



**OTOMOTİV SEKTÖRÜNDE BİR ARAÇ ROTALAMA  
PROBLEMİ UYGULAMASI**

**Aybüke KUMAŞ**



T.C.  
BURSA ULUDAĞ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**OTOMOTİV SEKTÖRÜNDE BİR ARAÇ ROTALAMA PROBLEMİ  
UYGULAMASI**

AYBÜKE KUMAŞ  
0009-0007-5231-3032

Prof. Dr. ÂLİ YURDUN ORBAK  
(Danışman)

YÜKSEK LİSANS TEZİ  
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA – 2024  
Her Hakkı Saklıdır

## ÖZET

Yüksek Lisans

OTOMOTİV SEKTÖRÜNDE BİR ARAÇ ROTALAMA PROBLEMİ UYGULAMASI

**Aybüke KUMAŞ**

Bursa Uludağ Üniversitesi  
Fen Bilimleri Enstitüsü  
Endüstri Mühendisliği Anabilim Dalı

**Danışman:** Prof. Dr. Âli Yurdun ORBAK

Tedarik zinciri yönetimi, bir ürün veya hizmetin hammaddeden başlayarak son müşteriye ulaşana kadar olan operasyonları, bilgi akışını ve fiziksel dağıtımını içeren bir sistemdir. Tedarik zinciri yönetimi, maliyetleri azaltmayı, verimliliği artırmayı ve müşteri memnuniyetini yükseltmeyi hedefler, böylelikle stratejik bir yaklaşım sunar. Günümüzde gelişen dünyada ve ilerleyen teknoloji ile tedarik zinciri düşük maliyetli ve fayda sağlayan bir strateji olarak değerlendirilmektedir. Bu çalışmada, bir otomotiv fabrikasının milk-run sürecindeki araç rotalama problemi, klasik sezgisel çözüm yöntemlerinden Clarke-Wright Tasarruf Algoritması kullanılarak çözümlenmiştir. Doğrusal Regresyon modeli kullanılarak tedarikçi talep miktarları için tahminleme yapılmıştır. Meta-sezgisel çözüm yöntemleri olan Genetik Algoritma ve Tavlama Benzetimi Algoritması, problemin farklı çözümleri için uygulanmıştır. Ele alınan araç rotalama problemi için optimal rotalar belirlenmiştir ve en etkili algoritma tespit edilmiştir.

Bu çalışmada önerilen uygulama ile, tedarik zinciri yönetiminde araç rotalama problemine farklı bir bakış açısı kazandırılmıştır. Araç rotalama problemi ile elde edilen rotalar sayesinde, tedarik zinciri yönetimindeki maliyetler minimize edilerek operasyonel verimlilik artırmıştır. Sonuç olarak, bu çalışma klasik sezgisel çözüm yöntemleri ve meta-sezgisel çözüm yöntemlerinin tedarik zinciri yönetimi alanındaki milk-run araç rotalama problemlerine çözüm getirirken, klasik sezgisel ve meta-sezgisel çözüm yöntemlerinin entegrasyonun stratejik bir yaklaşım olarak ne kadar değerli olduğunu vurgulamaktadır.

**Anahtar Kelimeler:** Araç rotalama problemi, tedarik zinciri yönetimi, otomotiv, optimizasyon, Clarke-Wright Tasarruf Algoritması, Tavlama Benzetimi Algoritması, Genetik Algoritma.

**2024, ix + 79 sayfa.**

## ABSTRACT

MSc Thesis

### A VEHICLE ROUTING PROBLEM APPLICATION IN THE AUTOMOTIVE INDUSTRY

**Aybüke KUMAŞ**

Bursa Uludağ University  
Graduate School of Natural and Applied Sciences  
Department of Industrial Engineering

**Supervisor:** Prof. Dr. Âli Yurdun ORBAK

Supply chain management is a system that includes the operations, information flow and physical distribution of a product or service from raw materials to the end customer. Supply chain management aims to reduce costs, increase efficiency and improve customer satisfaction, thus offering a strategic approach. In today's developing world and with advancing technology, supply chain is considered as a cost-effective and beneficial strategy. In this study, the vehicle routing problem in the milk-run process of an automotive factory is solved using the Clarke-Wright Savings Algorithm, one of the classical heuristic solution methods. Linear Regression model is used to predict the supplier demand quantities. The meta-heuristics Genetic Algorithm and Annealing Simulation Algorithm are applied for different solutions of the problem. The optimal routes for the considered vehicle routing problem were determined and the most efficient algorithm was identified.

The application proposed in this study provides a different perspective to the vehicle routing problem in supply chain management. Thanks to the routes obtained with the vehicle routing problem, costs in supply chain management are minimized and operational efficiency is increased. As a result, this study emphasizes how valuable the integration of classical heuristics and meta-heuristics is as a strategic approach when solving milk-run vehicle routing problems in supply chain management.

**Key words:** Vehicle routing problem, supply chain management, automotive, optimization, Clarke-Wright Savings Algorithm, Simulated Annealing Algorithm, Genetic Algorithm.

**2024, ix + 79 pages.**

## TEŐEKKÖR

Bu tez alıőmam boyunca, beni ynlendiren, destekleyen ve yol gsteren deęerli danıőmanım Prof. Dr. Āli Yurdun ORBAK'a sonsuz teőekkrlerimi sunarım.

Hayatım boyunca her zaman yanımda olan, hayatımın her aőamasında beni destekleyen baőta annem Selda KUMAŐ olmak zere deęerli aileme sonsuz sevgi ve teőekkrlerimi sunarım.

