman sıfır değerini alacağı kısımları modelden çıkardık. Ayrıca, kısıt (9) daki her bir arka taşınan yük miktarı kısıtlarını toplama ve dağıtım müşterileri için ayrı ayrı düşünerek tekrar tanımladık.

Kısıt (2) aşağıdaki gibi iki kısıt halinde yazılmıştır, burada birinci kısıt toplama müşterileri için ikinci kısıt ise dağıtım müşterileri için tanımlanmıştır.

$$\sum_{d=n+1}^{n+m} \left(\sum_{k=1}^K \sum_{i=1}^l x_{ijkd} + \sum_{k=1}^K \sum_{i=n+1}^{n+m} x_{ijkd} \right) = 1, \quad j = 1, \dots, l \quad (2a)$$

$$\sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{i=1}^n x_{ijkd} = 1, \quad j = l + 1, \dots, n \quad (2b)$$

Kısıt (3) de benzer şekilde iki ayrı kısıtla ifade edilmiştir ve birinci kısıt toplama müşterileri için ve ikinci kısıt dağıtım müşterileri için tanımlanmıştır.

$$\sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{j=1}^{n+m} x_{ijkd} = 1, \quad i = 1, \dots, l \quad (3a)$$

$$\sum_{d=n+1}^{n+m} \sum_{k=1}^K \sum_{j=l+1}^{n+m} x_{ijkd} = 1, \quad i = l+1, \dots, n \quad (3b)$$

Kısıt (4) toplama müşterileri, dağıtma müşterileri ve depolar için olmak üzere üç ayrı kısıda ayrılabilir:

$$\sum_{i=1}^l x_{ijkd} + \sum_{i=n+1}^{n+m} x_{ijkd} = \sum_{i=1}^{n+m} x_{jikd}, \quad k = 1, \dots, K; j = 1, \dots, l; d = n+1, \dots, n+m \quad (4a)$$

$$\sum_{i=1}^n x_{ijkd} = \sum_{i=l+1}^{n+m} x_{jikd}, \quad k = 1, \dots, K; j = l+1, \dots, n; d = n+1, \dots, n+m \quad (4b)$$

$$\sum_{j=n+1}^{n+m} \sum_{i=1}^l x_{jikd} = \sum_{i=1}^{n+m} \sum_{j=n+1}^{n+m} x_{ijkd}, \quad k = 1, \dots, K; d = n+1, \dots, n+m \quad (4c)$$

Kısıt (9) aşağıdaki dört kısıtla yer değiştirilebilir:

$$y_{dj} \leq \sum_{k=1}^K Q_k x_{djkd}, \quad j = 1, \dots, l, d = n+1, \dots, n+m \quad (9a)$$

$$y_{id} \leq \sum_{k=1}^K Q_k x_{idkd}, \quad i = l+1, \dots, n, d = n+1, \dots, n+m \quad (9b)$$

$$y_{ij} \leq \sum_{k=1}^K (Q_k - q_i) x_{ijkd}, \quad i \neq j = 1, \dots, l, d = n+1, \dots, n+m \quad (9c)$$

$$y_{ij} \leq \sum_{k=1}^K Q_k x_{ijkd}, \quad i \neq j = l + 1, \dots, n, d = n + 1, \dots, n + m \quad (9d)$$

Bu yeni gevşetilmiş model toplam kısıt sayısını azaltmış, fakat değişken sayısını arttırmıştır. Buna rağmen gevşetilmiş model genellikle daha az ya da aynı (3 saat) CPU zamanında daha iyi AS ve ÜS değerlerini elde ettiği görülmüş ve bu sonuçlar deney sonuçları kısmında raporlanmıştır.

Önerilen bu model literatürde çalışılan diğer geri toplamalı problem türlerine kolaylıkla uyarlanabilir. Örneğin, modelimizde hiçbir rotanın sadece toplama müşterilerinden oluşmasına izin verilmez, ancak bu kısıtlama kısıt (15) modelden çıkarılarak gevşetilebilir. Gevşetilmiş modelde kısıt (4) ve (9) için yapılan gevşetmeler yerine orijinal hallerini kullanmak yeterli olacaktır.

Bu modelde her bir araç tipinden sınırsız sayıda olduğu varsayılmıştır. Eğer her bir araç tipinin sayısı biliniyorsa, K_k , şu kısıtların $\sum_{d=n+1}^{n+m} \sum_{j=1}^l x_{djkd} \leq K_k$ yukarıda verilen modele eklenmesi gerekir. Bütün araçların sayısının, NV , bilindiği durumda $\sum_{d=n+1}^{n+m} \sum_{j=1}^l \sum_{k=1}^K x_{djkd} \leq NV$ kısıtı modele eklenmelidir. Ayrıca, bütün bu alternatif durumlar geliştirilecek olan Karar Destek Sisteminde kolaylıkla kullanıcıdan girdi olarak girmesi istenebilir.

3.2 Problem Örnekleri

3.2.1 HAÇARP Problem Örnekleri

Bu bölümde ilk olarak, HAÇARP problem örnekleri tanımlanacaktır. HAÇARP problem örnekleri ilk olarak Salhi ve Sari (1997) tarafından ve son olarak Salhi vd. (2014) tarafından kullanılmış ve tanımlanmıştır. Bu problem örnekleri toplam 26 problemden oluşmaktadır ve boyutu 50 müşteriden 360 müşteriye, 2 depodan 9 depoya değişmektedir. Her bir problemde toplam 5 farklı araç yer almaktadır ve 5 araç tipinin hepsi farklı kapasite, sabit ve değişken maliyetlere sahiptir.

Tablo 1. HAÇARP Problem Örnekleri

No	N	m	TD	Q1	f1	v1	Q2	f2	v2	Q3	f3	v3	Q4	f4	v4	Q5	f5	v5
1	55	4	0	60	80	0.8	80	90	0.9	100	100	1	120	110	1.1	140	120	1.2
2	85	3	0	60	80	0.8	80	90	0.9	100	100	1	120	110	1.1	140	120	1.2
3	85	3	0	96	80	0.8	128	90	0.9	160	100	1	192	110	1.1	224	120	1.2
4	50	4	0	48	80	0.8	64	90	0.9	80	100	1	96	110	1.1	112	120	1.2
5	50	4	0	96	80	0.8	128	90	0.9	160	100	1	192	110	1.1	224	120	1.2
6	75	5	0	84	80	0.8	112	90	0.9	140	100	1	168	110	1.1	196	120	1.2
7	100	2	0	60	80	0.8	80	90	0.9	100	100	1	120	110	1.1	140	120	1.2
8	100	2	0	120	80	0.8	160	90	0.9	200	100	1	240	110	1.1	280	120	1.2
9	100	3	0	60	80	0.8	80	90	0.9	100	100	1	120	110	1.1	140	120	1.2
10	100	4	0	60	80	0.8	80	90	0.9	100	100	1	120	110	1.1	140	120	1.2
11	249	2	310	300	80	0.8	400	90	0.9	500	100	1	600	110	1.1	700	120	1.2
12	249	3	310	300	80	0.8	400	90	0.9	500	100	1	600	110	1.1	700	120	1.2
13	249	4	310	300	80	0.8	400	90	0.9	500	100	1	600	110	1.1	700	120	1.2
14	249	5	310	300	80	0.8	400	90	0.9	500	100	1	600	110	1.1	700	120	1.2
15	80	2	0	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2
16	80	2	200	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2
17	80	2	180	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2
18	160	4	0	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2
19	160	4	200	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2
20	160	4	180	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2
21	240	6	0	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2
22	240	6	200	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2
23	240	6	180	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2
24	360	9	0	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2
25	360	9	200	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2
26	360	9	180	36	80	0.8	48	90	0.9	60	100	1	72	110	1.1	84	120	1.2

Problemler detaylı olarak Tablo 1’de verilmiştir. Burada, N toplam müşteri sayısını, m toplam depo sayısını, TD maksimum yol kısıdını ve daha sonraki her üç kolonda her bir aracın sırasıyla kapasitesi (Q), sabit maliyeti (f) ve değişken maliyeti (v) gösterilmektedir.

3.2.2 GHAÇARP Problem Örnekleri

Bu projede GHAÇARP örnek problemleri, Salhi v.d. (2013)’te Geri-toplamalı Heterojen Araç-filolu Araç Rotalama Problemi (GHAARP) için kullanılan problem örneklerine Salhi v.d. (2014)’te HAÇARP için kullanılan problem örneklerinin depo bilgileri eklenerek oluşturulmuştur. Önerilen problem örnekleri Golden v.d. (1984)’nin heterojen araç-filolu çok-depolu problem kümesine ve Toth ve Vigo (1997) geri-toplamalı problem kümesine dayanmaktadır.

Yeni veri kümesi içindeki GHAÇARP problemleri Tablo 2’de detaylı olarak verilmiştir. Bu tabloda, N toplam müşteri sayısını, L toplam dağıtım müşterileri sayısını, B toplam toplama müşterileri

sayısını, m toplam depo sayısını ve daha sonraki her üç kolonda her bir aracın sırasıyla kapasitesi (Q), sabit maliyeti (f) ve değişken maliyeti (v) gösterilmektedir.

Tablo 2. GHAÇARP Problem Örnekleri

Numara	N	L	B	m	Q1	f1	v1	Q2	f2	v2	Q3	f3	v3	Q4	f4	v4	Q5	f5	v5	Q6	f6	v6
TK1	20	10	10	2	20	20	17.2	30	35	21.2	40	50	25.2	70	120	26.2	120	225	28.4			
TK2	20	13	7	2	20	20	17.2	30	35	21.2	40	50	25.2	70	120	26.2	120	225	28.4			
TK3	20	16	4	2	20	20	17.2	30	35	21.2	40	50	25.2	70	120	26.2	120	225	28.4			
TK4	20	10	10	2	60	1000	17.2	80	1500	21.2	150	3000	28.4									
TK5	20	13	7	2	60	1000	17.2	80	1500	21.2	150	3000	28.4									
TK6	20	16	4	2	60	1000	17.2	80	1500	21.2	150	3000	28.4									
TK7	20	10	10	2	20	20	17.2	30	35	21.2	40	50	25.2	70	120	26.2	120	225	28.4			
TK8	20	13	7	2	20	20	17.2	30	35	21.2	40	50	25.2	70	120	26.2	120	225	28.4			
TK9	20	16	4	2	20	20	17.2	30	35	21.2	40	50	25.2	70	120	26.2	120	225	28.4			
TK10	20	10	10	2	60	1000	17.2	80	1500	21.2	150	3000	28.4									
TK11	20	13	7	2	60	1000	17.2	80	1500	21.2	150	3000	28.4									
TK12	20	16	4	2	60	1000	17.2	80	1500	21.2	150	3000	28.4									
TK13	50	25	25	3	20	20	10.8	30	35	17.2	40	50	21.2	70	120	25.2	120	225	26.2	200	400	28.4
TK14	50	33	17	3	20	20	10.8	30	35	17.2	40	50	21.2	70	120	25.2	120	225	26.2	200	400	28.4
TK15	50	40	10	3	20	20	10.8	30	35	17.2	40	50	21.2	70	120	25.2	120	225	26.2	200	400	28.4
TK16	50	25	25	3	120	1000	17.2	160	1500	21.2	300	3500	28.4									
TK17	50	33	17	3	120	1000	17.2	160	1500	21.2	300	3500	28.4									
TK18	50	40	10	3	120	1000	17.2	160	1500	21.2	300	3500	28.4									
TK19	50	25	25	4	50	100	17.2	100	250	21.2	160	450	28.4									
TK20	50	33	17	4	50	100	17.2	100	250	21.2	160	450	28.4									
TK21	50	40	10	4	50	100	17.2	100	250	21.2	160	450	28.4									
TK22	50	25	25	4	40	100	17.2	80	200	21.2	140	400	28.4									
TK23	50	33	17	4	40	100	17.2	80	200	21.2	140	400	28.4									
TK24	50	40	10	4	40	100	17.2	80	200	21.2	140	400	28.4									
TK25	75	37	38	5	50	25	17.2	120	80	21.2	200	150	26.2	350	320	28.4						
TK26	75	50	25	5	50	25	17.2	120	80	21.2	200	150	26.2	350	320	28.4						
TK27	75	60	15	5	50	25	17.2	120	80	21.2	200	150	26.2	350	320	28.4						
TK28	75	37	38	5	20	10	10.8	50	35	17.2	100	100	21.2	150	180	25.2	250	400	26.2	400	800	28.4
TK29	75	50	25	5	20	10	10.8	50	35	17.2	100	100	21.2	150	180	25.2	250	400	26.2	400	800	28.4
TK30	75	60	15	5	20	10	10.8	50	35	17.2	100	100	21.2	150	180	25.2	250	400	26.2	400	800	28.4
TK31	100	50	50	4	100	500	17.2	200	1200	21.2	300	2100	28.4									
TK32	100	66	34	4	100	500	17.2	200	1200	21.2	300	2100	28.4									
TK33	100	80	20	4	100	500	17.2	200	1200	21.2	300	2100	28.4									
TK34	100	50	50	3	60	100	17.2	140	300	21.2	200	500	28.4									
TK35	100	66	34	3	60	100	17.2	140	300	21.2	200	500	28.4									
TK36	100	80	20	3	60	100	17.2	140	300	21.2	200	500	28.4									

N: Toplam müşteri sayısı,

L: Toplam dağıtım müşterileri sayısı,

B: Toplam toplama müşterileri sayısı,

m: Toplam depo sayısı,

Qi: i tipi aracın aracın kapasitesi,

fi: i tipi aracın sabit maliyeti,

vi: i tipi aracın değişken maliyeti.

3.3 KDS Tanıtımı

Günümüz araç rotalama problemleri için geliştirilen çözümlere bakıldığında, kullanıcı etkileşimli çözümlerin arttığını söyleyebiliriz. Raporumuzun bu bölümünde, projenin araştırma konusu olan Heterojen Araç-filolu Çok-depolu Araç Rotalama Probleminin (HAÇARP) çözümü için geliştirilen KDS' nin genel yapısı anlatılacaktır. Geliştirilen KDS' nin adı yürütücünün daha önceki çalışmasına (Tütüncü v.d. (2009), Tütüncü (2010)) istinaden ADVISER2 olarak belirlenmiştir. ADVISER2' nin amacı kullanıcının yeni fikirler üretmesi ve alternatif fikirleri incelemesi için kolaylık sağlamaktadır. Ayrıca, ADVISER2 ile KDS için gerekli olan esneklik, kullanıcı uyumluluğu ve etkinliği kullanıcıya sunmak amaçlanmaktadır.

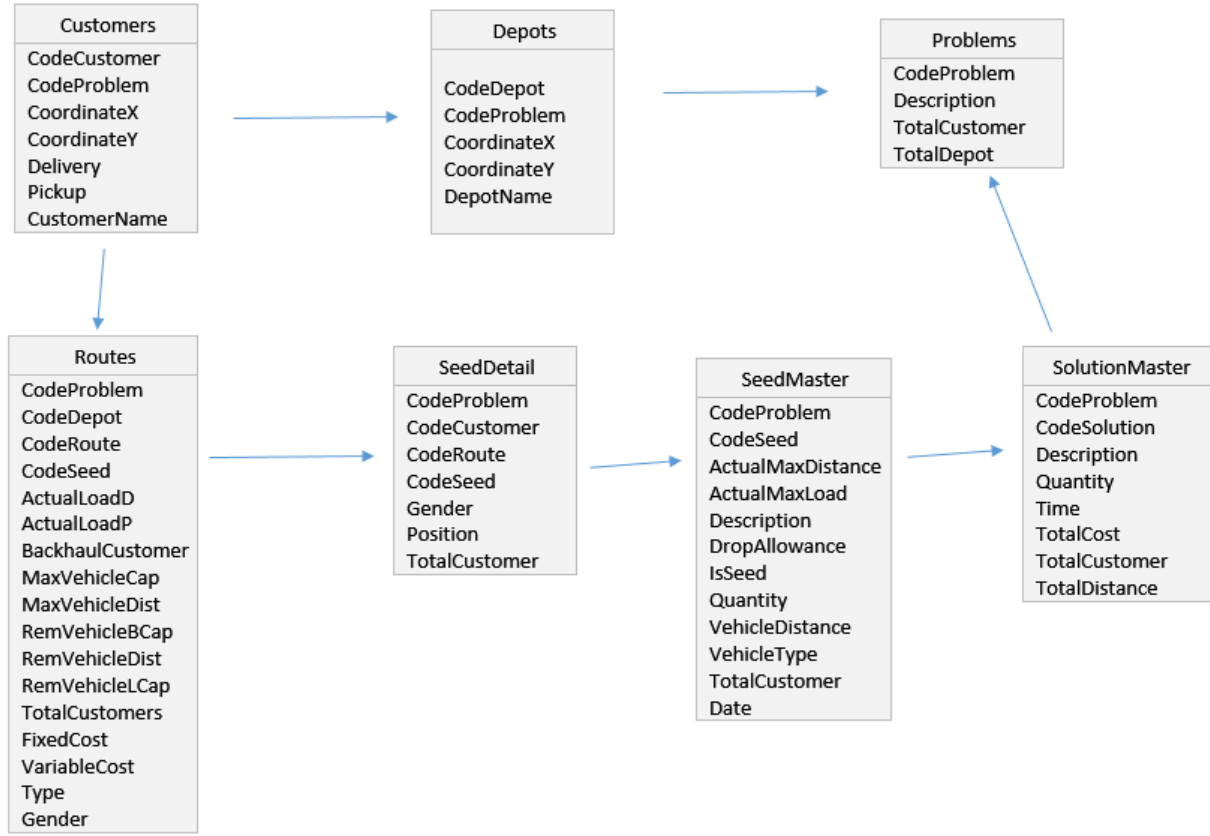
3.3.1 KDS Tasarımı

ADVISER2 üç ana modülden oluşmaktadır. Bunlar veri tabanı, model yönetim birimi ve kullanıcı ara yüzü modülleridir.

3.3.1.1 Veri Tabanı Modülü

ADVISER2' de kullanılan veri tabanı model yönetim sisteminin ve kullanıcı ara yüzünün ihtiyaç duyduğu verileri sunacaktır. Problem yapısıyla alakalı veriler veri tabanının girdilerini oluşturmaktadır. Bu girdiler metin dosyaları aracılığıyla veri tabanına girilmektedir.