Bu aőamaya gelene kadar zerimde emeęi olan btn hocalarıma ve arkadaőlarıma destekleri iin en iten teőekkrlerimi sunarım.

Aybke KUMAŐ  
14/06/2024



## İÇİNDEKİLER

	Sayfa
ÖZET.....	i
ABSTRACT.....	ii
TEŞEKKÜR.....	iii
SİMGELER ve KISALTMALAR DİZİNİ.....	vi
1. GİRİŞ.....	1
2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI.....	5
3. MATERYAL VE YÖNTEM.....	9
3.1. Araç Rotalama Problemleri.....	10
3.1.2. Araç Rotalama Problemlerinin Uygulama Alanları.....	12
3.1.3. Araç Rotalama Problemleri Türleri.....	13
3.1.3.1. Klasik Araç Rotalama Problemi.....	13
3.1.3.2. Zaman Pencereci Araç Rotalama Problemi.....	15
3.1.3.3. Heterojen Filolu Araç Rotalama Problemi.....	17
3.1.3.4. Talebi Belirsiz Araç Rotalama Problemi.....	17
3.1.3.5. Bölünmüş Talebe Sahip Araç Rotalama Problemi.....	18
3.1.3.6. Mesafe Kısıtlı Araç Rotalama Problemi.....	18
3.1.3.7. Çoklu Depoya Sahip Araç Rotalama Problemi.....	18
3.1.3.8. Geri Toplamalı Araç Rotalama Problemi.....	19
3.1.3.9. Eş Zamanlı Topla Dağıt Araç Rotalama Problemi.....	19
3.1.3.10. Dinamik Araç Rotalama Problemi.....	19
3.1.4. Araç Rotalama Problemleri Çözüm Yöntemleri.....	20
3.1.4.1. Dal ve Sınır Algoritması.....	21
3.1.4.2. Dal ve Kesme Algoritması.....	21
3.1.4.3. Dinamik Programlama.....	22
3.1.4.4. Kesme Düzlemi Algoritması.....	22
3.1.4.5. Dantzig ve Ramser Yöntemi.....	23
3.1.4.6. Clarke-Wright Tasarruf Algoritması.....	23
3.1.4.7. K-En Yakın Komşu Algoritması.....	25
3.1.4.8. 2-Opt İyileştirme Operasyonu.....	25
3.1.4.9. Süpürme(Sweep) Algoritması.....	26
3.1.4.10. Fisher ve Jaikumar Algoritması.....	26
3.1.4.11. Tabu Arama Algoritması.....	28
3.1.4.12. Genetik Algoritma.....	28
3.1.4.13. Parçacık Sürü Optimizasyonu.....	32
3.1.4.14. Yapay Arı Kolonisi Algoritması.....	32
3.1.4.15. Karınca Kolonisi Algoritması.....	33
3.1.4.16. Tavlama Benzetimi Algoritması.....	33
3.2. Doğrusal Regresyon.....	34
3.3. Otomotiv Sektöründe Araç Rotalama Problemi Uygulaması.....	36
3.3.1. Matematiksel Model.....	36
3.3.2. Problemin ve Kısıtların Tanımlanması.....	37
3.3.3. Metot.....	38
3.3.3.1. Clarke-Wright Tasarruf Algoritması Uygulaması.....	39
3.3.3.2. Tavlama Benzetimi Algoritması Uygulaması.....	44
3.3.3.3. Genetik Algoritma Uygulaması.....	46
3.3.3.4. Doğrusal Regresyon ile Geliştirilmiş Çözüm Uygulaması.....	50

4. BULGULAR.....	57
5. TARTIŞMA ve SONUÇ.....	60
KAYNAKLAR .....	62
EKLER.....	65
EK-1 Tavlama Benzetimi Algoritması Matlab Kodları.....	65
EK-2 Genetik Algoritma Matlab Kodları .....	67
EK-3 Python Kodları .....	71
ÖZGEÇMİŞ .....	79



## SİMGELER ve KISALTMALAR DİZİNİ

<b>Simgeler</b>	<b>Açıklama</b>
$W$	Ağırlık vektörü
$Min Z$	Amaç fonksiyonu
$Q$	Araç kapasitesi
$K$	Araçlar kümesi
$m$	Araç sayısı
$p_{ij}$	Aracın yük miktarı
$A$	Ayrıt kümesi
$k$	Boltzmann sabiti
$Y$	Değişken
$V$	Düğüm kümesi
$\Delta$	Enerji fonksiyonundaki değişim
$F$	Fabrika
$(g_{best})$	Global en iyi parçacık
$E$	Hata
$c_{ij}$	$i$ şehirden $j$ şehrine gitme maliyeti
$d_{ij}$	$i$ şehri ve $j$ şehri arasındaki mesafe
$L_k$	Maksimum tur uzunlukları
$m_{ij}$	Mesafe
$n$	Müşteri sayısı
$q_i$	Müşteri talep miktarı
$[a_i, b_i]$	$i$ . müşterinin zaman penceresi
$a_i$	$i$ . müşterinin en erken servis zamanı
$b_i$	$i$ . müşterinin en geç servis zamanı
$X$	Özellik vektörü
$G$	Serim kümesi
$T$	Sıcaklık parametresi
$d_i$	Talep
$D_j$	Tedarikçiden toplanılacak yük miktarı
$I$	Tedarikçi sayısı
$t_{ij}$	Ulaşım süresi
$x_{ijk}$	Veri kümesindeki değer
$(p_{best})$	Yerel en iyi parçacık

<b>Kısaltmalar</b>	<b>Açıklama</b>
ARP	Araç Rotalama Problemi
ASRP	Asimetrik Araç Rotalama Problemi
AKARP	Asimetrik Klasik Araç Rotalama Problemi
BTARP	Bölünmüş Talebe Sahip Araç Rotalama Problemi
GA	Genetik Algoritma
GSP	Gezgin Satıcı Problemi
GTARP	Geri Toplamalı Araç Rotalama Problemi
HFARP	Heterojen Filolu Araç Rotalama Problemi

KARP	Klasik Araç Rotalama Problemi
KKARP	Karma Kapasiteli Araç Rotalama Problemi
K-means	K-Ortalamlar Algoritması
LR	Lojistik Regresyon
MKARP	Mesafe Kısıtlı Araç Rotalama Problemi
K-NN	K-En Yakın Komşuluk Algoritması
SKARP	Simetrik Klasik Araç Rotalama Problemi
TB	Tavlama Benzetimi Algoritması
TZY	Tedarik Zinciri Yönetimi
ZPARP	Zaman Pencereli Araç Rotalama Problemi



## ÇİZELGELER DİZİNİ

	<b>Sayfa</b>
Çizelge 3. 1. Fabrika ve tedarikçilere ait konum tablosu.....	39
Çizelge 3. 2. Tedarikçi günlük malzeme talep miktarları.....	40
Çizelge 3. 3. Tedarikçiler ve fabrika arasındaki uzaklıklar(km) matrisi.....	41
Çizelge 3. 4. Tasarruf miktarları matrisi.....	42
Çizelge 3. 5. Hesaplanan tasarruf miktarları ve rotalar.....	43
Çizelge 3. 6. Probleme ait başlangıç popülasyonu ve kromozom yapısı.....	47
Çizelge 3. 7. Uygunluk değerleri.....	48
Çizelge 3. 8. Tedarikçilerin bir haftalık talep miktarları.....	51
Çizelge 3. 9. Talep tahminlemesi sonucu oluşturulan araç rotaları.....	55
Çizelge 4. 1. Çözülen algoritmaların sonuçları.....	58



## ŞEKİLLER DİZİNİ

	<b>Sayfa</b>
Şekil 3. 1. Araç rotalama problemine ait şematik gösterim.....	11
Şekil 3. 2. Clarke-Wright tasarruf algoritması şematik gösterimi.....	23
Şekil 3. 3. Clarke-Wright tasarruf algoritması akış şeması.....	24
Şekil 3. 4. K-en yakın komşu algoritması şematik gösterimi.....	25
Şekil 3. 5. 2-opt iyileştirme operasyonu şematik gösterimi.....	26
Şekil 3. 6. Fisher ve Jaikumar algoritması şematik gösterimi.....	27
Şekil 3. 7. Genetik algoritma akış şeması.....	31
Şekil 3. 8. Tavlama benzetimi algoritması akış şeması.....	34
Şekil 3. 9. Clarke-Wright tasarruf algoritması ile elde edilen araç rotası.....	44
Şekil 3. 10. Matlab'da oluşturulan tavlama benzetimi algoritması değişkenleri...	45
Şekil 3. 11. Tavlama benzetimi algoritması ile elde edilen araç rota diyagramı...	46
Şekil 3. 12. Rulet çemberi tekniği ile birey seçilme olasılıkları.....	48
Şekil 3. 13. Matlab'da oluşturulan genetik algoritma model değişkenleri.....	49
Şekil 3. 14. Genetik algoritma ile elde edilen araç rota diyagramı.....	50
Şekil 3. 15. Doğrusal regresyon ile tahminleme yapılan talep miktarları grafiği..	52
Şekil 3. 16. Doğrusal regresyon ile tahminleme yapılan talep miktarları.....	53
Şekil 3. 17. Fabrika ve tedarikçiler arasındaki mesafe matrisi.....	54
Şekil 3. 18. Optimal rotaya ait grafik.....	56

## 1. GİRİŞ

Otomotiv sektörü yarattığı katma değer ile dünya çapında önemli bir etki bırakmış ve günümüzde hala en önemli endüstrilerden biri olarak öne çıkmaktadır. Ekonomik büyümeyi desteklemesi ve diğer sektörlerle de katkı sağlamasıyla otomotiv sektörü, endüstriyel ve sosyal dönüşümün önemli bir unsuru olmuştur. Ancak günümüz rekabetçi ortamında, şirketlerin sadece müşteri memnuniyetini artırmakla kalmayıp aynı zamanda maliyetleri azaltmaları da gerekmektedir. Otomotiv sektörü büyüyen ticaret hacmi ile son yıllarda oldukça önemli bir konumdadır. Otomotiv sektörünün ekonominin en büyük ölçeğini kapsayan sektörlerden biri olması sebebiyle yarattığı katma değer, ülke ekonomisinde oldukça önemlidir.

Bu bağlamda, Tedarik Zinciri Yönetimi gibi stratejik uygulamalar, şirketlerin rekabet avantajı elde etmelerinde kritik bir rol oynamaktadır. Bu uygulamalar, otomotiv sektöründe yer alan işletmelerin rekabet avantajı elde etmesinde kritik bir rol üstlenir. Etkin bir tedarik zinciri yönetimi, malzeme akışının optimize edilmesini ve lojistik operasyonlarının verimli bir şekilde gerçekleştirilmesini sağlar. Bu da şirketlerin sürdürülebilirliklerini artırır ve büyümelerine destek olur.