ADVISER2' nin veri tabanı 7 ilişkili tablodan oluşmaktadır. Bunlar sırasıyla ARP ile ilgili temel parametreleri tutan Problemler tablosu, kullanıcıların tanımladığı başlangıç çekirdek tanımları tutan Çekirdek Detayları tablosu, çekirdek çözümleri tutan Çekirdek Çözümler tablosu, müşterilerin koordinat, tür ve taleplerini saklayan Müşteriler tablosu, problemler için tanımlanan müşteri sıralarını tutan Rotalar tablosu, depo bilgilerini tutan depolar tablosu ve üretilen çözümleri saklayan Çözümler tablosu olarak isimlendirilebilir.



Şekil 1. Karar destek sisteminin veri tabanışeması

3.3.1.2 Model Yönetim Modülü

Model yönetim modülü ARP' lerin görsel olarak tanımlandığı ve çözüldüğü süreçleri içermektedir. Kullanıcı ADVISER2' nin veri tabanından bir problem seçtikten sonra müşteriler ve depolardan oluşan harita ile karşılaşır ve istediği müşterileri istenen araçlara atayarak çekirdek çözüm oluşturabilir. Daha sonra, ADVISER2 bu çekirdek çözümleri göz önünde bulundurarak, 3.4. bölümünde verilen üst-sezgiseli kullanarak kullanıcıya tüm müşterileri kapsayan ve en iyi çözüme yakın bir çözüm sunmaktadır. Ayrıca kullanıcıya, müşterilerin otomatik olarak atanması seçeneği de sunulmaktadır. Kullanıcı müşterileri atarken herhangi bir kuralı ihlal ettiğinde uyarılır ve doğru bir atama yapması sağlanmaktadır. Özetle, ADVISER2' nin model yönetim modülü istenilen ARP varyasyonu için çözüm üretim kısmını kapsamaktadır.

3.3.1.3 Kullanıcı Ara Yüzü

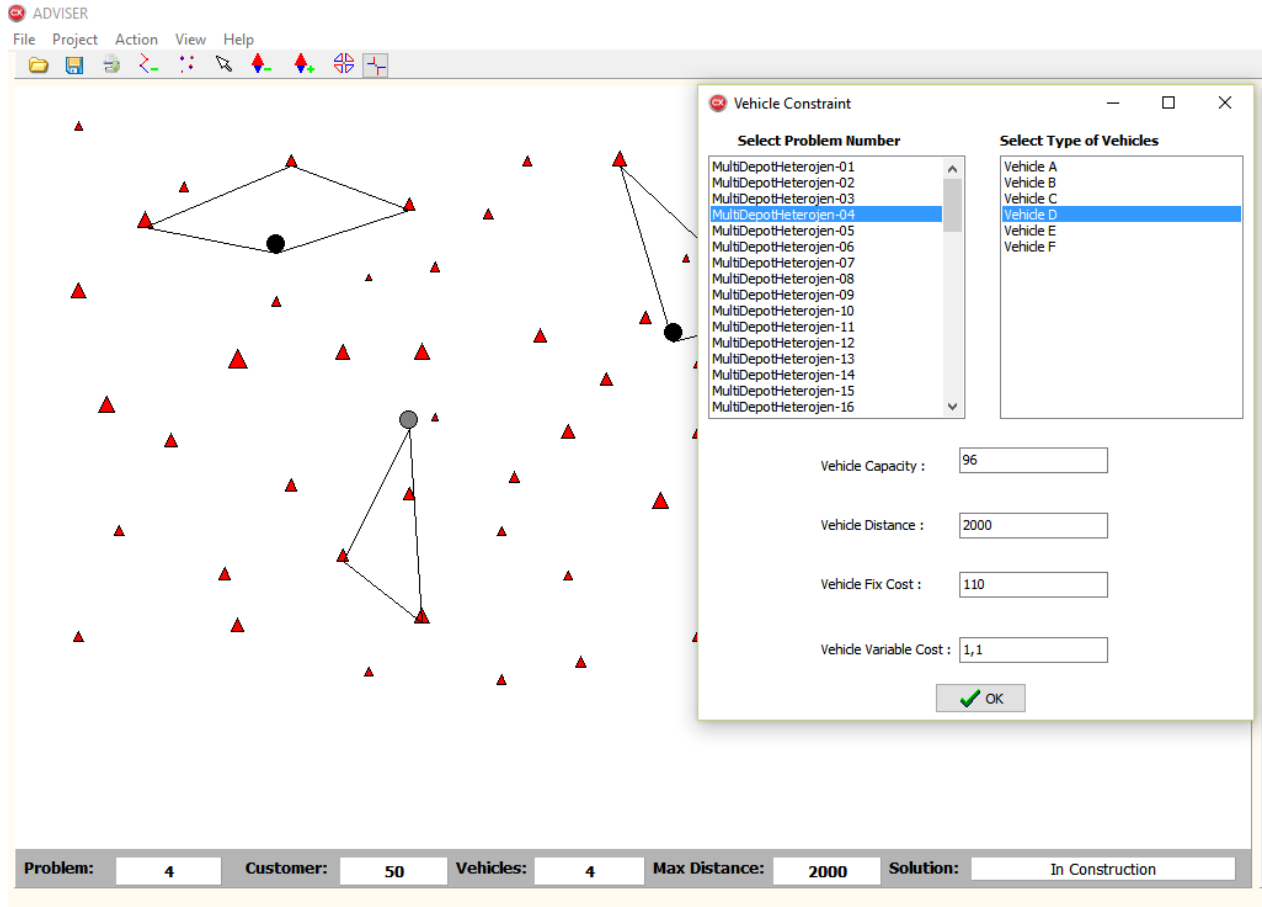
Sunulan KDS'nin temelinde kullanıcı etkileşimine ağırlık verilerek ARP problem türüne çözüm bulunması yer almaktadır. Bundan dolayı, proje konusu problemin çözümü için görsel etkileşimli bir çözüm yöntemi geliştirilmiş ve kullanıcı ara yüzünün kullanıcı ile karar vericiyi problemin çözüm sürecine direkt dahil etmesi sağlanmıştır. Uygulama ekranının en üstünde ana menü çubuğu ve araç çubuğu bulunmaktadır. KDS uygulamamız her bir müşteriyi talebiyle orantılı büyüklükte üçgen simgeyle göstermektedir.

Ara yüzün kullanıcıyla etkileşimini sağlamak amacıyla, kullanıcıya görsel uyarılar verilerek sistemle birlikte çalışması sağlanmıştır. Örneğin, müşterilerin artan taleplerine göre müşterileri temsil eden üçgen simgelerin büyümesi kullanıcıya yüksek talepli bölgelere araç ataması konusunda yardımcı olmaktadır.

Menü Çubuğu

ADVISER2' nin menü çubuğu dosya, proje, eylemler, görünüm ve yardım menülerinden oluşur. Dosya menüsü aç, kapat ve proje dosyaları ekle/çıkart, yazdır ve çıkış menülerinden oluşmaktadır. Kullanıcı aç komutunu seçtiğinde, veri tabanında kayıtlı problemlerin olduğu bir tablo açılmaktadır.

Kullanıcı problemi seçtikten sonra seçilen problem ana ekrana harita şeklinde yansıtılır. Bu ekranda kullanıcı çekerdek çözüm oluşturarak, problemleri analiz edebilir ve çözdürebilir. Araç eklemek için depoya tıklanıldığında araç kısıtları menüsü açılacaktır (Şekil 2).

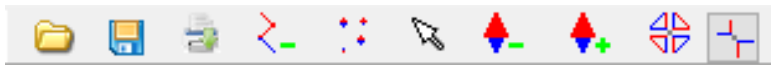


Şekil 2. Karar destek sisteminin ana sayfası ve araç kısıtları ekranı

Araç kısıtları ekranını kullanarak kullanıcı, problem numarasına göre araç tipi seçerek, araçların eşit veya farklı tipte olup olmadığına ya da kapasite veya uzaklık limitlerinin olup olmadığına karar verebilir. Ayrıca, kullanıcı her bir araç için maksimum kapasite, maksimum yol, değişken maliyet ve sabit maliyeti belirleyebilmektedir. Bu kısıtlar ARP' nin türünü tanımlamaktadır.

Araç çubuğu

ADVISER2'nin araç çubuğu iki bölümden oluşmaktadır. İlk üç buton sırasıyla aç, kaydet ve güncel çözümü kaydet komutlarının kısa yollarını içermektedir.



Şekil 3. Araç çubuğu

İkinci bölümü oluşturan butonlar ise kullanıcı etkileşimi sağlamakta önemli rol oynamaktadır. Kullanıcı her seferde tek bir butonu seçecek ve bu buton kullanıcının harita ve çözüm süreci ile etkileşimini sağlayacaktır. Her bir butonun işlevi aşağıda tanımlanmıştır.



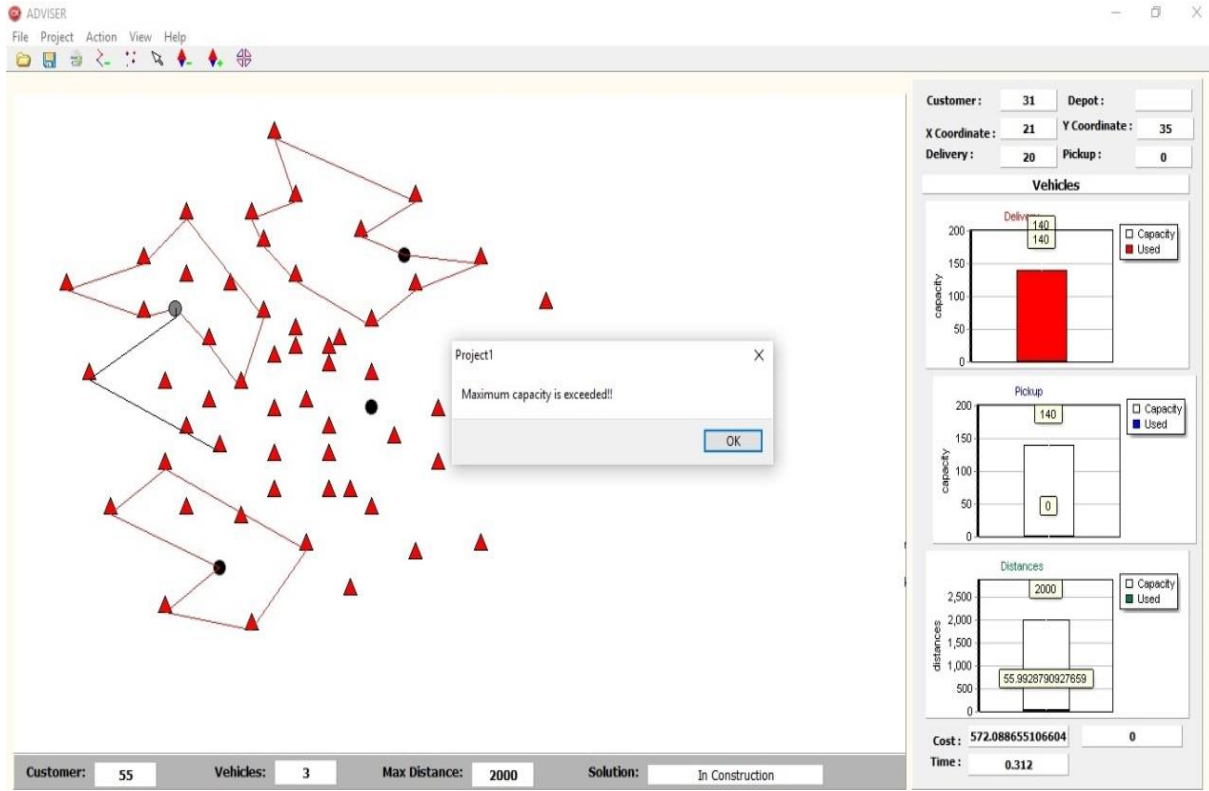
Varsayılan

Varsayılan butonu kullanıcının, önce depoya sonra müşterilere tıklamak şartıyla çekirdek çözümler oluşturması için kullanacağı butondur. Kullanıcı eğer kapasite aşımından dolayı seçemeyeceği bir müşteriye tıklarsa, açılan yeni bir pencere ile bu müşteriye seçemeyeceği konusunda uyarılır. Kullanıcı seçmek istediği müşterileri seçtikten sonra tekrar depoya tıklayarak rota seçme ve çekirdek çözüm oluşturma işlemini sonlandırabilir.



Müşteri Ekle

Müşteri ekle butonu, kullanıcının herhangi bir müşteriye istediği bir rotaya ekleyebilmesini sağlamak için tasarlanmıştır. Kullanıcı müşteri ekle butonunu seçtiğinde, öncelikle eklemek istediği müşteriye seçmesi gerekmektedir. Daha sonra eğer müşteriye diğer bir müşteri ve depo arasına eklemek isterse, eklemek istediği müşteriye ve sonra depoya tıklaması gerekir. Bu işlem sırasında kullanıcı herhangi bir hata yapar veya kapasite aşımı gerçekleşirse, kullanıcı tekrar açılan bir pencere yardımıyla uyarılmaktadır. Bu uyarının alındığı pencere Resim 5.1.2'te verilmiştir.

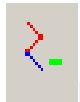


Şekil 4. Müşteri ekleme uyarı penceresi örneği



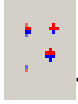
Müşteri Sil

Müşteri sil butonu kullanıcının herhangi bir rotadan müşteri silbilmesi için tasarlanmıştır. Müşteri sil butonu seçildikten sonra, kullanıcının bir rotadan silmek istediği müşterinin üzerine fareyle tıklaması yeterli olmaktadır.



Rota Sil

Rota sil butonu seçildikten sonra, kullanıcı silmek istediği rotada bulunan herhangi bir müşteriye tıklayarak tüm rotayı silbilmektedir.



Tüm Rotaları Sil

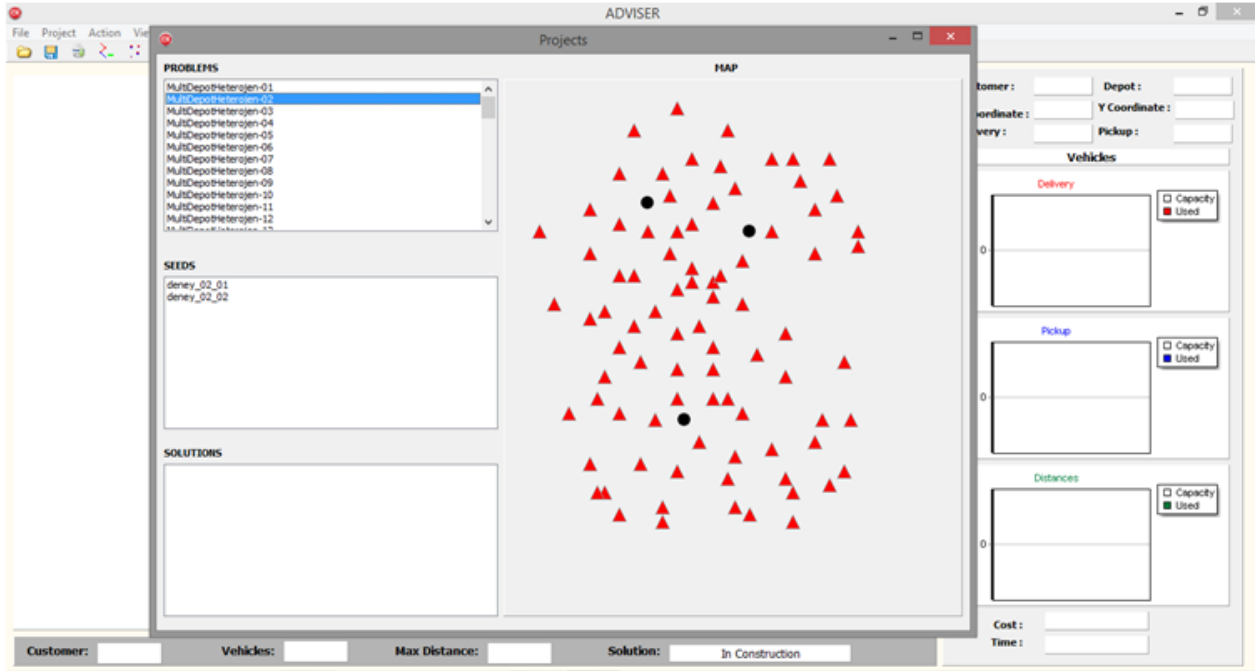
Kullanıcı tüm rotaları sil butonuna bastığında, haritadaki tüm rotalar silinmektedir. Tüm rotalar silindikten sonra tekrar varsayılan butonu aktifleştirecektir.



ARP-Çözücü

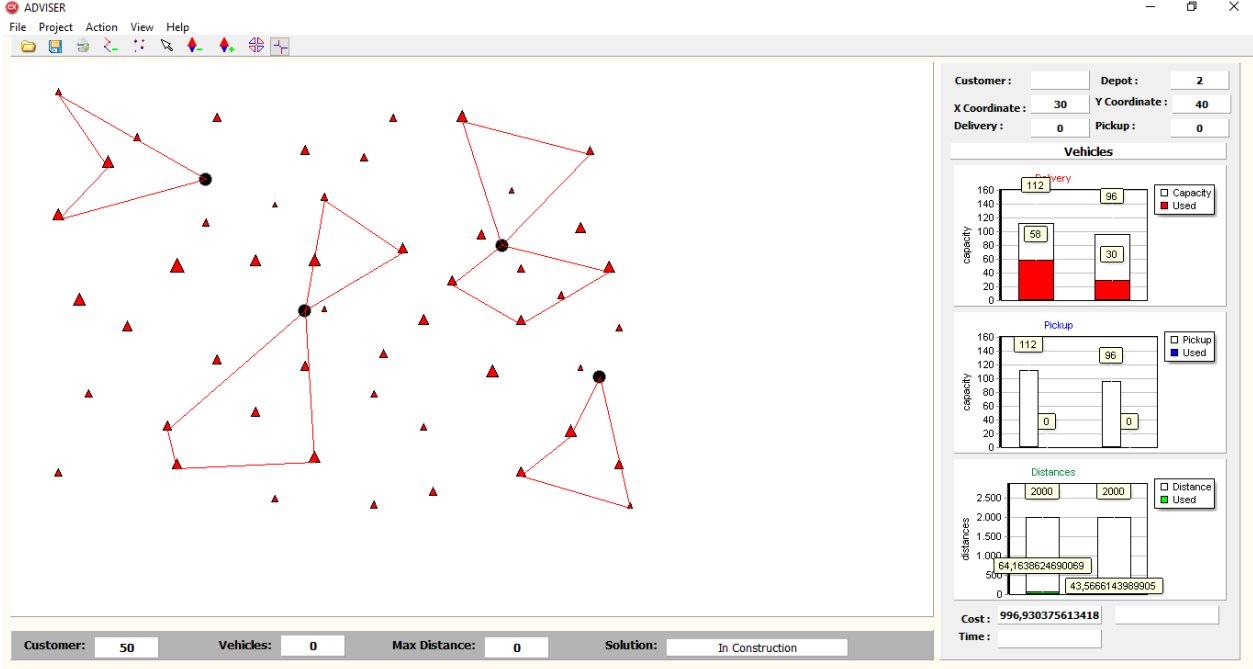
Kullanıcı ARP-Çözücü butonuna tıklarsa, seçilen probleme yukarıda tanımlanan algoritma ile çözüm uygulanır. Maksimum tekrar sayısına ulaşıldığında ARAHPA üst-sezgiseli durur ve elde edilen çözüm ADVISER2'nin çalışma alanında görüntülenir. Kullanıcı elde edilen çözümü kabul edebilir ya da bu bölümünde tanıtılan ADVISER2'nin araçlarını kullanarak çözümü iyileştirebilir.