Tedarik zinciri, hammaddenin temin edilmesi ile başlayan ve ürünün müşteriye ulaştırılmasına kadar geçen sürece ait paydaşların oluşturduğu bir ağıdır. Tedarik Zinciri Yönetimi tedarikçiler ile başlayan ve müşterilerle sonlanan bir zinciri oluşturmaktadır. Bu zincirde oluşabilecek herhangi bir kopukluk tüm sistemi etkileyebileceğinden dolayı, sistemin iyi bir şekilde yönetilmesi gerekir. Verimli ve etkili bir lojistik hizmetini en düşük lojistik maliyetlerle sunmak oldukça önemlidir.

Günümüz dünyasında var olan rekabet ortamı; tedarik zinciri işletmelerini, kapasite dağıtım stratejilerini yeniden gözden geçirmeye teşvik etmektedir. Tedarik zinciri yönetiminde dağıtım fonksiyonu, işletmeler için oldukça önemli olup, stratejik anlam teşkil etmektedir. Bu açıdan, dağıtım ağlarının etkin kullanımı, işletmeler için kritik bir öneme sahiptir. Tedarik zinciri ağlarının optimizasyonu yoluyla dağıtım ağlarının verimliliği maksimum seviyeye çıkarılmaya çalışılmaktadır.

Tedarik zinciri, dinamik ve sürekli deęişen bir yapıya sahiptir. Bu yapı, bilginin, ürünlerin ve malzemelerin karmaşık bir aę üzerinde akışını sağlar ve işletmelerin faaliyetlerini yönlendirir.

Tedarik zinciri süreçlerinin temel amacı, müşteri ihtiyaçlarını karşılayarak verimlilik sağlamaktır. Aynı zamanda müşteri ihtiyaçlarını karşılarırken şirketin karını maksimum seviyede tutabilmektir. Tedarik zincirinin amaçlarından bir dięeri ise ürün ve hizmetlerin en efektif olarak gerekli yerlere ulaştırılmasının sağlanmasıdır. Tedarik zinciri, ürün veya hizmetlerin tedarikçilerden üreticilere, toptancılara, perakendecilere ve en sonunda müşterilere zincir boyunca hareketini sağlar. Etkin ve verimli bir şekilde tedarik zincirinin yönetilmesi, işletmeler açısından büyük bir öneme sahiptir. Organizasyonlar ilk etapta tedarik zinciri ağını kurduklarında tüm alternatiflerden haberdar değildir, bundan dolayı birçok tedarik zinciri optimum düzeyde değildir. Optimum tedarik zincirine ulaşma arayışı ve yapısal çeşitlilikler fayda sağlamaktadır.

Dağıtım sisteminin amacı tedarik noktasından alınan ürün ve hizmetlerin işletmeye getirilmesini sağlamaktır. Artan rekabet ortamından ötürü günümüz işletmeleri için dağıtım planlamasına büyük ölçüde önem vermektedir. Dağıtım sisteminin etkinliği, tedarik zinciri ağının iyi yönetilmesindeki temel faktörlerden biridir. Maliyet avantajı sağlamak amacı ile işletmeler kazanç elde edebilmek için dağıtım planlaması yapmaktadır. Etkin bir dağıtım stratejisi ve planlaması sayesinde dağıtım kanallarından optimum düzeyde yararlanılabilmektedir. Bu şekilde, kaynakların etkin ve verimli kullanılabilmesi sağlanılarak maliyet avantajı sağlanabilmektedir. Böylece, işletmelerin rekabet ortamında kabiliyeti ön plana çıkmaktadır.

Etkili bir dağıtım sistemi oluşturmak için efektif bir konum planlamasıyla dağıtım kanallarından optimum düzeyde fayda sağlanabilmektedir. Kaynakların verimli kullanılması ile işletmelerin rekabet gücü artmaktadır. Dağıtım aracı seçimi, dağıtım planlamasında önemli bir konudur. Sistem verimliliğini direkt olarak etkilemektedir. Tedarik zincirinde dağıtım yapılırken araçların en yüksek etkinlik ile çalışması için dağıtım rotalaması yapılması gerekmektedir. Dağıtım araçlarının etkin bir rotalama ile verimliliği arttırılmaktadır. Etkili bir rotalama ile kullanım kapasitesi maksimize edilip

toplam mesafenin kısaltılması amaçlanmaktadır. Araç rotalamasının etkili bir şekilde gerçekleştirilebilmesi için literatürde çeşitli yöntemler ortaya konulmuştur.

Döngüsel sefer de denilebilen milk-run, belirli bir program dahilinde çeşitli noktalardan aynı veya farklı yükleri toplayarak bir veya birden fazla noktaya teslim etmesi ve teslimatlar sırasında boş kapların toplaması veya tüm teslimatlar bittikten sonra ters yönde ilerleyerek boş kapların geri getirmesi sistemidir (Demir, 2010).

Milk-run sisteminin en önemli faydalarından biri, her bir tedarikçinin ayrı ayrı taşıma maliyetleriyle uğraşmasının yerine bu maliyetlerin bir araya getirilmesi ve birleştirilen kaynağın lojistik giderlerini karşılamak üzere kullanılmasıdır. Milk-run sürecinin verimliliği, tedarik zincirinin organizasyonuna bağlıdır. Otomotiv sektöründe milk-run sistemleri sıklıkla kullanılmaktadır. Bu açıdan otomotiv sektöründe milk-run sistemi verimliliği oldukça önemlidir.