Kullanıcı proje aç butonuna tıkladığında veritabanında kayıtlı problemleri görebilir (Şekil 5).



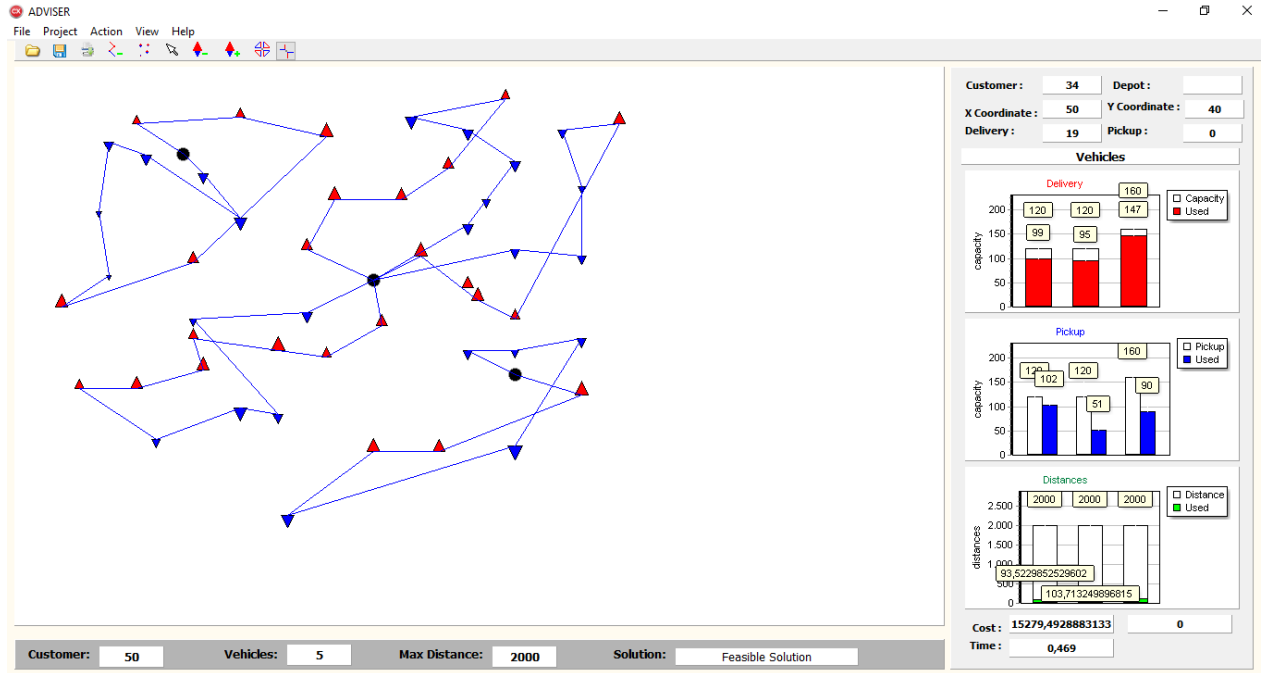
Şekil 5. Karar destek sisteminde yeni problem açma penceresi

Bu pencerede listelenen problemlerden birisine tıklanırsa problemin haritası gösterilir. Ayrıca kullanıcı seeds bölümünden daha önce kaydedilen çekirdek çözümleri seçebilir. Aşağıdaki resimde ise çekirdek çözümün oluşumuna dair bir örnek gösterilmiştir.



Şekil 6. Çekirdek çözüm oluşturma

Son olarak kullanıcı problemi çözdükten sonra Şekil 7'de verilen arayüzün sağında bulunan panelden, araçların kapasite bilgilerine erişebilmektedir. Böylece, bu bilgileri göz önünde bulundurarak yeni çekirdek çözümler üretip, daha iyi bir sonuca ulaşabilmesi hedeflenmiştir



Şekil 7. Etkileşim paneli örneği

3.4 Sezgisel Algoritmalar ve KDS Uygulamaları

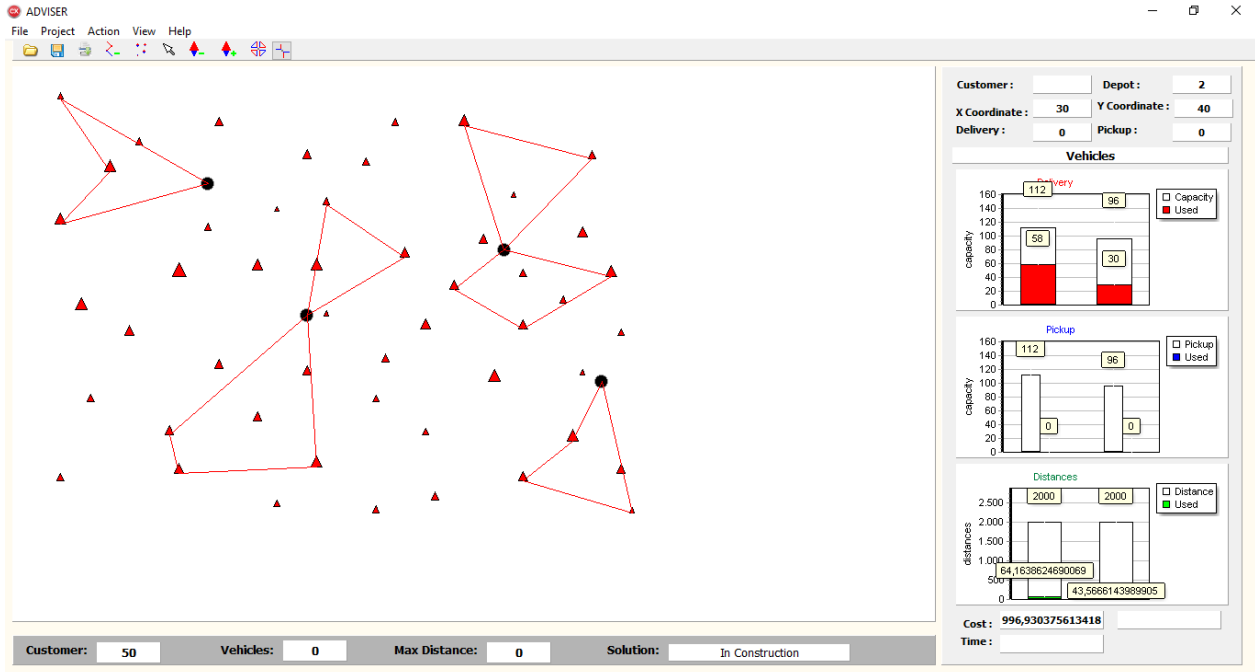
Bu bölümde, ilk önce HAÇARP için düzenlenen yeni Açgözlü Rastgele Adaptif Hafıza Programlama Arama (ARAHPA) sezgiseli verilmiştir. Daha sonra, Geri-toplamalı HAÇARP (GHAÇARP) için ARAHPA sezgiseli düzenlemeleri anlatılmıştır. Son olarak, HAÇARP ve GHAÇARP için ARAHPA sezgiseline otomatik çekirdek çözüm oluşturma sezgiseli eklenerek ve bazı kritik değişiklikler yapılarak geliştirdiğimiz ve ARAHPA2 ismini verdiğimiz sezgisel anlatılmıştır.

3.4.1 Aç-gözlü Rastgele Adaptif Hafıza Programlama Arama (ARAHPA) Sezgiseli KDS Uygulaması

KDS içinde uygulanan görsel etkileşimli çözüm yöntemi üç aşamadan oluşur: (i) Problem Tanımı ve Çekirdek Çözüm Seçimi, (ii) Çözüm Yapılandırma ve (iii) Çözüm İyileştirme. Bu adımlar aşağıdaki bölümlerde detaylı olarak açıklanmıştır.

3.4.1.1 Problem Tanımı ve Çekirdek Çözüm Seçimi Aşaması

Bu aşama, kullanıcının veri tabanından ilgili problemi seçtiği görsel etkileşimli yöntemin ilk adımıdır. Bu veri tabanı KDS 'nin müşterileri talepleriyle orantılı olacak büyüklükte harita üzerine çizmede kullanacağı müşteri koordinatlarını ve taleplerini içermektedir. Her bir aracın kapasite kısıdı, sabit ve değişken maliyetleri kullanıcı tarafından çekirdek müşterilerini seçmeden önce KDS içinde oluşturulan pencereler aracılığı ile çözüm için gerekli parametreler olarak algoritmaya iletilir. Daha sonra, kullanıcı tecrübesine ve hissiyatına dayanarak her bir araç için hizmet verilmesi gerektiğini düşündüğü müşterileri ara-yüz üstünde imleç ya da fare yardımıyla görsel etkileşimli olarak seçmelidir. Kullanıcı bu seçme işlemi yaparken başlangıç çekirdek çözümüne ekleyeceği her bir araç için rotayı eklemek istediği depoya fareyle harita üzerinde bir defa tıklayarak depoyu açmalı, ardından eklemek istediği aracın maksimum kapasitesini, sabit ve değişken maliyetini tanımlamalı, daha sonra sırasıyla eklemek istediği müşterilere tıklayarak müşterileri rotaya eklemelidir. Son olarak, kullanıcı seçilen depoya tekrar fareyle tıklayarak rotayı kapatmalıdır. Böylelikle, kullanıcı her bir araç için başlangıç çekirdek çözümlerini elle oluşturmuş olur. Şekil 8'de elle oluşturulan başlangıç çekirdek çözümü gösterilmektedir. Ayrıca program her bir araç için aracın başlangıçta verilen kapasitesini kontrol ederek kapasitenin aşılmasını engeller.



Şekil 8. Başlangıç çekirdek çözüm seçimi

3.4.1.2 Çözüm Yapılandırma Aşaması

Görsel etkileşimli çözüm oluşturma yöntemimiz seçilen çekirdek çözümleri içerecek rota oluşturabilen herhangi bir sezgisel ya da üst-sezgiselle çalışabilir. Önerilen ARAHPA yönteminin doğruluk, hız, kolaylık ve uyumluluk kıstaslarına uygun olması amaçlanmıştır. Bundan dolayı, iki aşamadan oluşan ve yeniden başlama yöntemine dayanan ARAHPA üst-sezgiseli önerilmiştir. İki aşamalı ARAHPA yönteminin detayları 3.4.2 bölümünde verilmiştir.

Çözüm yapılandırma aşamasında iki aşamalı ARAHPA algoritması başlangıç çekirdek çözümünü içeren bir çözüm önerir ve görsel etkileşimli çözüm yöntemi kullanıcıya elde edilen çözümü harita üzerinde gösterir. Aynı zamanda, elde edilen çözümle ilgili kalan araç kapasiteleri, toplam kat edilen mesafe ve toplam maliyet bilgileri ekranda kullanıcıya sunulmaktadır.

3.4.1.3 Çözüm İyileştirme Aşaması

Görsel etkileşimli çözüm yönteminin son aşaması olan çözüm iyileştirme aşamasında kullanıcı bir önceki aşamada algoritmanın elde ettiği çözümü görsel olarak inceleyebilir ve tecrübelerine, hissiyatına dayanarak çeşitli araçlarla interaktif olarak çözüm üzerinde iyileştirmeler yapabilir.

Bu aşamada belli bir algoritma yoktur. KDS içinde bulunan bazı araçlarla kullanıcı çözüm üzerinde kolayca değişiklikler yapabilir. Örneğin, çözüm önerme aşamasında ARAHPA algoritmasının önerdiği çözüm üzerindeki bazı üst üste gelen (overlap) rotalara atanan müşterileri rotadan çıkartarak ARAHPA algoritmasını tekrar çalıştırabilir. Böylelikle, algoritma çözümün son halini başlangıç çekirdek çözüm olarak boşta kalan müşterileri yeni rotalarına atayacak ve yeni yerel en iyi çözümlere ulaşabilecektir. Kullanıcı çözüm iyileştirme aşamasını, rotaya müşteri ekleme, rotadan müşteri silme, rotayı silme araçlarını kullanarak istediği kadar tekrar edebilir.

Algoritma 1'de, 3.4.2 bölümünde tanımlanan ÜS1 ve ÜS2 algoritmalarının KDS içinde birlikte çalıştığı bileşik algoritma verilmiştir. Burada, görsel etkileşimli KDS içinde ARAHPA dışta ÜS1 algoritmasıyla maksimum mümkün olmayan çözüm sayısına ulaşıncaya kadar her tekrarda mümkün ya da mümkün olmayan bir çözüm önerir ve iç kısımda ÜS1 algoritmasıyla bulunan en iyi çözüm ÜS2 algoritmasının başlangıç çözümü olacak şekilde ÜS2 algoritmasını çalıştırır.

Algoritma 1. Görsel etkileşimli KDS için ARAPHA algoritması

Başla:

Kullanıcı başlangıç düğümlerini seçer;

En iyi mümkün çözüm = ∞ ;

En iyi mümkün olmayan çözüm = ∞ ;

Tekrarla ÜS1(Maksimum mümkün ve mümkün olmayan çözüm sayılarına ulaşıncaya kadar)

Başlangıç düğümleriyle başla;

Açgözlü rastgele çözümü oluştur;

Komşuluk arama kullanarak çözümü iyileştir;

eniye_çözüm' ü tut;

ortalama_mümkün_maliyet' i hesapla;

Tekrarla ÜS2 (Maksimum düğüm değişikliğine ulaşıncaya kadar)

İlk döngüde bulunan en iyi mümkün çözümle başla;

Kümeleme fonksiyonunu en çoklayan düğümleri değiştir;

Tekrarla (Maksimum ortalama_mümkün_maliyet' ten kötü sonuç sayısına ulaşıncaya kadar)

Değiştirilmiş düğümlerle başla;

Açgözlü rastgele bir çözüm oluştur

Komşuluk arama kullanarak çözümü iyileştir;

eniye_çözüm_değişmiş_düğüm' ü tut;

ortalama_mümkün_değişmiş_maliyet' i hesapla;

Eğer (ortalama_mümkün_değişmiş_maliyet >
ortalama_mümkün_maliyet ise)

ortalama_mümkün_maliyet' ten kötü sonuç sayısını bir arttır;

Son.

3.4.2 Yeni Aç-gözlü Rastgele Adaptif Hafıza Programlama Arama (ARAHPA) Sezgiseli

Genel olarak, ARAHPA yönlendirilmiş yapılandırma üst-sezgiselleri (guided construction metaheuristics) sınıfında tanımlanabilir. ARAHPA, Aç-gözlü Rastgele Adaptif Arama Yöntemi (ARAAY) ve Adaptif Hafıza Programlama (AHP) yöntemlerinin birleştirilmiş halidir. ARAHPA ilk olarak Ahmedi ve Osman (2005) tarafından kapasiteli kümeleme problemi için önerilmiş, daha sonra Tütüncü vd. (2009) ve Tütüncü (2010) tarafından görsel etkileşimli KDS içerisinde sırasıyla geri-toplamalı ve heterojen araç-filolu araç rotalama problemlerini çözmek için kullanılmıştır. Tütüncü vd. (2009) ve Tütüncü (2010) çalışmalarında yeni bir iki aşamalı ARAHPA akıllı sezgiselini önermişlerdir.

Bu projede, tek depolu araç rotalama problemi için geliştirilen iki aşamalı ARAHPA algoritması Heterojen Araç-filolu Çok-depolu Araç Rotalama Problemi (HAÇARP) ve Geri-toplamalı Heterojen Araç-filolu Çok-depolu Araç Rotalama Problemi (GHAÇARP) için yeniden düzenlenmiş ve geliştirilmiştir.

ARAHPA yönteminin ilk aşamasında Reaktif ARAAY (RARAAY) kullanan yönlendirilmiş arama üst-sezgiseliyle rotalar oluşturulur, ikinci aşamasında ise ilk aşamada elde edilen çözümü geliştirmek amacıyla RARAAY ve AHP 'nin birleşimi olan başka bir yönlendirilmiş arama üst-sezgiseli önerilmiştir. İlk aşamada kullanılan üst-sezgisel ÜS1 olarak isimlendirilmiştir. ÜS1 çok aşamalı bir üst-sezgiseldir ve her tekrarı çözüm önerme ve yerel arama aşamalarını içerir. ÜS1 algoritmasının ilk adımı başlangıç çekirdek çözümleri seçme adımıdır. Kullanıcı 3.4.1 bölümünde açıklanan görsel etkileşim araçlarını kullanarak istediği başlangıç çözümünü oluşturabilmektedir. Bu adımda kullanıcı istediği depoya istediği araç türünden istediği kadar atayabilmektedir. Geliştirilen karar destek sistemi rotalara müşteri ekleme aşamasında ilgili problemin tüm kısıtlarını dikkate almakta ve kullanıcıya gerekli uyarıları vermektedir. Ayrıca, kullanıcı başlangıç çekirdek çözümü oluştururken araçların doluluk oranını, toplam kat ettiği mesafeyi haritanın sağ tarafında yer alan görsel etkileşim panelinden kontrol edebilmektedir.

Çözüm önerme aşamasında rotalar yeni bir çözüm önerme sezgiseli olan Göreceli Uzaklık Arama Algoritması (GUAA), tarafından oluşturulur. Bu sezgisel ARAHPA algoritmasının ÜS1 ve ÜS2 üst-sezgiselleri için genel-amaçlı çözüm önerme sezgiseli olarak kullanılmıştır.

Algoritma 2. US1 Algoritması

Başla:

Kullanıcı başlangıç çekirdek çözümü seçer;

En iyi mümkün maliyet $f_{c_{best}} = \infty$;

En iyi mümkün olmayan maliyet $i_{c_{best}} = \infty$;

Toplam mümkün maliyet $f_{c_{total}} = 0$;

$Tekrar_{sayısı} = 0$;

Tekrarla (Durdurma Koşulu Sağlanıncaya Kadar)

Başlangıç çekirdek çözümünüyle başla;

GUAA sezgiseli kullanarak açgözlü rastgele çözümü, x , oluştur;

RKDKA kullanarak x çözümünü x' çözümüne iyileştir;

Eğer (x' mümkün bir çözümse)

$$f_{c_{total}} = f_{c_{total}} + f_c(x');$$

$$Tekrar_{sayısı} = Tekrar_{sayısı} + 1;$$

Eğer ($f_c(x') < f_{c_{best}}$)

$$f_{c_{best}} = f_c(x');$$

$$x_f^* = x';$$

Başka Eğer ($f_c(x') < i_{c_{best}}$)

$$i_{c_{best}} = f_c(x');$$

$$x_i^* = x';$$

$$f_{c_{average}} = \frac{f_{c_{total}}}{Tekrar_{sayısı}};$$

En iyi mümkün çözüm ve en iyi mümkün maliyeti göster;

En iyi çözümü ve parametrelerini, ortalama mümkün maliyeti içeren Adaptif Hafızayı oluştur;

Son.

ÜS1 üst-sezgiselinin yerel arama adımında yerel en iyi çözümü bulmak için çözüm önerme kısmında önerilen mümkün ya da mümkün olmayan çözümlerden başlayabilen Reaktif Kısıtlı Değişken Komşuluk Arama (RKDKA) yöntemi kullanılmıştır. Yerel arama adımında iki tane iyi bilinen rota içi değiştirme (inter-route interchange) yöntemlerinden tek-düğüm değiştirme (one-

node interchange) ve iki-düğüm deęiřtirme yöntemleri sırayla kullanılmıř ve sonrasında 3-opt sezgiseli ile elde edilen rotalar son kez iyileřtirilmiřtir. GUAA sezgiseline benzer řekilde Kısıtlı DKA (KDKA) sezgiseli ÜS1 ve ÜS2 üst-sezgiselleri için genel-amaçlı sezgisel olarak önerilmiřtir.

Algoritma 3. US2 Algoritması

Bařla:

ÜS1' de bulunan en iyi çözüm bařlangıç çekirdek çözümü olarak seçilir;
 En iyi mümkün maliyet $f_{c_{best}} = f_{c_{best}}(x_f^*)$;
 En iyi mümkün olmayan maliyet $i_{c_{best}} = i_{c_{best}}(x_i^*)$;
 Ortalama mümkün maliyet $f_{c_{average}}$ = ÜS1' de bulunan ortalama mümkün maliyet;
 Tekrarla (Durdurma Kořulu Saęlanıncaya Kadar)
 {
 Bařlangıç çekirdek çözümü ÇÇG kullanarak deęiřtir;
 Tekrarla ($f_c(x') < f_{c_{average}}$)
 {
 Deęiřtirilmiř bařlangıç çekirdek çözümle bařla;
 GUAA sezgiseli kullanarak aęgözlü rastgele çözüm, x , oluřtur;
 KDKA kullanarak x çözümünü x' çözümüne iyileřtir;
 Eđer (x' mümkün bir çözümse)
 Eđer ($f_c(x') < f_{c_{best}}$)
 $f_{c_{best}} = f_c(x')$;
 $x_f^{**} = x'$;
 Bařka Eđer ($f_c(x') < i_{c_{best}}$)
 $i_{c_{best}} = f_c(x')$;
 $x_i^{**} = x'$;
 }
 Eđer ($x_f^{**} \neq \emptyset$ yada $x_i^{**} \neq \emptyset$)
 Hafızayı güncelle;
 Bařlangıç çekirdek çözüm=bulunan en iyi çözüm;
 }
 Bulunan en iyi çözümü göster;

Son.

ÜS1 algoritması aşağıda Algoritma 2' de verilmiştir. Burada en iyi mümkün çözüm x_f^* olarak ve en iyi mümkün olmayan çözüm x_i^* olarak tanımlanmıştır. Diğer araç rotalama problemlerine de uyumlu olan GUAA sezgiseli kullanılarak iyi kalitede başlangıç çözümü oluşturulur.

İkinci aşamada kullanılan üst-sezgisel ÜS2 üç aşamadan oluşmaktadır: öğrenme, yapılandırma ve yerel arama. Öğrenme aşamasında geçmiş çözümlerin bazılarını oluşturma aşamasında kullanmak üzere hafızada saklar. Öğrenme aşamasının amacı geçmiş bilgileri kullanarak yeni başlangıç çekirdek çözümleri oluşturmaktır. ÜS2 algoritması Algoritma 3' de verilmiştir. Burada, en iyi mümkün çözüm x_f^{**} olarak ve en iyi mümkün olmayan çözüm x_i^{**} olarak tanımlanmıştır.

ÜS1 algoritmasının sonunda mümkün çözümlerin ortalama maliyeti ve bulunan en iyi çözüm hafızaya kaydedilir. ÜS2 algoritmasının her tekrarının başlangıcında ÜS1' de kaydedilen çözümlerden elde edilen başlangıç çekirdek çözümleri kullanılır. Kaydedilen çözümlerden başlangıç çekirdek çözümü elde etmek için Çekirdek Çözüm Geliştirme (ÇÇG) sezgiseli kullanılır. Bu sezgisel hafızaya kaydedilen çözümün yan yana iki rotasından belli bir kurala göre çekirdekler seçer ve diğer rotaları değiştirmeden kullanır. Daha sonra bu çekirdek çözüm, önerilen yapılandırma sezgiseli GUAA tarafından kullanılır. Yapılandırma aşaması sonunda mümkün ya da mümkün olmayan bir çözüm elde edilir ve yerel arama adımı bu çözüm KDKA sezgiseli kullanılarak iyileştirilir. Yapılandırma aşaması ve yerel arama aşaması, yeni tanımlanan başlangıç çözümü için üst-sezgiselin ilk aşamasında hesaplanan ortalama mümkün maliyetten daha düşük bir mümkün maliyet bulununcaya kadar devam ettirilir.

3.4.2.1 ARAHPA Sezgiseli Çözüm Önerme Aşaması

Önceki bölümlerde belirtildiği üzere, ARAHPA sezgiselinde kullanılan ÜS1 ve ÜS2 üst-sezgisellerinin ilk aşaması çözüm önerme aşamasıdır. Bu aşamada her iki sezgisel yeni düzenlenen Göreceli Uzaklık Arama Algoritması (GUAA) sezgiseli ile yeni rotaları oluşturur. Bu bölümde GUAA sezgiselinin yapısı ve bu sezgisel içerisinde kullanılan Kısıtlı Aday Listesi (KAL, Restricted Candidate List) açıklanacaktır.

3.4.2.1.1 Yeni Çözüm Önerme Sezgiseli (GUAA)

GUAA sezgiselinin amacı başlangıç çekirdek çözümü olarak seçilmeyen müşterileri problem kısıtlarına bağlı kalarak ve toplam kat edilen mesafeyi en azlayacak şekilde rotalara atamaktır. Çok-depolu problemimizi çözmek için öncelikle bütün müşteriler depolara aşağıdaki yarıçap fonksiyonu kullanılarak kümelendirilir. Bu fonksiyonda, r_i , i deposunun yarıçapını ve d_{ij} , i müşterisi ve j müşterisi arasındaki uzaklığı göstermektedir.

$$r_i = \frac{1}{2} \min_j \{d_{ij} | i \neq j\}, \quad i, j = \{1, \dots, m\}$$

Burada, her bir depo için yarıçapı kendisine en yakın olan depoya arasındaki uzaklığın yarısı olan bir çember çizilmiş ve bu çemberin içinde kalan müşteriler ilgili depodan çıkan bir rotadaki ekleme maliyetinin en azlandığı pozisyona atanmıştır. Böylelikle, her bir depo için çizilen çemberlerin kesişmesi önlenmiş, fakat arada atanamayan müşteriler kalmıştır. Arada kalan müşteriler belli bir depoya bağlı kalmaksızın ekleme maliyetinin en az olduğu rotaya eklenmektedir. GUAA sezgiselinde her müşteri bir rotaya eklenir ve rotalar üç adımda oluşturulur.

GUAA sezgiselinin ilk aşamasında, herhangi bir rotaya atanmamış her bir müşteri için ekleme maliyeti ve ekleme pozisyonu Baker (1992) tarafından önerilen yaklaşıma göre hesaplanır. Bahsedilen yaklaşık hesaplama maliyeti şu şekilde hesaplanır: c_{rj} , j müşterisini r rotasına ekleme maliyetini göstermektedir.

Ekleme maliyeti,

$$c_{rj} = \alpha_r \min_{0 \leq k \leq K} \{d_{i_k j} + d_{j i_{k+1}} - d_{i_k i_{k+1}}\}$$

Burada i_0 ve i_{K+1} depoyu ve α_r , r rotasının değişken maliyetini temsil etmektedir. Bu hesaplama bir rotaya atanmamış her müşteri için ve atanabileceği her rota için hesaplanır. GUAA sezgiseli bu hesaplamaları yaparken, aynı zamanda problem kısıtlarını da sürekli kontrol eder ve j müşterisi için en az ekleme maliyetinin elde edildiği r rotasındaki pozisyonu, p_{rj} , da tutar.

GUAA sezgiselinin ikinci adımında, atanamayan müşteriler rotalara sırasıyla birer birer atanır. Müşterilerin rotalara eklenme sırası düşük ekleme maliyeti olan müşterilere ve atanabileceği rota sayısı en az olan müşterilere ekleme önceliği verilerek düzenlenir. Bu kurala göre, sadece bir rotaya eklenebilecek olan müşteriler ilk olarak eklenir, daha sonra birden fazla rotaya eklenebilecek müşteriler için en küçük ve ikinci en küçük ekleme maliyetleri hesaplanır. Her tekrarda, j müşterisinin eklenebileceği rota sayısı, R_j , tekrar hesaplanır. Eğer hiçbir rotaya atanamayacak olan, $R_j = 0$, en az bir j müşterisi varsa mümkün bir çözüm yoktur. Fakat sezgisel bütün mümkün rota atamalarını tamamlayıncaya kadar devam eder. GUAA sezgiseli sonucunda mümkün olmayan bir çözüm oluşması durumunda ARAHPA çalışmaya devam eder, çünkü ÜS1 ve ÜS2 algoritmaları içerisinde uygulanacak olan yerel arama adımlarının sonunda mümkün olmayan çözümlerden kurtulmak amacıyla hiçbir rotaya atanamayan müşteriler tekrardan mümkün olan rotalara atanması için kontrol edilir. Eğer sadece bir rotaya atanabilecek, $R_j = 1$, birden fazla müşteri varsa, sezgisel talebi en büyük olan müşteriyi seçer.

c_j^{1st} ve c_j^{2nd} sırasıyla j müşterisinin en küçük ve ikinci en küçük ekleme maliyetlerini ve IP_j müşterisinin ekleme önceliği katsayısını göstermektedir. IP_j katsayısı, j müşterisinin en küçük ve ikinci en küçük ekleme maliyetlerinin farkına eşittir. IP_j katsayısı en küçük ve ikinci en küçük ekleme maliyetleri arasındaki farkın en büyük olduğu müşterilere öncelik verilmesinde kullanılmaktadır.

Eğer $R_j = 0$ ya da $R_j = 1$ koşulunu sağlayan müşteri kalmadıysa, sezgisel eklenecek müşteriyi bir KAL' den (Restricted Candidate List) seçmektedir. KAL' den seçilecek olan bir sonraki müşteri bütün aday müşteriler için IP_j katsayısı hesaplanarak belirlenir. KAL yapısının detayları bir sonraki bölümde açıklanacaktır. GUAA çözüm oluşturma sezgiselinin algoritması Algoritma 4' te verilmiştir. Burada R^0 hiçbir rotaya atanamayan müşterilerin sayısını ve R^1 sadece bir rotaya atanabilen müşterilerin sayısını göstermektedir.

İkinci adımda, bir müşteri bir rotaya eklendiği zaman bütün ekleme maliyetleri $c_{r,j}$, bunlara karşılık gelen ekleme pozisyonları $p_{r,j}$ ve ekleme önceliği katsayıları IP_j her bir atanamayan j müşterisi için tekrar hesaplanmalıdır. Bu yöntem mümkün olan bütün müşteriler bir rotaya atanıncaya kadar tekrar edilir ve sonunda üçüncü adıma geçilir.

Algoritma 4. GUAA Çözüm Önerme Algoritması

Başla:

Her rotaya atanamayan j müşterisi ve her r rotası için c_{rj} ve p_{rj} hesapla;

Her rotaya atanamayan j müşterisi için c_j^{1st} ve c_j^{2nd} bul;

Her rotaya atanamayan j müşterisi için IP_j ve R_j hesapla;

$R^0 = 0, R^1 = 0$;

Tekrarla (Mümkün Müşteri Ataması Kalmayıncaya Kadar)

{

Eğer ($R^1 > 0$)

{

$j^* = j$ (talebi en yüksek olan ve $R_j = 1$);

}

Değilse

{

IP_j katsayılarını kullanarak KAL' ni oluştur;

$j^* = j$ müşterisi KAL' den rastgele seçilir;

$r^* = j^*$ müşterisi için ekleme maliyeti en küçük olan rota;

j^* müşterisini $p_{r^*j^*}$ pozisyonuna ekle;

j^* müşterisini atandı olarak işaretle;

r^* rotası için kalan kapasite ve uzaklığı güncelle;

$c_{r^*j^*}, p_{r^*j^*}, c_j^{1st}, c_j^{2nd}, IP_j$ ve R_j değerlerini her atanamayan j müşterisi

için

güncelle;

R^0 ve R^1 değerlerini güncelle;

}

}

Her rotaya 3-opt yöntemini uygula;

Son.

GUAA sezgiselinin üçüncü adımında 3-opt sezgiseli en iyi-iyileştirme stratejisi kullanılarak bütün rotalara uygulanır. Bu yöntemde, sezgisel bir rotadaki bütün mümkün üç ark değişimlerini hesaplar ve en yüksek kazancı (saving) sağlayan rotaya uygular.

3.4.2.1.2 Kısıtlı Aday Listesi (KAL)

Genel olarak KAL, içerisinde parça çözüme eklendiğinde sonuç çözümün maliyetini en az arttıracak olan en iyi elemanlar bulunur. En iyi çözüme yakın çözümün bir parçası olma ihtimali olmayan elemanlar KAL içerisine dâhil edilmemelidir. Bu düzenleme, iyi bir aday listesi değerlendirilmesi ve mantıklı bir şekilde daraltılması ile yapılabilir.

Bir önceki bölümde, GUAA sezgiselinin sadece bir rotaya atanabilen müşterilere atama önceliği verdiğinden bahsedilmişti. Birden fazla rotaya atanabilen diğer müşterilerin seçimi bütün müşterilerin açgözlü bir hesaplama fonksiyonuna göre değerlendirilmesiyle belirlenir. Bütün aday müşteriler için aşağıda verilen ve ekleme önceliği katsayılarına dayanan fonksiyon hesaplanmıştır. Müşterilerin ekleme önceliği katsayılarına göre değerlendirilmesi ile Aday Listesi (AL) ve en yüksek IP değerine sahip olan müşterilerle KAL oluşturulur. Temelde bu yaklaşım ve sadece bir rotaya atanabilecek olan müşterilere öncelik verilmesi önerdiğimiz ARAHPA algoritmasının açgözlü yönünü oluşturmaktadır. GUAA sezgiseli için, l parametresi KAL' nin uzunluğu olsun ve AL henüz bir rotaya atanmamış ve $R_j > 1$ özelliğini sağlayan müşterilerin listesi olsun. Eğer AL boş değilse ve $|AL| > l$ ise KAL, AL' den en büyük IP_j katsayısına sahip l adet müşteri içerir. Bir sonraki atanacak olan müşteri KAL' den rastgele seçilir. Bu rastgele seçim müşterilerin talepleriyle orantılı olan bir olasılıkla yapılır. w_j , j müşterisinin talebi olsun. j müşterisini bir sonraki bir rotaya eklenecek müşteri olarak seçme olasılığı şu şekilde verilmiştir,

$$P(j = \text{seçilen}) = \frac{w_j}{\sum_{vi \in KAL} w_i}$$

Eğer $|AL| \leq l$, KAL, AL içindeki bütün müşterileri içerir ve seçme işlemi hala olasılık fonksiyonuna göre yapılır. Eğer AL' de sadece bir müşteri varsa bu müşteri seçilir. Bu seçim yöntemi ARAHPA algoritmasının rastgele yönünü oluşturur ve bu seçme yönteminin parametresi KAL uzunluğudur ve l ile gösterilir.

ÜS1 algoritmasında KAL uzunluğu 5 ile başlar ve her 20 tekrarda bir artarak devam eder. ÜS1 algoritması süresince en iyi sonucun elde edildiği KAL uzunluğu adaptif hafızada tutulur ve ÜS2 algoritmasında ÜS1' de elde edilen en iyi KAL uzunluğu kullanılır.

3.4.2.2 ARAHPA Sezgiseli Yerel Arama Aşaması

ÜS1 ve ÜS2 algoritmalarının yerel arama adımları Kısıtlı Değişken Komşuluk Arama (KDKA) adı verilen aynı yerel arama algoritmasını kullanırlar. KDKA sezgiseli çözüm önerme aşamasında elde edilen mümkün ya da mümkün olmayan çözümlerden başlayarak yerel en iyi çözümü bulabilen bir iyileştirme sezgiselidir. KDKA yerel arama algoritması Algoritma 5' da verilmiştir. KDKA çözüm önerme aşamasında bulunan mümkün olmayan çözümlerden genellikle mümkün bir çözüm elde eder.

KDKA içerisinde çözüm önerme aşamasında bulunan çözümü iyileştirmek amacıyla iki adet çok iyi bilinen rotalar arası değiştirme yöntemleri, bir düğüm değiştirme ve iki düğüm değiştirme, bir adet rota içi iyileştirme yöntemi olan 3-opt kullanılır. Bu yöntemler sırayla uygulanır, ilk olarak bir düğüm değiştirme ve sonra iki düğüm değiştirme yöntemi uygulanır ve son olarak 3-opt sezgiseli uygulanır.

Değişiklikler ρ komşuluğunda kısıtlıdır. Her müşteri j , sadece kendisine en yakın ρ müşteriden birini içeren rotalara taşınabilir. Bu durumda, depolarda müşteri olarak düşünülür ve eğer j müşterisine en yakın ρ müşteriden biri depo ise j müşterisi söz konusu depodan çıkan bütün rotalara ve en yakın ρ müşteriden en az birini içeren rotalara taşınabilir. ÜS1 algoritmasının yerel arama adımında ρ değeri sistematik olarak KAL ile beraber değiştirilir ve en iyi çözümü bulan ρ hafızaya kaydedilir. ÜS1 algoritmasında en iyi çözümün elde edildiği ρ değeri, ÜS2 algoritmasında komşuluk boyutu olarak kullanılır. Bundan dolayı, ÜS1 içindeki yerel arama yöntemi Reaktif KDKA olarak isimlendirilirken ÜS2 içindeki yerel arama yöntemi sadece KDKA olarak isimlendirilir.