Otomotiv sektörü, son yıllarda hızla büyüyen bir ticaret hacmine sahip olmasıyla dikkat çekmektedir. Ekonomide önemli bir rol oynayan bu sektör, istihdam sağlaması ve ülke ekonomisine katkıda bulunmasıyla öne çıkmaktadır. Bu çalışmanın temel amacı, otomotiv yan sanayisinde faaliyet gösteren bir fabrikanın dağıtım kanallarını modellemek ve karşılaştığı problemlere matematiksel bir yaklaşımla çözüm getirmektir.

Çalışmanın amacı, tedarik zinciri milk-run süreçlerinde verimlilik sağlayacak stratejilerin belirlenebilmesi için araç rotalama problemi modeli oluşturmaktır. Bu kapsamda otomotiv sektöründe yer alan milk-run sürecindeki araç rotalama problemi, Tavlama Benzetimi Algoritması, Clarke-Wright Tasarruf Algoritması ve Genetik Algoritma çözüm yöntemleriyle çözümlenmiştir. Algoritma, Doğrusal Regresyon ile yapılan tedarikçi talep miktarlarındaki tahminleme ile desteklenmiştir. Algoritmalar çözümlenerek en iyi çözümü veren algoritma saptanmıştır.

Bu çalışma, tedarik zinciri yönetiminde araç rotalama problemlerine yeni bir bakış açısı getirmiştir. Geliştirilen uygulama sayesinde, araç rotalama problemi çözüm yöntemleri kullanılarak tedarik zinciri yönetimindeki maliyetler minimize edilerek verimlilik

artırılmıştır. Sonuç olarak, bu çalışma, klasik ve meta sezgisel çözüm yöntemleriyle birlikte doğrusal regresyonun, tedarik zinciri yönetimindeki araç rotalama problemlerinin çözümünde nasıl entegre edilebileceğini göstermektedir.

Bu çalışmanın organizasyonu aşağıdaki gibidir. Kuramsal Temeller ve Kaynak Araştırması Bölüm 2’de, Materyal ve Yöntem kısmı Bölüm 3’te, Bulgular kısmı Bölüm 4’te, Bölüm 5’te Tartışma ve Sonuç kısmı mevcuttur.



## 2. KURAMSAL TEMELLER ve KAYNAK ARAŞTIRMASI

Bu araştırma çerçevesinde, otomotiv yan sanayi fabrikası tedarik zinciri süreçlerinde yer alan araç rotalama problemini çözmek için gelişmiş çözüm yöntemleri kullanılmıştır. Araç rotalama problemleri (ARP), bir merkez noktasından başlayarak, farklı talep noktalarına dağıtım veya toplama rotalarının en kısa mesafeyle belirlenmesini amaçlar. ARP, bazen en kısa mesafeyi tespit etmeyi hedeflerken bazen de araç sayısını minimize etmeyi amaçlar. Bu nedenle, ARP'ne yönelik çeşitli yaklaşımlar ve çözümler geliştirilmiş olup, bu çalışmalar halen devam etmektedir. Literatürde, Araç Rotalama Problemleri üzerine birçok araştırma bulunmaktadır. Araç rotalama problemlerinin çözüm yöntemlerine dair yapılan çalışmalar aşağıda açıklanmaktadır.

Araç rotalama problemleri 1959 yılında ilk olarak Dantzig ve Ramser tarafından araştırılmıştır. Ardından Clarke-Wright (1964), bu araç rotalama metodunu geliştirmiştir. Bu geliştirilen problemler, maliyetleri, yol mesafelerini, araç kullanımını ve harcanan zamanı en aza indirmeyi amaçlarken, çeşitli kısıtları da ele alır. Bu kısıtlar; araçların ağırlık veya hacim bazında kapasite, toplam harcanan zaman, araçların çalışma saatlerine ilişkin zaman sınırlamaları, müşteri talepleri, dağıtım operasyonları ve rota seçimleri için özel kısıtlar ve kullanılan depo sayısı olarak özetlenir.

Laporte (1992), araştırmaları sonucunda araç rotalama problemine katkıda bulunmak amacıyla kesin ve sezgisel olmak üzere iki farklı türde model geliştirmiştir. Bu çalışmayla, araç rotalama probleminin hem doğrudan çözümüne yönelik kesin matematiksel modeller geliştirmiş hem de daha hızlı ve pratik çözümler sunan sezgisel modeller üzerine çalışarak bu alanda literatüre katkı sağlamıştır.

Eryavuz ve Gencer (2001), çalışmalarında dağıtım probleminin çözümünde bir araç rotalama problemi modeli geliştirmiştir. Problemin çözümünde klasik sezgisel çözüm metodlarından Clarke-Wright Tasarruf Algoritması kullanılmıştır. Bulunan çözümler 2-opt iyileştirme metodu kullanılarak geliştirilmeye çalışılmış ve elde edilen veriler karşılaştırılmıştır.

Düzakın ve Demircioğlu (2009), çalışmalarında eşit tip araçlarla kapasite kısıtlı araç rotalama problemi için Clarke-Wright Tasarruf Algoritması'nı uygulamıştır. Bu çalışmalar, Clarke-Wright Tasarruf Algoritması'nın çeşitli tedarik zinciri yönetimi senaryolarında etkili bir şekilde uygulanabileceğini göstermektedir. Bu çözümler, tedarik zinciri operasyonlarının optimize edilmesi açısından önemli bir rol oynamaktadır.