Yerel arama adımında uygulanan bir düğüm değiştirme ve iki düğüm değiştirme sezgisellerinde, birinci rotadan başlayarak aramaya başlanır ve ilk iyileştirme (first-improvement) stratejisi kullanılır. Söz konusu rota içindeki her müşteri atanabileceği rotalar içindeki en iyi pozisyona bir kazanç sağlıyorsa hemen taşınır ve bir sonraki rotanın ilk müşterisiyle aramaya devam edilir. Bu şekilde bütün rotalar için önce bir düğüm değiştirme sezgiseli daha sonra iki düğüm değiştirme

sezgiseli uygulanır. Son olarak, her bir rotaya en iyi iyileştirme (best-improvement) stratejisi kullanılarak 3-opt sezgiseli uygulanır.

Algoritma 5. KDKA Yerel Arama Algoritması

Başla:

Çözüm oluşturma aşamasında oluşturulan rastgele ağgözlü çözüm x_1 le başla,
Her müşteriye en yakın ρ müşteriyi bul,
 x_1 çözümü etrafında ρ komşuluğunda bir ve iki düğüm değiştirme yöntemlerini kullanarak yerel en iyi çözüm x_2' yi bul,

Eğer (x_2 mümkün bir çözümse)

{

x_2 çözümü etrafında ρ komşuluğunda bir ve iki düğüm değiştirme yöntemlerini kullanarak yerel en iyi çözüm x_3' yi bul,

Eğer ($x_3 < x_f^*$)

$$x_f^* = x_3;$$

}

Değilse (x_2 mümkün bir çözüm değilse)

{

x_2 çözümü içerisinde rotaya atanamayan her müşteriyi en az ekleme maliyetini veren rotadaki en iyi pozisyona ekleyerek x_3 oluştur,

Eğer (x_3 mümkün bir çözümse)

{

Eğer ($x_3 < x_f^*$)

$$x_f^* = x_3;$$

}

Değilse ($x_3 < x_i^*$)

$$x_i^* = x_3$$

}

Her rotaya 3-opt yöntemini uygula;

Son.

ARAHPA sezgiselinin çözüm önerme aşamasında müşteriler rotalara atanırken ve yerel arama adımında müşteriler rotalar arasında değiştirilirken yapılan değişiklikler her zaman araçların kapasite kısıdı ve diğer problem kısıtları kontrol edilerek yapılır. Bundan dolayı, ARAHPA sezgiseli çözüm önerme aşaması sonunda mümkün olmayan bir çözüm elde edilmişse bu çözümde herhangi bir rotaya atanamamış en az bir müşteri vardır demektir. Çözüm önerme aşamasında elde edilen çözüm mümkün olmayan bir çözüme, KDKA algoritması mümkün olmayan çözüme aynı yerel arama sezgisellerini uygular ve son olarak çözümün mümkün olmasına engel olan herhangi bir rotaya atanamayan müşteriler, problemin diğer kısıtları dikkate alınarak atanabileceği en iyi rotada en iyi pozisyona atanır. Bu yöntem her yerel arama adımında tekrar edilerek mümkün bir çözüm elde edilmesi amaçlanmıştır. Bu yöntemle mümkün olmayan çözüm elde etme ihtimalide bulunmaktadır fakat yapılan deneylerde bütün problem örnekleri için mümkün bir çözüm elde edilmiştir.

3.4.2.3 ARAHPA Sezgiseli Öğrenme Aşaması

Öğrenme aşaması geçmiş bilgilerden yararlanılarak oluşturma sürecini yönlendirmek anlamına gelmektedir. ARAHPA sezgiseli çözüm önerme ve çözüm iyileştirme aşamalarında bir tür hafıza kullanır. ARAHPA sezgiseli çözüm önerme aşamasında algoritma parametrelerinden biri olan atanacak olan müşteriyi seçmede kullanılan KAL uzunluğu kullanılmaktadır. ARAHPA sezgiselinin ÜS1 aşamasında KAL uzunluğu sistematik olarak değiştirilir ve en iyi çözümü veren değeri hafızada tutulur. Hafızada tutulan KAL uzunluğu ÜS2 algoritmasında kullanılır.

İkinci olarak, ÜS1 algoritmasının yerel arama adımında komşuluk boyutunun parametre olduğu reaktif KDKA yöntemi uygulanır. ÜS1 algoritmasında komşuluk boyutu ρ da sistematik olarak değişir ve yerel arama adımında en iyi çözümü veren ρ değeri hafızada tutulur. Hafıza tutulan ve ÜS1' de en iyi çözümün elde edildiği komşuluk boyutu olan ρ ÜS2 algoritmasında komşuluk boyutu olarak kullanılır.

Daha öncede belirtildiği üzere çekirdek çözüm seçme aşaması ARAHPA sezgiselinin başlangıç aşamasıdır ve çekirdek başlangıç çözümü iyi seçilmezse elde edilecek çözüm kalitesiz ya da mümkün olmayabilir. ARAHPA sezgiselinin adaptif hafıza yönü daha iyi başlangıç çekirdek çözümü seçmede etkilidir. ÜS1 algoritması sonunda, ÜS2 algoritması için en iyi KAL uzunluğu,

komşuluk boyutu, ortalama mümkün maliyet, en iyi maliyet ve bulunan en iyi çözüm hafızaya kaydedilir ve ÜS2 algoritmasında bu çözümden elde edilecek olan çekirdek çözümler başlangıç çekirdek çözümü olarak kullanılır.

ÜS1' de elde edilen en iyi çözümden yeni bir başlangıç çekirdek çözümü elde etmek için öncelikle en uzun maliyete sahip rota belirlenir ve onun yanındaki bir rota seçilir. Burada seçilen iki rotanın aynı depoya ait olması gerekmektedir. Seçilen bu rotalar değişikliğin yapılacağı rotalardır. Bu rotalardan aşağıdaki fonksiyonu en çoklayan bir müşteri çifti yeni çekirdekler olarak seçilir. Yani seçilen iki rotada sadece seçilen müşteri çiftini içeren birer müşteri bulunur ve rotalardaki diğer müşteriler rotalardan silinir ve boşta kalır.

$$g_{ij} = \frac{\theta_{ij}}{\max \theta_{ij}} + \frac{(d_{0i} + d_{0j})}{2 * \max d_{ij}}$$

Bu fonksiyon çözüm uzayının farklı bölgelerini aramak amacıyla yeni başlangıç çekirdek çözümleri tanımlamak için kullanılır. Burada θ_{ij} , i ve j müşterileri arasındaki açığı ve d_{ij} bu müşteriler arasındaki uzaklığı göstermektedir. d_{0i} ve d_{0j} bu müşterilerin bağlı oldukları depoya olan uzaklıklarını göstermektedir. Buradaki amaç rotalar arasındaki iç içe geçmeyi azaltmaktır. Bu amaç doğrultusunda, silinen iki rotanın oluşturduğu alandaki en geniş açıyla ayrılmış müşterilerin yeni çekirdek olarak seçilmesi amaçlanmıştır. Seçilen yeni çekirdeklerin depoya çok yakın olmalarını önlemek amacıyla da yukarıdaki g_{ij} fonksiyonuna ikinci kısım dahil edilmiştir.

Her yeni başlangıç çekirdek çözümü oluşturulduğunda yeni bir ÜS2 algoritması tekrarı başlar. ÜS2 algoritması ÜS1 de bulunan ortalama en iyi çözümden daha kötü olan mümkün çözüm sayısı 20 oluncaya kadar tekrar eder. Bu limite ulaşıldığında, ikinci en büyük maliyete sahip rota kullanılarak yeni çekirdek çözümler oluşturulur. Bu yöntem Çekirdek Çözüm Geliştirme (ÇÇG) olarak isimlendirilir ve bütün rotaların çekirdekleri en az bir kere değişinceye kadar devam edebilir.

3.4.3 GHAÇARP için yeni ARAHPA Sezgiseli

Bu bölümde, HAÇARP için uygulanan yeni ARAHPA sezgiseline GHAÇARP'ı çözebilmesi için yapılan düzenlemelerden bahsedilmiştir.

3.4.3.1.Yeni Aç-gözlü Rastgele Adaptif Hafıza Programlama Arama (ARAHPA) Sezgiseli GHAÇARP Düzenlemeleri

Önerilen yeni ARAHPA sezgiselinin GHAÇARP problemini çözebilmesi için sezgisel içerisine problem kısıtlarının kontrolleri eklenmiştir. Öncelikle, ARAHPA sezgiselinin başlangıç çekirdek çözümü oluşturma aşamasında müşteriler rotalara atanırken dağıtım müşterilerinin toplama müşterilerden önce gelmesi kısıdı eklendi. Algoritma sadece dağıtım müşterilerinden oluşan rotalara izin verirken sadece toplama müşterilerinden oluşan rotalara izin vermeyecek şekilde düzenlendi. Ayrıca, araç kapasiteleri toplama ve dağıtım müşterileri için ayrı ayrı kontrol edilmektedir.

Çözüm önerme aşamasında müşteriler rotalara atanırken dağıtım müşterilerinin toplama müşterilerden önce gelmesi kısıdı eklendi ve bu kısıda göre p_{rj} güncellenmiştir. Ayrıca sadece dağıtım müşterilerinden oluşan rotalara izin verilerken sadece toplama müşterilerinden oluşan rotalara izin verilmemiştir. Bu öncelik kısıtlarına GUAA sezgiselinin son aşaması olan 3-opt yönteminde de dikkat edilmiştir.

Yerel arama aşamasında kullanılan bir ve iki düğüm değiştirme sezgiselleri düzenlenirken yine öncelik kısıdına dikkat edilerek düzenlenmiştir. Yine KDKA sezgiseli içerisinde kullanılan 3-opt sezgiseli de öncelik kısıdına göre güncellenmiştir.

Öğrenme aşamasında bir değişiklik yapmaya gerek kalmamıştır. ÜS1 sezgiselinde bulunan en iyi çözüm üzerinden yeni başlangıç çekirdek çözümleri oluşturulur. Öncelik kısıtlarına uyularak ÜS1 sezgiselinde elde edilen en iyi KAL uzunluğu ve komşuluk boyutu kullanılarak ÜS2 algoritması tamamlanır.

3.4.4 Yeni Aç-gözlü Rastgele Adaptif Hafıza Programlama Arama 2 (ARAHPA2) Sezgiseli

Bu bölümde, 3.4.2 bölümünde detaylı olarak anlatılan ARAHPA sezgiselini geliştirerek ve Otomatik Başlangıç Çözümü Oluşturma (OBÇO) aracını ekleyerek geliştirdiğimiz ARAHPA2 sezgiseli anlatılacaktır.

3.4.4.1 Otomatik Başlangıç Çözümü Oluşturma (OBÇO) Sezgiseli

Proje kapsamında geliştirilen Karar Destek Sistemi (KDS) içerisinde ARAHPA sezgiselinin başlayabilmesi için kullanıcının ya elle başlangıç çözümünü oluşturması ya da veritabanına kaydedilmiş olan çekirdek çözüm ya da çözümlerden birini seçmesi gerekmektedir. Ancak, kullanıcının problem hakkında çok az ön bilgisi olduğunda ya da büyük boyutlu bir problemle karşılaşıldığında elle başlangıç çözümü oluşturmak hem daha çok zaman alacak hem kötü sonuç elde etme ihtimali artacaktır. Bu nedenlerden dolayı, kullanıcıya hem hızlı çalışan hem de problemin çözümü için gerekli olan en iyi araç kombinasyonuna yakın bir araç kombinasyonu sunabilen bir OBÇO sezgiseli geliştirilmiştir.

OBÇO sezgiseli ilk olarak müşterileri depolara kümeleyerek başlamaktadır. Bu kümeleme işlemini Oliveira vd. (2016) tarafından kullanılan yöntemle yapılmaktadır. Kümeleme yöntemi şu şekilde çalışmaktadır: Bir j müşterisi için iki kural tanımlanmaktadır (i) En yakın depo, (ii) j müşterisine en yakın olan müşteriye en yakın olan depo. Eğer bu iki kuralın sonucu aynı depo (A deposu) çıkıyorsa j müşterisi A deposuna atanır. Eğer j müşterisine en yakın depo A, ama ona en yakın olan müşterinin en yakın deposu B oluyorsa, j müşterisi hem A hem de B deposuna atanabilir.

OBÇO sezgiselinin ikinci aşamasında, depolara atanan müşterilere göre her deponun toplam talebi hesaplanır. Daha sonra, her depoda Süpürme Algoritması (SA) (Gillet ve Miller, 1974) kullanılarak müşteriler rotalara atanır. SA, her bir depo için depo merkezli bir referans noktası belirler ve her depoda müşteri, depo ve referans noktası arasındaki açıyı hesaplarken kullanılan referans noktasını rastgele belirlemektedir. Kullanıcı algoritmanın hangi depodan (En yüksek talepli, En küçük talepli, Rastgele) başlayarak müşterilerin rotalara atanacağını ve ilgili depoda müşteri atama sırasının referans noktasına göre saat yönüne doğru mu yoksa saatin yönünün tersine doğru mu olacağını KDS içerisinde OBÇO butonuna tıkladıktan sonra açılan ekrandan seçebilir. Bu ekranı Şekil 9.'da görebilirsiniz. SA, referans noktasından seçilen yönde ilerleyerek referans noktasından en küçük açıya sahip olan müşteriye çekirdek çözüm olarak seçer, daha sonra bir sonraki en küçük ikinci açıya sahip müşterinin talebini birinci çekirdek müşterinin talebine ekler. Bu şekilde açıyı arttırarak ve müşterilerin talepleri toplanarak devam edilir. Eğer araç kapasitesi aşıyorsa, kapasiteyi aşan müşteri ikinci çekirdek müşteri olarak seçilir. Bu adımlara depodan hizmet alacak olan bütün müşteriler ziyaret edilinceye kadar devam edilir.

Algoritma 6. Otomatik Başlangıç Çözümü Oluşturma (OBÇO) Algoritması

Başla:

Kümeleme yöntemini kullanarak müşterileri depolara göre sınıflandır;

Depolara atanan müşterilerin taleplerini toplayarak depo taleplerini bul;

Kullanıcı bir sonraki adım için depo sıralamasını seçer;

Kullanıcı süpürme algoritmasının yönünü seçer;

Tekrarla (Her depo için)

Rastgele bir referans noktası belirle;

Belirlenen referans noktasına göre merkezi depo olacak şekilde her müşterinin referans noktasıyla yaptığı açığı bul;

Rastgele bir araç türü seç;

Araç kapasitesi aşılmaya kadar süpürme algoritmasıyla belirlenen yönde müşterileri araca ata;

Eğer (Araç Kapasitesi Aşılırsa)

Rastgele bir araç türü ekleyerek müşterileri atamaya devam et;

Eğer (Toplam rota sayısı ≥ 2)

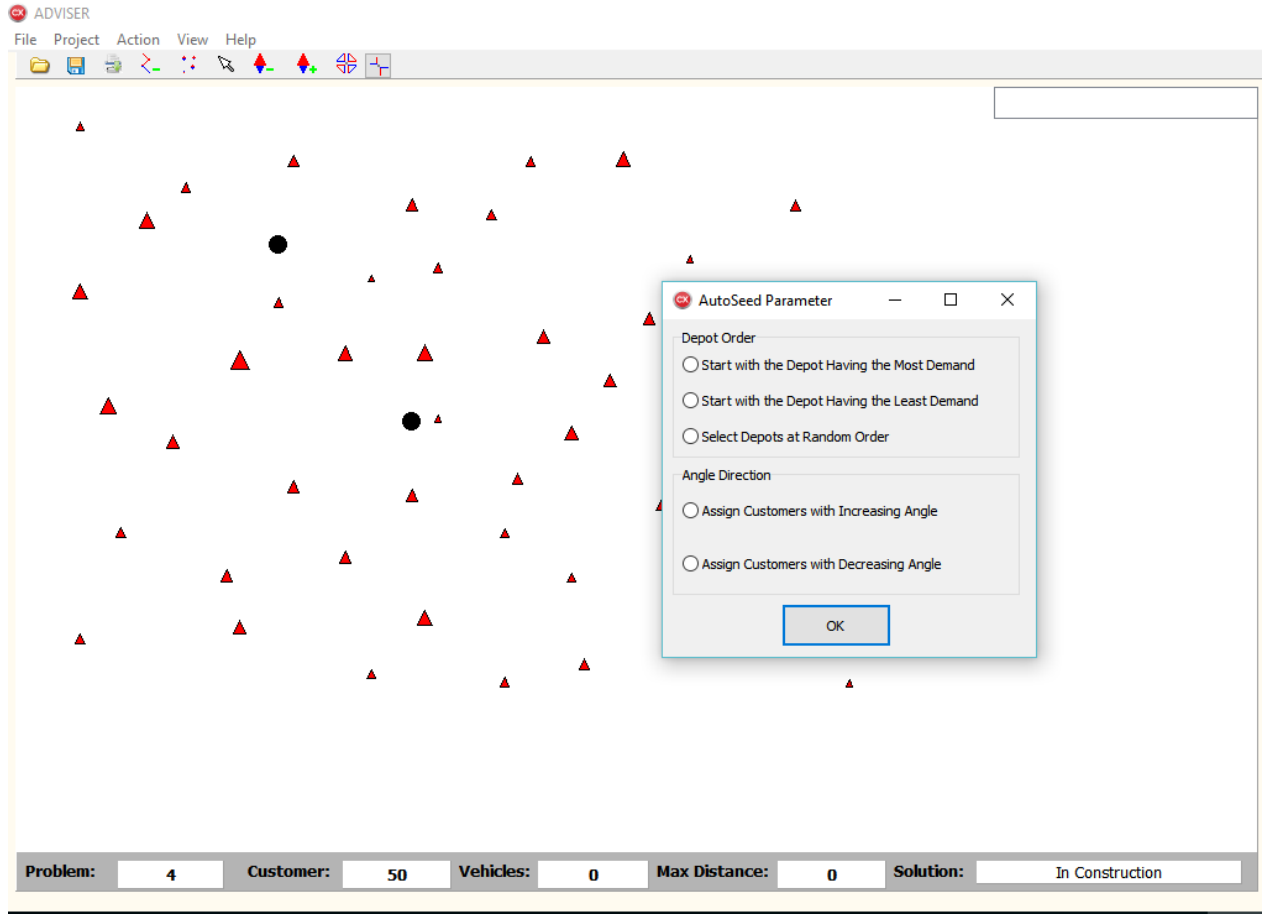
İlgili depodaki tüm olası rota ikililerini bul;

Tekrarla (Tüm olası ikililer için)

Maliyet kontrolü yaparak iki rotadan kapasitesi büyük olan araçla ya da daha büyük kapasiteli araçla rotaları birleştir;

Son

Kullanılan SA, her bir depoda müşterilerin atanacağı ilk araç tipini ve mevcut aracın kapasitesinin aşılması durumunda eklenecek olan yeni aracın tipini rastgele belirlemektedir. Ancak, geliştirdiğimiz diğer bir sezgisel olan Rota Birleştirme Algoritması (RBA) yardımıyla toplam rota sayısının 2 ve üzeri olması halinde her depoda oluşan rotaların bütün ikili kombinasyonlarına bakıp, her bir ikiliyi mümkün olan daha yüksek kapasiteli ya da iki rotadan kapasitesi daha yüksek olan rotadaki araçla maliyet ve kapasite kontrolü de yapılarak birleşme durumunu kontrol etmekte ve maliyeti düşüren bir değişiklikse rotalar birleştirilmektedir. Böylelikle, algoritmanın rastsal özelliğinin artması amaçlanmış ve en iyi araç kombinasyonuna yakın araç kombinasyonları kısa sürede elde edilmiştir.