Kesen (2012), dağıtım kararlarının alınmasında araç rotalama probleminin incelenmesine odaklanarak matematiksel bir model geliştirilmiştir. Bu model, eş zamanlı olarak dağıtım problemlerini çözebilmektedir. Model bir örnek üzerinde uygulanarak problem çözümü sağlanmıştır.

Doulabi ve Seifi (2013), araştırmalarında araç kapasitesi sınırlı tek ve çok depolu araç rotalama probleminde Tavlama Benzetimi Algoritması ve Tamsayı Programlama ile çözüm bularak karşılaştırma yapmıştır.

Yazgan, Ercan ve Arslan (2014), çalışmalarında kapasite kısıtlı araç rotalama problemine odaklanmıştır. Yeni bir melez algoritma, Clarke ve Wright Tasarruf Algoritması ile en kısa yol yöntemini birleştirerek geliştirilmiştir. Sonuçlar, bu yeni melez yöntemin performansının daha iyi olduğunu göstermiştir.

Wei ve Zhou (2014), araştırmalarında Tavlama Benzetimi ve Genetik Algoritma'nın bileşimi ile tek depolu araç rotalama probleminde toplam maliyeti en küçüklemiştir.

Durak (2015), çalışmasında araçların milk-run sürecindeki yol süresini minimize eden ve kısıtları optimize eden bir model geliştirilmiştir.

Göçken, Yaktubay ve Kılıç (2018), araştırmalarında Zaman Pencere Araç Rotalama Problemi'ni (ZPARP) ele almıştır. Bu problemin meta-sezgisel çözüm yöntemlerinden Genetik Algoritma ile çözümü sağlanmıştır. En Yakın Komşu ve Süpürme Algoritması, genetik algoritmanın başlangıç popülasyonunu oluşturmak için kullanılmıştır.

Kılıç (2020), çalışmasında bir firmanın depolarından yapılan milk-run dağıtım sürecindeki en uygun rotaların belirlenmesi problemi çözülmüştür. Çalışmada Clarke-

Wright Tasarruf Algoritması ve 2-opt iyileştirme yöntemi uygulanmıştır. Sonuçlar Friedman testi kullanılarak karşılaştırılmıştır.

Meydan (2023), araştırmasında farklı ihtiyaçlara yönelik olan araç rotalama problemleri çözümü için klasik sezgisel ve meta sezgisel yaklaşımlar incelenmiştir. Clarke-Wright tasarruf algoritması kullanılarak iyileştirme yapılmıştır.

Yazgan, Ercan Cömert ve Kılıç (2020), bu çalışmada Clarke-Wright Tasarruf Algoritması ve Yerel Aramalı Sezgisel Algoritma kullanılarak kapasiteli araç rotalama problemi için uygun rotalama yapılmıştır. Elde edilen bulgular, gerçek verilerle karşılaştırılmış ve yapılan testler sonucunda yerel aramalı sezgisel algoritmanın diğerlerine göre üstün performans gösterdiği tespit edilmiştir.

Keser, Sarıçiçek ve Yazıcı (2020), çalışmalarında iç lojistik ile alakalı araç rotalaması için Tavlama Benzetimi Algoritması'nı önermiştir. Bu algoritma Tabu Arama Algoritması ile karşılaştırılmıştır. Tavlama Benzetimi Algoritması'nın daha iyi sonuçlar verdiği saptanmıştır.

Şahin, Karagül ve Aydemir (2021), araştırmalarında bir firmanın dağıtım problemi için Tavlama Benzetimi Algoritması kullanılarak iyileştirme yapılmıştır.

Emeksiz ve Demir (2021), çalışmalarında regresyon modellerini kullanarak tahminleme yapmıştır. En düşük hata oranını veren regresyon yöntemi saptanmıştır.

Şehitoğlu ve İşleyen (2023), çalışmalarında bir dağıtım modeline ait stokastik talepli model oluşturularak araç rotaları Tavlama Benzetimi Algoritması ile elde edilmiştir. Sonuçlar, algoritmadan elde edilen rotaların mevcut rotalardan daha iyi performans gösterdiğini belirtmiştir.

Sanlı ve Kartal (2024), çift aşamalı bir strateji önererek kapasiteli araçların rotalama problemini çözmüşlerdir. İlk adımda, düğümlerin hangi araçlara atanacağı makine öğrenimi algoritmaları kullanılarak belirlenir ve kapasite dengelemesiyle her aracın toplam talebi aşmaması sağlanır. İkinci aşamada, her araç depodan tur oluşturularak gezgin

satıcı problemi matematiksel modeli kullanılarak en kısa mesafeyi kat eder. Çalışmada K-En Yakın Komşuluk Algoritması (K-NN), K-Ortalamalar (K-means) ve Lojistik Regresyon (LR) algoritmaları kullanılmıştır. Tavsiye edilen yöntem, birbirinden farklı araç sayıları ve veri seti boyutları kullanılarak duyarlılık analizlerine tabi tutulmuştur.

Balcı, Işıklı Esener ve Kurban (2012), çalışmalarında en küçük kareler yöntemi ve regresyon analizi yöntemiyle kısa dönemli talep miktarında tahminleme yapmıştır. Çoklu regresyon modeli kurularak model oluşturulmuştur.

İncelenen literatür sonucu elde edilen sonuçlar, endüstrideki milk-run süreçlerinin ihtiyaçlarına yönelik araç rotalama problemlerinin üzerine yapılan çalışmaların sınırlı olduğunu göstermektedir. Literatürde genellikle klasik araç rotalama problemlerine daha fazla odaklanıldığı gözlemlenmektedir. Bu nedenle, milk-run süreçlerinde rotalama konusunda daha fazla araştırma ve geliştirme çalışmalarına ihtiyaç vardır. Milk-run süreçlerinin verimliliğini artırmak için daha etkili araç rotalama stratejileri geliştirmek amacıyla bu çalışma yapılmıştır.