Şekil 9. OBÇO sezgisel parametre seçme ekranı

GHAÇARP uygulaması için SA ve RBA sırasında müşterilerin öncelik kısıdı (toplama müşterisine bütün dağıtım müşterileri ziyaret edildikten sonra gidilebilir) dikkate alınmadan rotalar oluşturulmuştur. Son olarak, bütün müşteriler rotalara atandıktan sonra toplama müşterileri mevcut rotalarından çıkarılmış ve boşta kalmışlardır. ARAHPA içinde çalışan çözüm oluşturma algoritması öncelik kısıdına göre toplam müşterilerini rotalara atamaktadır. Ayrıca, kullanıcı OBÇO algoritmasının oluşturduğu çözüm üzerinde KDS'nin görsel etkileşim araçlarını kullanarak istediği gibi değişiklik yapabilmektedir.

3.4.4.2 Aç-gözlü Rastgele Adaptif Hafıza Programlama Arama (ARAHPA) Sezgisel Düzenlemeleri

ARAHPA algoritması için yapılan deneylerde, algoritmanın depolara atanan müşterileri çok değiştirmediyani depolar arasındaki müşteri değişiminde yetersiz kaldığı gözlemlenmiştir. Bundan dolayı, ARAHPA sezgiselinin ÜS1 aşamasının her tekrarında başlangıç çekirdek çözümünde her

depo için oluşturulan rotaların araç tipleri kendi aralarında rastgele değiştirilmiştir. Böylelikle, başlangıç çekirdek çözümünde seçilen müşterilere farklı araç tipleriyle hizmet verilerek toplam maliyetin azaltılması amaçlanmıştır. Eğer bir aracın mevcut yükü kendisiyle değiştirilecek olan aracın kapasitesinden fazlaysa, o rotada rastgele seçilen bir müşteri bırakılmakta ve diğer müşteriler boşa çıkarılmaktadır. Eğer kapasite kısıdı aşılmıyorsa rotalardaki müşteriler aynı kalmak koşuluyla sadece hizmet aldıkları araç tipi değiştirilmektedir.

GHAÇARP problemine uyum sağlaması için kapasite kısıdı aşılması halinde rotada rastgele bir dağıtım müşterisi kalması sağlanmıştır. Sadece toplama müşterilerinden oluşan bir rotaya izin verilmediği için her rotada en az bir dağıtım müşterisi bulunmaktadır. Böylelikle, ARAHPA sezgiselinde yapılan geliştirme HAÇARP ve GHAÇARP için uyumlu hale getirilmiştir.

4. BULGULAR

4.1 GHAÇARP Deney Sonuçları

Bu bölümde, GHAÇARP için önerilen temel model ve gevşetilmiş matematiksel modeller GAMS 23.9.4 ortamında modellenmiş ve IBM ILOG CPLEX 12.4.0.1 çözücüsü kullanılarak bulunmuştur. Deneyler Intel (R) Core (TM) i5-2310 CPU @ 2.90 GHz işlemcili 4 GB RAM' e ve Windows 7 Prof. 64 bit işletim sistemine sahip bilgisayarlar üzerinde yapılmıştır. Tablo 3'de GHAÇARP için elde edilen CPLEX sonuçları raporlanmıştır. Temel model ve gevşetilmiş model çözüldükçe 3 saat zaman kısıdı kullanılmıştır. 12 adet 20 müşterili problemin 10 tanesinin en iyi çözümleri her iki modelde de bulunmuştur. Kalan 2 tane 20 müşterili problemin en iyi çözümleri ise CPU kısıdı kaldırıldığında ya da CPLEX içinde çeşitli düğüm arama seçenekleri kullanıldığında (depth-first, breadth-first) elde edilebilir. Elde edilen en iyi çözümlere bakıldığında, gevşetilmiş model daha az CPU zamanında %89,34 aralıkla en iyi çözümleri elde etmiştir. Gevşetilmiş modelde, en iyi çözümleri bulunamayan diğer problemlerde ise aynı CPU zamanında genellikle daha iyi Alt Sınır (AS) değerlerini elde etmiş, sadece 2 problemde (TK12 ve TK31) temel modelden daha kötü bir AS bulunmuştur. 50 ve yukarısı müşteriye sahip toplam 24 problemin ÜS değerlerinin 12 tanesini temel model, 12 tanesini gevşetilmiş model bulmuştur.

Tablo 3. GHAÇARP için CPLEX Deney Sonuçları

Problem Bilgileri					CPLEX Çözümü (Temel Model)			CPLEX Çözümü (Gevşetilmiş Model)		
Problem	N	L	B	m	ÜS	En İyi Çözüm/AS	CPU (sn.)	ÜS	En İyi Çözüm/AS	CPU (sn.)
TK1	20	10	10	2	8584.87	8584.87	635.34	8584.87	8584.87	82.37
TK2	20	13	7	2	8412.76	8412.76	73.04	8412.76	8412.76	19.23
TK3	20	16	4	2	8604.8	8604.8	5091.7	8604.8	8604.8	941.15
TK4	20	10	10	2	9991.1	9991.1	15.44	9991.1	9991.1	4.09
TK5	20	13	7	2	11191.66	11191.66	618.37	11191.66	11191.66	60.75
TK6	20	16	4	2	11729.72	11421.02	10833.31	11729.72	11514.78	10801.26
TK7	20	10	10	2	8301.97	8301.97	224.73	8301.97	8301.97	27.28
TK8	20	13	7	2	8472.11	8472.11	158.09	8472.11	8472.11	41.57
TK9	20	16	4	2	8309.15	8309.15	642.36	8309.15	8309.15	75.97
TK10	20	10	10	2	9859.35	9859.35	15.19	9859.35	9859.35	3.82
TK11	20	13	7	2	11051.13	11051.13	42.31	11051.13	11051.13	40.26
TK12	20	16	4	2	11848.11	11609.68	10801.05	11848.11	11604.88	10807.87
TK13	50	25	25	3	15167	11152.1	10803.61	15506.95	11967.13	10823.51
TK14	50	33	17	3	14869.5	11515.62	10805.14	14457.46	12444.99	10820.55
TK15	50	40	10	3	15226.64	11797.81	10801.24	15964.77	12917.5	10833.91
TK16	50	25	25	3	14919.58	13362.35	10801.05	14658.83	13409.08	10801.13
TK17	50	33	17	3	15460.52	14153.42	10801.15	15427.15	14185.28	10801.68
TK18	50	40	10	3	16067.5	14911.92	10800.77	16329.97	14949.96	10800.98
TK19	50	25	25	4	11619.1	10835.09	10802.03	11691.6	10901.7	10801.04
TK20	50	33	17	4	12194.92	11174.03	10801.16	12600.08	11316.03	10801.33
TK21	50	40	10	4	15200.09	11359.27	10804.03	13561.53	11546.17	10804.39
TK22	50	25	25	4	13078.65	11455.71	10802.6	12440.13	11585.4	10800.94
TK23	50	33	17	4	12972.93	11772.94	10801.57	13121.07	11989.22	10801.07
TK24	50	40	10	4	13882.03	11993.12	10812.64	14573.47	12373.18	10803.25
TK25	75	37	38	5	43207.03	12364.72	10802.22	64054.62	13005.62	10802.88
TK26	75	50	25	5	67120.56	12346.75	10802.73	22245.36	12578.69	10802.05
TK27	75	60	15	5	34268.06	12442.08	10802.15	23567.46	12711.42	10802.04
TK28	75	37	38	5	34002.2	11764.6	10803.72	28893.98	12170.95	10802.88
TK29	75	50	25	5	40039.87	11906.7	10803.46	35236.64	12117.03	10803.43
TK30	75	60	15	5	36914.88	12026.39	10802.94	84174.95	12933.87	10802.75
TK31	100	50	50	4	22019.57	17687.43	10802.33	22331.23	17662.38	10802.1
TK32	100	66	34	4	24402.28	18719.58	10802.38	22168.34	18772.6	10802.22
TK33	100	80	20	4	23781.69	19888.16	10802.8	26517.18	19935.94	10802.63
TK34	100	50	50	3	21666.42	16075.16	10801.75	52873.14	16152.47	10801.72
TK35	100	66	34	3	54898.89	16437.38	10801.93	45792.06	16585.22	10801.74
TK36	100	80	20	3	70885.71	17097.26	10801.83	30331.06	17252.16	10801.69

ÜS: Üst sınır AS: Alt sınır

GHAÇARP için geliştirilen ve bölüm 3.4.2 de verilen ARAHPA sezgiselinin 36 örnek problem üzerinde denenmesi ile elde edilen sonuçlar Tablo 4 de verilmiştir. Bu sonuçlara göre GAMS in üst sınırlarında verilen çözümlerden ortalama %12,33 daha iyi sonuçlar elde edilmiştir. Ayrıca bu sonuçlara ilaveten KDS'nin araçları kullanılarak uygulanan kullanıcı etkileşimi ile bu sonuçlar %4,85 daha da iyileştirilebilmiştir. Bu sonuçlar Tabla 4 de GRAMPS1 kolonunda verilmiştir. Ayrıca önerilen sezgiselle, GAMS ile bulunan on en iyi çözüm ortalama 0,496 CPU(sn) elde edilerek önerilen sezgiselin sonuca ulaşmada ki üstün performansı tespit edilmiştir.

Tablo 4. GHAÇARP için yeni ARAHPA Deney Sonuçları

Problem	LB	Gams UB	GRAMPS1	Etkileşim Öncesi	GRAMPS1 CPU (sn)	Gap (%) vs. LB	Gap (%) vs. Gams	Gap (%) vs. Etkileşim	Araç Kombinasyonları
TK1	8584,87	8584,87	8584,87	8628,79	1,70	0,00	0,00	-0,51	EBB, DA
TK2	8412,76	8412,76	8412,76	9813,95	0,17	0,00	0,00	-14,28	AED, DA
TK3	8604,80	8604,80	8604,80	8997,47	0,32	0,00	0,00	-4,36	EE, DA
TK4	9991,10	9991,10	9991,10	10049,90	0,31	0,00	0,00	-0,59	AA, AA
TK5	11191,66	11191,66	11191,66	11259,76	0,30	0,00	0,00	-0,60	AAA, AA
TK6	11514,78	11729,72	11729,72	12056,34	0,30	1,87	0,00	-2,71	AAB, AA
TK7	8301,97	8301,97	8301,97	8531,22	0,31	0,00	0,00	-2,69	EB, DA
TK8	8472,11	8472,11	8472,11	8678,10	0,22	0,00	0,00	-2,37	EA, EAB
TK9	8309,15	8309,15	8309,15	8754,12	0,33	0,00	0,00	-5,08	E, ADE
TK10	9859,35	9859,35	9859,35	10124,21	0,30	0,00	0,00	-2,62	AA, AA
TK11	11051,13	11051,13	11051,13	11121,00	0,30	0,00	0,00	-0,63	AA, AAA
TK12	11609,68	11848,11	11848,11	12134,15	0,28	2,05	0,00	-2,36	AA, AAB
TK13	11967,13	15167,00	13482,50	14454,83	0,36	12,66	-11,11	-6,73	AAAF, AAF, ACF
TK14	12444,99	14457,46	14124,70	14265,60	2,34	13,50	-2,30	-0,99	AAE, AAAAB, AAEF
TK15	12917,50	15226,64	14626,30	14832,74	0,52	13,23	-3,94	-1,39	AAA, FF, EE
TK16	13409,08	14658,83	14276,74	14631,25	0,36	6,47	-2,61	-2,42	A, AA, AA
TK17	14185,28	15427,15	15325,10	15741,12	0,38	8,04	-0,66	-2,64	AA, AAA, A
TK18	14949,96	16067,50	15919,00	16316,22	0,38	6,48	-0,92	-2,43	AA, AAA, AA
TK19	10901,70	11619,10	11598,94	11764,43	0,53	6,40	-0,17	-1,41	AB, AAA, AAA, A
TK20	11316,03	12194,92	12142,53	12814,07	0,41	7,30	-0,43	-5,24	AAA, AAAAB, AAA, B
TK21	11546,17	13561,53	13372,56	13372,56	0,55	15,82	-1,39	0,00	A2, A4B, A2B, A3
TK22	11585,40	12440,13	12368,26	12896,79	0,31	6,76	-0,58	-4,10	A2B, A2B, AB, A2
TK23	11989,22	12972,93	12392,69	13345,19	0,52	3,37	-4,47	-7,14	A2B, A2B2, A3, AB
TK24	12373,18	13882,03	13955,62	14314,93	0,50	12,79	0,53	-2,51	AB3, AB2, B2, AB
TK25	13005,62	43207,03	14416,82	15708,25	0,88	10,85	-66,63	-8,22	B3, B2, AB, AB, AB
TK26	12578,69	22245,36	15333,31	15817,72	0,83	21,90	-31,07	-3,06	AB, B2, BC, AB, AB
TK27	12711,42	23567,46	15445,98	16437,96	0,80	21,51	-34,46	-6,03	A2B, B2, AB2, AB, B2
TK28	12170,95	28893,98	15099,45	19424,45	0,92	24,06	-47,74	-22,27	A2D, BC2, BD, BC, C2
TK29	12117,03	35236,64	15769,88	20955,93	3,89	30,15	-55,25	-24,75	BD, BC2, A2E, B2, CD
TK30	12933,87	36914,88	15409,49	21569,75	1,17	19,14	-58,26	-28,56	C2, BCD, BC2, A2E, B4
TK31	17687,43	22019,57	19655,81	19701,04	3,27	11,13	-10,73	-0,23	A4, A2, A2, A3
TK32	18772,60	22168,34	21098,84	21163,64	2,81	12,39	-4,82	-0,31	A4, A3, A2, A3
TK33	19935,94	23781,69	21751,00	22271,12	0,52	9,10	-8,54	-2,34	A4, A3, A3, A3
TK34	16152,47	21666,42	19037,48	19526,29	0,45	17,86	-12,13	-2,50	A5B, A2B, A4B
TK35	16585,22	45792,06	21245,60	21349,11	0,38	28,10	-53,60	-0,48	A3BC, A2B2C, A3BC
TK36	17252,16	30331,06	20465,21	20465,21	1,84	18,62	-32,53	0,00	A2BC, A4B, A6B
Ortalama Değerler:		17773,79	13740,85	14535,81	0,83	9,49	-12,33	-4,85	

GHAÇARP için geliştirilen ve bölüm 3.4.4 de verilen yeni ARAHPA2 sezgiselinin 36 örnek problem üzerinde denenmesi ile elde edilen sonuçlar Tablo 5 de verilmiştir. Bu sonuçlara göre GAMS in üst sınırlarında verilen çözümlerden ortalama %11,87 daha iyi sonuçlar elde edilmiştir. Ayrıca bu sonuçlara ilaveten KDS'nin araçları kullanılarak uygulanan kullanıcı etkileşimi ile bu sonuçlar %13,52 daha da iyileştirilebilmiştir. Bu sonuçlar Tabla 5 de GRAMPS2 kolonunda verilmiştir. Ayrıca önerilen sezgiselle, GAMS ile bulunan on en iyi çözüm ortalama 0,142 CPU(sn) elde edilerek önerilen sezgiselin sonuca ulaşmada ARAPHA dan daha hızlı olduğu ve oldukça yüksek performans gösterdiği tespit edilmiştir.

Tablo 5. GHAÇARP için yeni ARAHPA2 Deney Sonuçları

Problem	LB	Gams UB	GRAMPS1	GRAMPS2	Etkileşim Öncesi	CPU (sn)	Gap (%) vs. LB	Gap (%) vs. GRAMPS1	Gap (%) vs. Gams	Gap (%) vs. Etkileşim	Araç Kombinasyonları
TK1	8584,87	8584,87	8584,87	8584,87	9040,90	0,16	0,00	0,00	0,00	-5,04	EBB, DA
TK2	8412,76	8412,76	8412,76	8412,76	9427,76	0,16	0,00	0,00	0,00	-10,77	AED, DA
TK3	8604,80	8604,80	8604,80	8604,80	8758,58	0,14	0,00	0,00	0,00	-1,76	EE, DA
TK4	9991,10	9991,10	9991,10	9991,10	12261,41	0,17	0,00	0,00	0,00	-18,52	AA, AA
TK5	11191,66	11191,66	11191,66	11191,66	13588,99	0,16	0,00	0,00	0,00	-17,64	AAA, AA
TK6	11514,78	11729,72	11729,72	11729,72	13101,20	0,19	1,87	0,00	0,00	-10,47	AAB, AA
TK7	8301,97	8301,97	8301,97	8301,97	8448,35	0,14	0,00	0,00	0,00	-1,73	EB, DA
TK8	8472,11	8472,11	8472,11	8472,11	8636,33	0,14	0,00	0,00	0,00	-1,90	EA, EAB
TK9	8309,15	8309,15	8309,15	8309,15	8926,46	0,16	0,00	0,00	0,00	-6,92	E, ADE
TK10	9859,35	9859,35	9859,35	9859,35	12289,29	0,16	0,00	0,00	0,00	-19,77	AA, AA
TK11	11051,13	11051,13	11051,13	11051,13	13788,38	0,16	0,00	0,00	0,00	-19,85	AA, AAA
TK12	11609,68	11848,11	11848,11	11848,11	14213,06	0,16	2,05	0,00	0,00	-16,64	AA, AAB
TK13	11967,13	15167,00	13482,50	15089,24	15309,69	0,30	26,09	11,92	-0,51	-1,44	AAAF, AAF, ACF
TK14	12444,99	14457,46	14124,70	15194,86	15922,17	0,28	22,10	7,58	5,10	-4,57	AAE, AB, BEFF
TK15	12917,50	15226,64	14626,30	15388,37	15896,30	0,27	19,13	5,21	1,06	-3,20	AEE, FF, EE
TK16	13409,08	14658,83	14276,74	14212,34	19111,49	0,28	5,99	-0,45	-3,05	-25,63	A, AA, AA
TK17	14185,28	15427,15	15325,10	15312,05	17554,52	0,27	7,94	-0,09	-0,75	-12,77	AA, AAA, A
TK18	14949,96	16067,50	15919,00	15918,99	18718,74	0,25	6,48	0,00	-0,92	-14,96	AA, AAA, AA
TK19	10901,70	11619,10	11598,94	11506,90	14186,74	0,27	5,55	-0,79	-0,97	-18,89	AB, AAA, AA, AA
TK20	11316,03	12194,92	12142,53	12443,46	13685,35	0,30	9,96	2,48	2,04	-9,07	AAA, AAAB, AAA, B
TK21	11546,17	13561,53	13372,56	13321,98	13857,98	0,28	15,38	-0,38	-1,77	-3,87	A2, A4B, A2B, A3
TK22	11585,40	12440,13	12368,26	12312,14	15433,00	0,28	6,27	-0,45	-1,03	-20,22	A2B, A2B, AB, A2
TK23	11989,22	12972,93	12392,69	12973,34	16262,90	0,31	8,21	4,69	0,00	-20,23	A2B, A2B2, A3, AB
TK24	12373,18	13882,03	13955,62	13965,18	16513,25	0,28	12,87	0,07	0,60	-15,43	AB3, AB2, B2, AB
TK25	13005,62	43207,03	14416,82	14607,17	18687,65	0,58	12,31	1,32	-66,19	-21,84	B3, B2, AB, AB, AB
TK26	12578,69	22245,36	15333,31	15063,17	17488,57	0,56	19,75	-1,76	-32,29	-13,87	AB, B2, BC, AB, AB
TK27	12711,42	23567,46	15445,98	15368,48	18705,28	0,58	20,90	-0,50	-34,79	-17,84	A2B, B2, AB2, AB, B2
TK28	12170,95	28893,98	15099,45	15975,15	18557,66	0,58	31,26	5,80	-44,71	-13,92	A2D, BC2, BD, BC, C2
TK29	12117,03	35236,64	15769,88	16993,06	20474,87	0,59	40,24	7,76	-51,77	-17,01	BD, BC2, A2E, B2, CD
TK30	12933,87	36914,88	15409,49	16010,30	18141,75	0,61	23,79	3,90	-56,63	-11,75	C2, BCD, BC2, A2E, B4
TK31	17687,43	22019,57	19655,81	18751,38	24970,59	0,95	6,02	-4,60	-14,84	-24,91	A2, A2, A2, A3
TK32	18772,60	22168,34	21098,84	20265,07	26268,49	1,09	7,95	-3,95	-8,59	-22,85	A3, A3, A2, A3
TK33	19935,94	23781,69	21751,00	21668,24	26097,47	1,09	8,69	-0,38	-8,89	-16,97	A4, A3, A3, A3
TK34	16152,47	21666,42	19037,48	18314,11	23702,75	1,00	13,38	-3,80	-15,47	-22,73	A4B, A2B, A4
TK35	16585,22	45792,06	21245,60	18982,56	20807,81	1,05	14,45	-10,65	-58,55	-8,77	A3B, A2B2, A3B
TK36	17252,16	30331,06	20465,21	19929,48	22918,92	1,19	15,52	-2,62	-34,29	-13,04	A2BC, A4B, A5B
Ortalama Değerler:		17773,79		13775,66	16159,85	0,42	10,12	0,56	-11,87	-13,52	

4.2 HAÇARP için yeni ARAHPA Deney Sonuçları

Tablo 6' da HAÇARP örnek problemleri için verilen sonuçlara bakıldığında önerilen yeni ARAHPA sezgiseli ile bir problem için (#5) bilinen en iyi çözümden %0,12 daha iyi sonuç bulunmuştur ve literatürde yeni en iyi çözüm olarak geçecektir. Ancak ortalamada bilinen en iyi çözümlerden ortalama %0,97 uzaklaşmıştır ve bu sonuçlara ortalama %3,75 sn. gibi çok kısa bir CPU zamanında ulaşmıştır. Şu aşamada elde edilen sonuçlar önerilen sezgiselin performansının yüksek olduğunu göstermektedir ve yeni ARAHPA sezgiselinin KDS içinde kullanılarak kullanıcıyla interaktif şekilde çalışması halinde çok kısa sürelerde daha iyi sonuçlar alınması planlanmaktadır.

Tablo 6. HAÇARP için yeni ARAHPA Deney Sonuçları

Problem	N	m	TD	Q1	Toplam Maliyet		GAP (%)	CPU (sn)		Araç Kombinasyonları
					Salhi vd.	GRAMPS1		Salhi vd.	GRAMPS	
1	55	4	∞	60	1419,70	1441,90	1,56	102	0,27	B2CE6
2	85	3	∞	60	2131,90	2154,50	1,06	84	0,28	CD4E8
3	85	3	∞	96	1489,60	1508,30	1,26	42	0,32	CE7
4	50	4	∞	48	1499,30	1516,10	1,12	12	0,25	CD4E3
5	50	4	∞	96	974,90	973,70	-0,12	42	0,24	BC3D
6	75	5	∞	84	1571,20	1588,80	1,12	90	0,36	BCDE5
7	100	2	∞	60	2313,70	2316,40	0,12	84	0,57	D3E8
8	100	2	∞	120	1459,30	1483,80	1,68	180	0,64	CD2E3
9	100	3	∞	60	2224,00	2258,50	1,55	126	0,70	CE10
10	100	4	∞	60	2249,20	2268,30	0,85	126	0,85	BC2D2E7
11	249	2	310	300	6610,10	6666,30	0,85	1290	7,46	ABC5D6E8
12	249	3	310	300	6068,80	6100,10	0,52	438	8,22	AC3D7E9
13	249	4	310	300	5943,30	6014,50	1,20	1140	9,86	BCD4E13
14	249	5	310	300	5882,80	5990,70	1,83	384	8,24	C8D7E6
15	80	2	∞	36	2076,20	2101,60	1,22	72	0,42	B3C4D
16	80	2	200	36	2096,40	2102,20	0,28	66	0,36	BC7
17	80	2	180	36	2139,30	2165,40	1,22	66	0,41	C8
18	160	4	∞	36	4001,80	4012,30	0,26	468	0,66	A2B4C8D2
19	160	4	200	36	4148,70	4215,30	1,61	162	3,42	A2B2C10D2
20	160	4	180	36	4338,20	4379,80	0,96	198	3,76	A2BC14
21	240	6	∞	36	5970,20	6021,40	0,86	288	5,96	A2B4C12D4E
22	240	6	200	36	6196,30	6214,30	0,29	306	8,95	A5B2C12D5
23	240	6	180	36	6493,10	6589,70	1,49	318	9,03	A2BC22
24	360	9	∞	36	8883,10	8921,30	0,43	654	5,86	AB5C13D6E6
25	360	9	200	36	9263,00	9298,40	0,38	3324	9,66	A8B6C15D8
26	360	9	180	36	9706,60	9862,50	1,61	294	10,66	C36
Ortalama Değerler:					4121,18	4160,23	0,97	398,31	3,75	

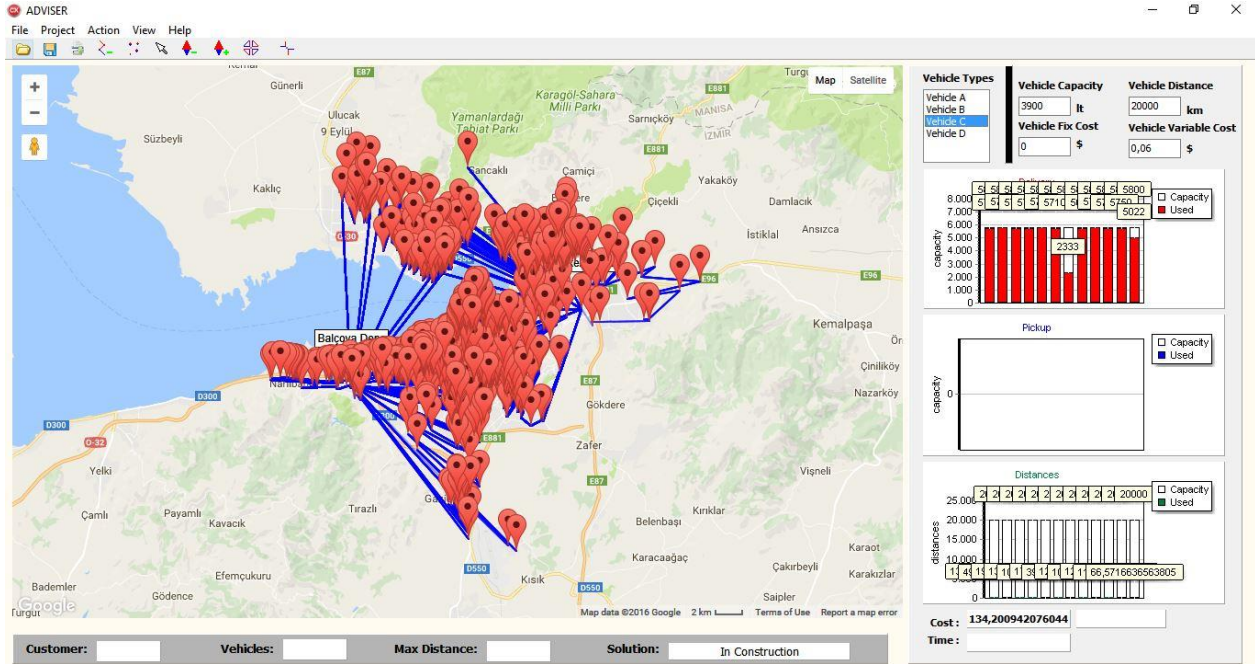
Tablo 7. HAÇARP için yeni ARAHPA2 Deney Sonuçları

Problem	N	m	TD	Q1	Toplam Maliyet				CPU (sn)			Araç Kombinasyonları
					Salhi vd.	Etkileşim Öncesi	GRAMPS2	GAP (%)	Salhi vd.	Etkileşim Öncesi	GRAMPS	
1	55	4	∞	60	1419,70	1506,07	1408,42	-0,79	102	0,31	0,28	E,E3,E2D,E
2	85	3	∞	60	2131,90	2186,51	2148,84	0,79	84	0,69	0,64	E6,E3A,BE2
3	85	3	∞	96	1489,60	1587,07	1504,25	0,98	42	0,66	0,59	E4,E2,E2
4	50	4	∞	48	1499,30	1627,04	1515,81	1,10	12	0,27	0,30	BD,CE2,E2,E
5	50	4	∞	96	974,90	1093,03	969,29	-0,58	42	0,25	0,23	E,C,D,E
6	75	5	∞	84	1571,20	1713,36	1642,56	4,54	90	0,53	0,48	E3D,E2D,E4
7	100	2	∞	60	2313,70	2431,39	2337,37	1,02	84	0,94	0,84	E3,E3,E2D,DE
8	100	2	∞	120	1459,30	1601,36	1490,65	2,15	180	0,81	0,81	CE2,BE2
9	100	3	∞	60	2224,00	2359,89	2253,90	1,34	126	0,94	0,91	DE3,DE2,E4
10	100	4	∞	60	2249,20	2351,88	2229,86	-0,86	126	0,94	0,88	E3,E3,DE2,DE
11	249	2	310	300	6610,10	7712,63	7163,16	8,37	1290	15,89	11,28	E4B3C4D,E4AB2C2D3
12	249	3	310	300	6068,80	6626,25	6355,02	4,72	438	14,05	13,61	E5C2D2,ABC2D2E6
13	249	4	310	300	5943,30	6396,67	6128,64	3,12	1140	12,06	11,22	CDE4,BDE3,DE4,BE3
14	249	5	310	300	5882,80	6366,98	6130,39	4,21	384	11,80	11,56	CE3,DE3,E3,ABDE3,CDE2
15	80	2	∞	36	2076,20	2176,70	2142,72	3,20	72	0,55	0,53	BE2,CE2
16	80	2	200	36	2096,40	2387,52	2107,65	0,54	66	0,61	6,41	BC3,ABCE
17	80	2	180	36	2139,30	2537,60	2242,82	4,84	66	0,62	0,97	A2BCD,A2BC2
18	160	4	∞	36	4001,80	4401,48	4183,80	4,55	468	3,56	3,27	CE2,CE2,CE2,CE2
19	160	4	200	36	4148,70	4648,92	4311,61	3,93	162	3,69	32,64	B2CD,BC2D,B2C2,AD2E
20	160	4	180	36	4338,20	4923,66	4563,03	5,18	198	3,61	9,36	BC2E,A3B3,A2C3,ABC3
21	240	6	∞	36	5970,20	6474,85	6324,71	5,94	288	12,44	12,59	DE2,CE2,DE2,DE2,CE2,DE2
22	240	6	200	36	6196,30	6856,24	6464,56	4,33	306	11,64	108,52	ACD2,ABD2,A2B2D,AB4,ACDE,ABDE
23	240	6	180	36	6493,10	7663,97	6726,33	3,59	318	11,34	99,88	A2BC2,A2BC2,C4,BC2E,A2C2D,BCDE
24	360	9	∞	36	8883,10	9640,16	9502,29	6,97	654	49,02	47,66	E3,DE2,DE2,BE2,DE2,CE2,E3,CE2,DE2
25	360	9	200	36	9263,00	10597,95	9967,53	7,61	3324	37,86	176,44	A3CD,ACDE,BC2D,ABC2D,AC2E,AB2CD,ABCE,BCE2,BD2E
26	360	9	180	36	9706,60	11698,46	10347,26	6,60	294	38,45	182,55	A2BCD,AC2D,A2BCD,B2C3,A2BCE,A2C2E,B2C2D,A2BD2,A2BD
Ortalama Değerler:					4121,18	4598,76		3,36	398,31	8,98	28,25	

Tablo 7' de HAÇARP örnek problemleri için verilen sonuçlara bakıldığında önerilen yeni ARAHPA2 sezgiseli ile üç problem için (#1, #5 ve #10) bilinen en iyi çözümlerden sırasıyla %0,79, %0,58 ve %0,86 daha iyi sonuç bulunmuştur ve literatüre yeni en iyi çözümler olarak geçecektir. Ancak ortalamada bilinen en iyi çözümlerden ortalama %3,36 uzaklaşmıştır ve bu sonuçlara ortalama %28,25 sn. gibi çok kısa bir CPU zamanında ulaşılmıştır. Şu aşamada elde edilen sonuçlar önerilen sezgiselin performansının yüksek olduğunu göstermektedir ancak büyük ölçütlü problemlerde daha sonraki çalışmalarda daha farklı yerel arama yöntemlerinin kullanılmasının gerekliliği ortaya çıkmıştır. Ancak son haliyle bile ARAHPA2 sezgiselinin KDS içinde kullanılarak kullanıcıyla interaktif şekilde çalışması halinde çok kısa sürelerde daha iyi sonuçlar alınması beklenmektedir.

4.3 EDAK İzmir Dağıtım Problemi Sonuçları

Projenin sonunda elde edilen KDS, proje ortağımız EDAK'ın İzmir ili içinde iki depodan, dört farklı araç tipi ile 939 eczaneye yaptığı günlük ilaç dağıtım problemini çözmek için kullanılmıştır. Şekil 10'da görüldüğü üzere bu gerçek hayat problemini çözmek için KDS'nin arayüzüne gerçek harita eklenmiş ve uzaklıklar gerçek uzaklık olarak kullanılmıştır. Elde edilen rotalamanın EDAK da elle yapılan rotalamadan ortalama %2 daha iyi olduğu bulunmuştur.



Şekil 10. EDAK İzmir ili ilaç dağıtım problemi çözümü

5. SONUÇ

Bu final raporu Heterojen Araç-filolu Çok-depolu Araç Rotalama Problemi (HAÇARP) için Görsel Etkileşimli Sezgisel Çözüm Yöntemleri başlıklı projenin 28 aylık döneminde gerçekleştirilen çalışmaların detaylarını içermektedir.

Projenin ilk bir yıllık döneminde, ilk olarak HAÇARP için literatür taraması detaylı bir şekilde yapılmış, problemin en gelişmiş matematiksel programlama modeli incelenmiş ve çalışma esnasında kullanılacak problem kümeleri ve yöntemleri net bir şekilde belirlenmiştir. İkinci olarak, Geri-toplamalı Heterojen Araç-filolu Çok-depolu Araç Rotalama Problemi (GHAÇARP) için geniş kapsamlı bir literatür taraması yapılmış ve bu problem için literatürde ilk defa olmak üzere, temel

ve gevşetilmiş iki adet yeni matematiksel model geliştirilmiştir. GHAÇARP modellerinin doğruluğunun test edilmesi ve görsel etkileşimli sezgisel çözüm yaklaşımının netleşmesi için yeni oluşturulan matematiksel modeller için literatürdeki problem kümeleri de göz önüne alınarak yeni problem kümeleri tanımlanmıştır. Tanımlanan problemler önerilen modellerle çözülmüş elde edilen alt ve üst sınır değerleri raporlanmıştır.