Bu çalışma, otomotiv yan sanayi fabrikasındaki milk-run sürecindeki dağıtım rotalama problemini ele almaktadır. Clarke-Wright Tasarruf Algoritması gibi klasik Araç Rotalama Problemi çözüm yöntemlerinin yanı sıra Tavlama Benzetimi Algoritması ve Genetik Algoritma gibi meta-sezgisel çözüm yöntemleri incelenerek sürecin verimliliğini artırmak hedeflenmektedir. Bu yaklaşım, endüstrideki milk-run süreçlerinin verimliliğini artırmak ve işletme maliyetlerini azaltmak için önemli bir adım olarak değerlendirilmektedir.

Bu kapsamda, elde edilen bulgular, fabrikadan toplanan güncel veriler ile karşılaştırılmıştır. Çalışmanın, endüstride milk-run süreçlerinin etkin yönetimi için önemli bir adım olacağı ve literatüre önemli bir katkı sağlayacağı ve endüstriyel uygulamalara değerli bir katkı sunmayı amaçlamaktadır. Ayrıca ileriki araştırmalar için bu alanda yapılacak katkıların önemli olduğu vurgulanarak bu çalışmanın sadece bir başlangıç noktası olduğu belirtilmiştir.

### 3. MATERYAL VE YÖNTEM

Bu bölümde çalışmanın amacı ve yöntemi belirtilmiştir. Clarke-Wright Tasarruf Algoritması, Tavlama Benzetimi Algoritması ve Genetik Algoritma çözülmüştür. Doğrusal Regresyon ile tedarikçilerin talep miktarları ile ilgili tahminleme yapılmıştır. Bölüm 3.1.'de araç rotalama problemleri detaylı bir şekilde açıklanmıştır. Araç rotalama problemlerinin uygulama alanları, çeşitleri ve çözüm yolları hakkında detaylı bir şekilde bilgi sunulmuştur. Bölüm 3.2.'de Doğrusal Regresyon metodolojisi hakkında ayrıntılı bilgi sağlanmıştır. K-En Yakın Komşu Algoritması hakkında Bölüm 3.3'te bilgi verilmiştir. Bölüm 3.4.'de çalışmaya ait uygulamalar yer almaktadır. Problem çözümüne ait; problem tanımlanması, matematiksel model, problem çözümünde kullanılan metotlar açıklanmıştır.

Bu çalışmanın temel amacı otomotiv yan sanayinde faaliyet gösteren bir fabrikanın dağıtım kanallarını matematiksel modelleme yoluyla analiz etmek ve mevcut operasyonlarda karşılaşılan sorunları çözmek amacıyla verimliliği arttırmaktır. Bu amaçla, dağıtım süreçlerini daha etkin bir hale getirebilmek için araç rotalama problemi çözüm yöntemleri kullanılacaktır.

Dağıtım kanallarının modellenmesi, tedarik zinciri yönetimi perspektifinden ele alınacak ve mevcut operasyonlarda gözlemlenen verimlilik düşüklükleri ve kaynak israfı gibi sorunların çözümü için stratejik kararlar alınacaktır. Elde edilecek sonuçlar, fabrikanın dağıtım operasyonlarını iyileştirmek ve rekabet avantajı elde etmek için stratejik kararlar alınmasına yardımcı olacaktır. Bu çalışma, otomotiv sanayisinde faaliyet gösteren fabrikalar için bir model oluşturarak, operasyonel verimliliği artırmak ve rekabetçi bir avantaj sağlamak amacıyla önemli bir katkı sağlayacaktır.

Fabrikanın mevcut milk-run taşıma sistemiyle ilgili veriler tüm detayları ile toplandıktan sonra araç rotalama problemi için çözüm geliştirilecektir. Dağıtım rotasının belirlenmesi sürecinde, dağıtım sistemine ilişkin maliyetlerin en düşük seviyeye çekilmesi amaçlanmaktadır.

Uygulamadaki problemin karmaşıklığı ve tedarikçi sayısının fazlalığı nedeniyle, yöntemler ile manuel olarak çözümlene yapmaktır mümkün olmayacaktır. Bu sebeple problem çözümü Excel, Python ve Matlab desteğiyle sağlanacaktır.