Daha sonra, bundan önceki raporda detaylı olarak anlatılan Görsel Etkileşimli Sezgisel Çözüm yaklaşımları kullanacak olan Karar Destek Sistemi (KDS) içinde kullanılması planlanan sezgisel yaklaşımlar geliştirilmiştir. Başka bir deyişle projenin 13.ve 18.ayları arasında ağırlıklı olarak HAÇARP ve GHAÇARP için gerçek yaşam problemlerini kısa sürelerde çözmemize yardım edecek sezgisel yaklaşımlar geliştirilmiştir. İlk olarak, Aç-gözlü Rastgele Adaptif Hafıza Programlama Arama (ARAHPA) üst-sezgiseli ve sonrasında gerekli görülen düzeltmeler ve eklemeler yapılarak ARAHPA2 üst-sezgiseli geliştirilmiştir. Geliştirilen sezgiseller HAÇARP için literatürde verilen problem örnekleri üzerinde, GHAÇARP için proje kapsamında geliştirilen problem örnekleri üzerinde denenmişlerdir. Önerilen sezgiseller, HAÇARP problemlerinde literatürde verilen çözümlere hız anlamında üstünlük sağlamış, ARAHPA literatürde verilen sonuçlara ortalama % 0,97 daha uzak sonuçlar elde ederken ARAHPA2 ortalama %3,36 daha uzak çözümler elde etmiş ve dahası ARAHPA bir problem (#5) için %0,12 daha iyi sonuç bulurken ARAHPA2 literatürdeki üç problem için (#1, #5, #10)sırasıyla %0,79, %0,58 ve %0,86daha iyi sonuçlar veren çözümler bulabilmiştir. GHAÇARP problemi için ARAHPA sezgiseli GAMS ile bulunan sonuçlardan ortalama %12,33 daha iyi sonuçlar bulurken ARAHPA2 ortalama %11,87 daha iyi sonuçlar bulmuştur. Her iki sezgisel GAMS ile bulunan on en iyi çözümü kısa sürede bulma başarısını göstermiştir.

Görsel Etkileşimli Sezgisel Çözüm yaklaşımları kullanacak olan Karar Destek Sistemi (KDS) içinde HAÇARP ve GHAÇARP problemlerini çözmek için KDS'nin öncelikle ara-yüzü modifiye edilmiş ve sonrasında KDS içinde çağrılacak olan ARAHPA üst-sezgiseli yeniden detaylı bir şekilde yapılandırılmış ve geliştirilerek ARAHPA2 üst-sezgiseli tanımlanmıştır. KDS'nin veritabanı problemlerin yapısına göre yeniden oluşturulmuştur. Böylelikle, geliştirilen sezgisel metotların KDS içerisinde uyumlu bir şekilde çalışabilmesi ve deneylerin başarıyla tamamlanması sağlanmıştır.

Proje kapsamında geliştirilen KDS kullanılarak proje destekçimiz EDAK'ın İzmir ili içindeki iki depodan 939 eczaneye bir gün içinde yapılan ilaç dağıtım problemi çözülmüştür. Sonuçların paylaşımı sonunda yaklaşık %2 'lik bir iyileştirme ve yüksek zaman kazancı ile projemiz başarıyla sonuçlanmıştır.

KAYNAKLAR

ADELZADEH, M., Asl, V. M. ve Koosha, M., A mathematical model and a solving procedure for multi-depot vehicle routing problem with fuzzy time window and heterogeneous vehicle, *International Journal Advanced Manufacturing Technologies* , 75,793–802, (2014).

AHMADI, S. ve Osman, I. H., Greedy random adaptive memory programming search for the capacitated clustering problem clustering problem, *European Journal of Operational Research*, 162, 1, 30–44, (2005).

ARAS, N., Aksen, D., Tekin, M. T., Selective multi-depot vehicle routing problem with pricing, *Transportation Research Part C*, 19, 866-884, (2011).

BALL, M. O., Golden, B. L., Assad, A. A., Bodin, L. D., Planning for truck fleet size in the presence of a common-carrier option, *Decision Sciences*, 14, 103–120, (1983).

BENSLIMANE, M. T. ve Benadada, Y., Ant Colony Algorithm for the Multi-Depot Vehicle Routing Problem in Large Quantities by a Heterogeneous Fleet of Vehicles, *INFOR*, 51, 1, 31–40, (2013).

BETTINELLI, A., Ceselli, A. ve Righini, G., A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time Windows, *Transportation Research Part C*, 19, 723–740, (2011).

BIANCHESSI, N. ve Righini, G., Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery, *Computers and Operations Research*, 34, 2, 578- 594, (2007).

BRANDAO, J., A new tabu search algorithm for the vehicle routing problem with backhauls, *European Journal of Operational Research*, 173, 540–555, (2006).

COSCO, D.O., Golden, B.L. ve Wasil, E.A., Vehicle routing with backhauls: models, algorithms and case studies, In: Vehicle routing: method and studies, Amsterdam: Elsevier, 127–47, (1988).

CHAO, I.-M., Golden, B. L., Wasil, E., A new heuristic for the multi-depot vehicle routing problem that improves upon best-known solutions, American Journal of Mathematical and Management Sciences, 13, 371–406, (1993).

CHEN, J. F. ve Wu, T. H., Vehicle routing problem with simultaneous deliveries and pickups, The Journal of the Operational Research Society, 57, 5, 579–587, (2006).

CLARKE, G., Wright, J., Scheduling of vehicles from a central depot to a number of delivery points, Operations Research, 12 (4), 568–581, (1964).

CORDEAU, J. F., Gendreau, M., Laporte, G., A tabu search heuristic for periodic and multi-depot vehicle routing problems, Networks, 30, 105–119, (1997).

CRISPIM, J. ve Brandao, J., Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls, The Journal of the Operational Research Society, 56, 11, 1296-1302, (2005).

CUERVO, D. P., Goos, P., Sörensen, K. ve Arraiz, E., An iterated local search algorithm for the vehicle routing problem with backhauls, European Journal of Operational Research, 237, 454-464, (2014).

DANTZIG, G. B., Ramser, J. H., The truck dispatching problem, Management Science, 6, 1, 80–91, (1959).

DEIF, I. ve Bodin, L., Extension of the Clarke and Wright algorithm for solving the vehicle routing problem with backhauling, In: Kidder A, editor, Proceedings of the Babson conference on software uses in transportation and logistic management, Babson Park, USA, 75–96, (1984).

DELLAMICO, M., Righini, G. ve Salani, M., A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection, *Transportation Science*, 40, 2, 235-247, (2006).

DETHLOFF, J., Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up, *OR Spektrum*, 23, 1, 79-96, (2001).

DONDO, R., Cerdá, J., A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows, *European Journal of Operational Research*, 176 (3), 1478-1507, (2007).

GAJPAL, Y. ve Abad, P., An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup, *Computers and Operations Research*, 36, 3215-3223, (2009).

GANESH, K. ve Narendran, T. T., CLOVES: A cluster-and-search heuristic to solve the vehicle routing problem with delivery and pick-up, *European Journal of Operational Research*, 178, 699–717, (2007).

GEETHA, S., Vanathi, P. T., Poonthalir, G., Metaheuristic approach for the multi-depot vehicle routing problem, *Applied Artificial Intelligence*, 26:878–901, (2012).

GILLET, B., Johnson, J., The multiple terminal vehicle dispatching algorithm, *Omega*, 4, 711–718, (1976).

GOETSCHALCKX, M. ve Jacobs-Blecha, C., The vehicle routing problem with backhauls, *European Journal of Operational Research*, 42, 39–51, (1989).

GOLDEN, B. L., Magnanti, T. L., Nguyen, H. Q., Implementing vehicle routing algorithm, *Networks*, 7, 113–148, (1977).

GOLDEN, B., Assad, A., Levy, L. ve Gheysens, F. G., The fleet size and mix vehicle routing problem, *Computers & Operations Research*, 11, 49–66, (1984).

GOLDEN, B., Baker, E., Alfaro, J. ve Schaffer, J., The vehicle routing problem with backhauling: two approaches, In: Hammesfahr R, editor, Proceedings of the XXI annual meeting of S.E. Times, (1985).

GULCZYNSKI, D., Golden, B., Wasil, E., The multi-depot split delivery vehicle routing problem: An integer programming-based heuristic, new test problems, and computational results, Computers & Industrial Engineering, 61, 794–804, (2011).

HAGHIGHI, M. M. S., Zahedi, M. H., Ghazizadeh, M., A multi level priority clustering GA based approach for solving heterogeneous Vehicle Routing Problem (PCGVRP), T. Sobh (Ed.) Innovations and Advances in Computer Sciences and Engineering, Springer Netherlands, 331-335, (2010).

HALSE, K., Modeling and solving complex vehicle routing problems, PhD thesis, Institute of Mathematical Statistics and Operations Research, Technical University of Denmark, Lyngby, (1992).

HO, W., Ho, T. S., Ji, P., Lau, C. W., A hybrid genetic algorithm for the multi-depot vehicle routing problem, Applied Artificial Intelligence, 21, 548–57, (2008).

IRNICH, S., A multi-depot pickup and delivery problem with a single hub and heterogeneous vehicles, European Journal of Operational Research, 122, 310-328, (2000).

KUO, Y., Wang, C-C., A variable neighborhood search for the multi-depot vehicle routing problem with loading cost, Expert Systems with Applications, 39, 6949-6954, (2012).

LAPORTE, G., Nobert, Y., Arpin, D., Optimal solutions to capacitated multidepot vehicle routing problems, Congressus Numerantium, 44, 283–292, (1984).

LAPORTE, G., Nobert, Y., Taillefer, S., Solving a family of multi-depot vehicle routing and location-routing problems, Transportation Science, 22, 161–172, (1988).

LI, J., Pardalos, P. M., Sun, H., Pei, J. ve Zhang, Y., Iterated local search embedded adaptive neighborhood selection approach for the multi-depot vehicle routing problem with simultaneous deliveries and pickups, *Expert Systems with Applications*, 42, 3551–3561, (2015).

MANCINI, S., A real-life Multi Depot Multi Period Vehicle Routing Problem with a Heterogeneous Fleet: Formulation and Adaptive Large Neighborhood Search based Matheuristic, *Transportation Research Part C*, (2015), <http://dx.doi.org/10.1016/j.trc.2015.06.016>

MIN, H., The multiple vehicle routing problem with simultaneous delivery and pickup, *Transportation Research Part A*, 23, 377–386, (1989).

MINGOZZI, A., Giorgi, S. ve Baldacci, R., An exact method for the vehicle routing problem with backhauls, *Transportation Science*, 33, 3, 315–329, (1999).

MIRABI, M., Ghomi, S. M. T. F., Jolai, F., Efficient stochastic hybrid heuristics for the multi-depot vehicle routing problem, *Robotics and Computer-Integrated Manufacturing*, 26, 564-569, (2010).

MONTOYA-TORRES, J. R., Franco, J. L., Isaza, S. N., Jimenez, H. F. ve Herazo-Padilla, N., A literature review on the vehicle routing problem with multiple depots, *Computers & Industrial Engineering*, 79, 115–129, (2015).

MOSHEIOV, G., Vehicle routing with pick-up and delivery: tour-partitioning heuristics, *Computers Industrial Engineering*, 34, 3, 669-684, (1998).

NAGY, G. ve Salhi, S., Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries, *European Journal of Operational Research*, 162, 1, 126–141, (2005).

OLIVEIRA, F. B., Enayatifar, R., Sadaei, H. J., Guimaraes, F. G. ve Potvin, J., AcooperativecoevolutionaryalgorithmfortheMulti-DepotVehicleRoutingProblem, *Expert SystemsWithApplications*, 43, 117–130, (2016).

OSMAN, I. ve Wassan, N., A reactive tabu search for the vehicle routing problem with back-hauls, *Journal of Scheduling*, 5, 4, 263–285, (2002).

POTVIN, J., Rousseau, J., A parallel route building algorithm for the vrptw, *European Journal of Operational Research*, 66, 331–340, (1993).

RAFT, O. M., A modular algorithm for an extended vehicle scheduling problem, *European Journal of Operational Research*, 11, 67–76, (1982).

RENAUD, J., Laporte, G., Boctor, F. F., A tabu search heuristic for the multi-depot vehicle routing problem, *Computers & Operations Research*, 23, 229–235, (1996).

SALHI, S., Sari, M., A multi-level composite heuristic for the multidepot vehicle fleet mix problem, *European Journal of Operational Research*, 103, 95–112, (1997).

SALHI, S. ve Nagy, G., A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling, *Journal of the Operational Research Society*, 50, 1034–1042, (1999).

SALHI, S., Wassan, N. ve Hajarat, M., The Fleet Size and Mix Vehicle Routing Problem with Backhauls: Formulation and Set Partitioning-based Heuristics, *Transportation Research Part E*, 56, 22–35, (2013).

SALHI, S., Imran, A., Wassan, N. A., The multi-depot vehicle routing problem with heterogeneous vehicle fleet: Formulation and a variable neighborhood search implementation, *Computers & Operations Research*, 52, 315-325, (2014).

TAILLARD, E. D., A Heuristic Column Generation Method for the Heterogeneous Fleet VRP, Publication CRT-03-96, Universite Montreal, (1996).

TANG, F. A. ve Galvao, R. D., A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service, *Computers and Operations Research*, 33, 3, 595–619, (2006).

THANGIAH, S. R., Salhi, S., Genetic clustering: An adaptive heuristic for the multidepot vehicle routing problem, *Applied Artificial Intelligence*, 15, 361–383, (2001).

TILLMAN, F. A., The multiple terminal delivery problem with probabilistic demands, *Transportation Science*, 3, 192–204, (1969).

TOTH, P. ve Vigo, D., A heuristic algorithm for the vehicle routing problem with backhauls, *Advanced Methods in Transportation Analysis*, 585-608, (1996).

TOTH, P. ve Vigo, D., An exact algorithm for the vehicle routing problem with backhauls, *Transportation Science*, 31, 4, 372–385, (1997).

TOTH, P. ve Vigo, D., *The vehicle routing problem*, Philadelphia: SIAM Monographs on Discrete Mathematics and Applications, (2002).

TÜTÜNCÜ, G. Y., Carreto, C. A. C., Baker, B. M., A visual interactive approach to classical and mixed vehicle routing problems with backhauls, *Omega*, 37, 138 – 154, (2009).

TÜTÜNCÜ, G. Y., An interactive GRAMPS algorithm for the heterogeneous fixed fleet vehicle routing problem with and without backhauls, *European Journal of Operational Research*, 201, 593–600, (2010).

WADE, A. C. ve Salhi, S., *An ant system algorithm for the mixed vehicle routing problem with backhauls*, Metaheuristics computer decision-making, Dordrecht: Kluwer Academic Publishers, 699–719, (2004).

WASSAN, N., Reactive tabu adaptive memory programming search for the vehicle routing problem with backhauls, *Journal of the Operational Research Society*, 58, 12, 1630–1641, (2007).

WASSAN, N., Wassan, A. ve Nagy, G., A reactive tabu search algorithm for the vehicle routing problem with simultaneous pickups and deliveries, *Journal of Combinatorial Optimization*, 15, 4, 368–386, (2008).

WREN, A., Holliday, A., Computer scheduling of vehicles from one or more depots to a number of delivery points, *Operational Research Quarterly*, 23, 333–344, (1972).

XU, Y., Wang, L., ve Yang, Y., A New Variable Neighborhood Search Algorithm for the Multi Depot Heterogeneous Vehicle Routing Problem with Time Windows, *Electronic Notes in Discrete Mathematics*, 39, 289–296, (2012).

XU, Y. ve Jiang, W., An Improved Variable Neighborhood Search Algorithm for Multi Depot Heterogeneous Vehicle Routing Problem based on Hybrid Operators, *International Journal of Control and Automation*, 7, 3, 299-316, (2014).

YAMAN, H., Formulations and valid inequalities for the heterogeneous vehicle routing problem, *Mathematical Programming Series A*, 106, 365-390, (2006).

YÜCENUR, G. N., Demirel, N. Ç., A new geometric shape-based genetic clustering algorithm for the multi-depot vehicle routing problem, *Expert Systems with Applications*, 38, 11859–11865, (2011).

ZACHARIADIS, E. ve Kiranoudis, C., An effective local search approach for the vehicle routing problem with backhauls, *Expert Systems with Applications*, 39, 3, 3174–3184, (2012).

TÜBİTAK
PROJE ÖZET BİLGİ FORMU

Proje Yürütücüsü:	Doç. Dr. GÖZDE YAZGI TÜTÜNCÜ
Proje No:	213M438
Proje Başlığı:	Heterojen Araç Filolu Çok Depolu Araç Rotalama Problemi İçin Görsel Etkileşimli Sezgisel Çözüm Yöntemleri
Proje Türü:	1001 - Araştırma
Proje Süresi:	24
Araştırmacılar:	
Danışmanlar:	
Projenin Yürütüldüğü Kuruluş ve Adresi:	İZMİR EKONOMİ Ü. FEN EDEBİYAT F. MATEMATİK B.
Projenin Başlangıç ve Bitiş Tarihleri:	15/03/2014 - 15/07/2016
Onaylanan Bütçe:	116870.0
Harcanan Bütçe:	90495.65
Öz:	<p>Araç Rotalama Problemleri (ARP) her birine belli büyüklükteki ve nitelikteki ürün ya da yolcuların araç filosu ile ulaştırılması gereken birçok sayıda müşteri ya da durak içerir. ARP yöneylem araştırması ve yönetim bilimleri disiplinlerinin hem akademik araştırma hem de uygulama açısından en önemli alanlarından birisidir. Gerçek hayatta ARP problemleri temel ARP ?nin ötesinde ek kısıtlar ve zorluklar içermektedir. Bu projede gerçek hayatta çokça rastlanan ve birçok tedarik firması tarafından göz önüne alınıp önemsenmeyen Heterojen Araç-filolu Çok-depolu Araç Rotalama Problemi (HAÇARP) ve HAÇARP ?in yaygın kullanımlı bir versiyonu olan Geri-toplamalı HAÇARP (GHAÇARP) ele alınmıştır.</p> <p>Bu projede öncelikle HAÇARP için yeni sezgisel yöntemler geliştirilmiş, geliştirilen sezgiseller literatürdeki karşılaştırmalı problem örnekleriyle test edilmiş ve gerçek hayat problemlerini çözmek üzere uygulamaya sunulmuştur. Daha sonra GHAÇARP için yeni bir matematiksel model önerilmiş, gerçek hayat problemlerini baz alan yeni problem örnekleri oluşturulmuş ve önerilen model kullanılarak bu problemlerin gevşetilmiş çözümleri elde edilmiştir. Buna ek olarak, HAÇARP için geliştirilen yeni sezgisel yöntemler göz önüne alınarak GHAÇARP için de yeni bir sezgisel geliştirilmiş ve önerilen problem örnekleriyle test edilmiş ve gerçek hayat problemlerini çözmek üzere uygulamaya sunulmuştur.</p> <p>Sonuç olarak günlük hayatta çeşitli firmalarca kullanılabilir, farklı araç rotalama problemlerinin çözümü için uyarlanabilir, görsel etkileşimli ve sezgisel yöntemlere dayalı ve ele alınan problemleri iyi sonuçlarla çözebilen bir Karar Destek Sistemi geliştirilmiştir</p>
Anahtar Kelimeler:	Çok Depolu Heterojen Araç Rotalama, Sezgisel Yöntemler, Karar Destek Sistemi, Matematiksel Modelleme
Fikri Ürün Bildirim Formu Sunuldu Mu?:	Hayır