### **3.1. Araç Rotalama Problemleri**

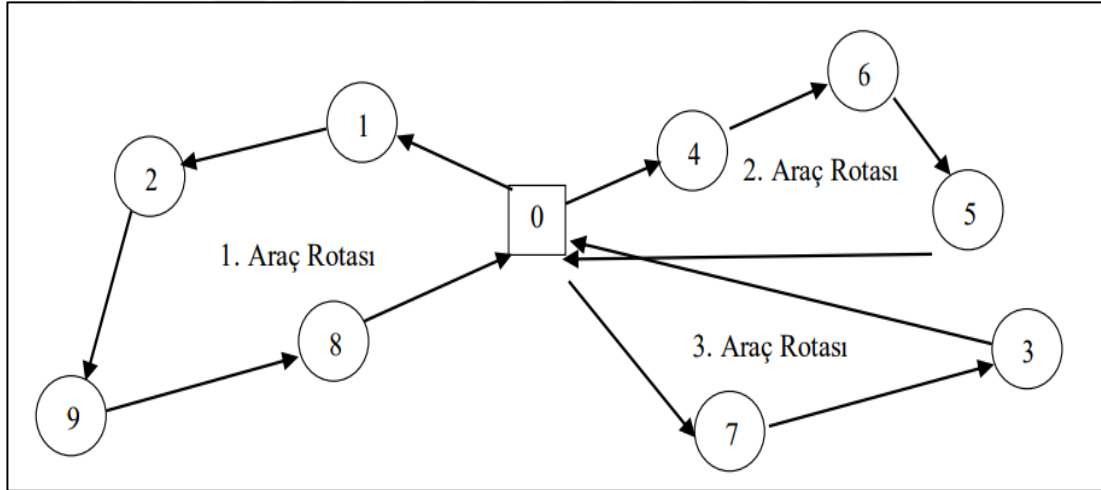
Araç Rotalama Problemleri(ARP), araçların en uygun dağıtım rotalarının belirlendiği problemlerdir. Bu algoritma ilk kez 1959 yılında Dantzig ve Ramser öncülüğünde ortaya konulmuştur (Dantzig ve Ramser, 1959). Çalışmalarında benzin istasyonlarına yakıt dağıtımına odaklanmışlardır. Bu araştırmada, depodan çıkıp benzin istasyonlarına giden araçlar için belirli şartları sağlayarak taşıma ve yol maliyetlerini en aza indirecek en uygun rotanın her bir araç için belirlenmesi hedeflenmiştir. Clarke-Wright, 1964 yılında bu teoriye katkı sağlayarak bu probleme daha sezgisel olan bir yaklaşım önermiştir.

Klasik Araç Rotalama Problemi (KARP), homojen araçlara sahip olduğu için araçların kapasiteleri ve türleri aynıdır. Bu durum, araçların belirlenen noktalara doğru hareket etmeden önce başlangıç noktası olarak belirlenen dağıtım deposundan hareket ettiğini gösterir. Araçlar, müşterilere hizmetlerini sağladıktan sonra yeniden merkeze dönmek zorunda olup belirli bir taşıma kapasitesine sahiptir. Klasik ARP'de müşteri açısından bakıldığında, her bir müşterinin tek bir rota üzerinden hizmet alması gerekmektedir ve bu rotayı izleyen mevcut araçlardan sadece biri olmalıdır. Bu kısıtlar, toplam ulaşım maliyetlerini en aza indirecek rotaların bulunması amacıyla hizmet etmektedir. Bundan dolayı, araç rotalama probleminin tedarik zinciri ve lojistik yönetiminde büyük bir değeri bulunmaktadır. Araç rotalama probleminin etkin bir şekilde çözülmesi, lojistik maliyetlerin minimize edilmesi, teslimat sürelerinin optimize edilmesi ve müşteri memnuniyetinin artırılması gibi önemli hedeflerin gerçekleştirilmesine katkıda bulunmaktadır.

Taşıma maliyetlerinin azaltılması ve müşteri memnuniyetinin artırılması, işletmeler için hayati bir öneme sahiptir. Bu hedeflere ulaşmak için, araç rotalarının mesafesi ve tur sürelerinin kısaltılması, birçok işletmenin öncelikli hedeflerinden biridir. Bunun nedeni, ulaştırma maliyetlerinin genellikle firmaların en büyük harcamalarından biri olmasıdır. Bu bağlamda, mevcut işgücü ve malzeme kaynaklarının etkin bir şekilde kullanılması

giderek daha önemli hale gelmektedir. Araç rotalama probleminde, başlangıç noktasından çıkan araçlar, çeşitli konumlarda bulunan müşterilere hizmet etmeye çalışır. Bu süreçte, toplam zamanı ve toplam yol mesafesini minimize eden ve maliyeti en aza indiren rotaları belirlemek amaçlanmaktadır. Araç rotalama problemleri için genel kısıtlar aşağıdaki şekildedir:

- Tüm müşterilere, taleplerine uygun hizmet sunulmalıdır.
- Her varış noktasına bir kez ziyaret edilmeli ve bu noktaya sadece bir araç tarafından hizmet edilmelidir.
- Rota, depodan başlamalı ve depoda son bulmalıdır.
- Rotadaki müşteri taleplerinin toplamı, aracın taşıma kapasitesini aşmamalıdır.
- Her araç sadece tek bir rota üzerinde görev yapmalıdır.



**Şekil 3. 1.** Araç rotalama problemine ait şematik gösterim

Araç rotalama problemlerinde, çeşitli amaç fonksiyonları üzerinde iyileştirme çalışmaları yapılmaktadır. Bu amaçlar genellikle şu şekildedir:

- Taşıma masraflarını ve araçların kullanımı ile ilgili sabit maliyetleri en aza indirmek.
- Araç ve/veya sürücü sayısını en aza indirmek.
- Hizmet verilemeyen müşteriler için gereken ceza maliyetlerini en aza indirmek.

- Noktalar arası toplam mesafeyi en aza indirmek.
- Toplam süreyi en aza indirmek.

Tek depolu bir klasik araç rotalama probleminin formülasyonu aşağıdaki gibi örneklendirilebilir:

$m$ : araç sayısı

$n$ : müşteri sayısı

$d_{ij}$ :  $i$  ve  $j$  noktaları arasındaki mesafe

$q_i$ :  $i$ . müşterinin talep miktarı

$X_{ijk}$ :  $\begin{cases} 1, \text{ eğer } k. \text{ araç } i \text{ noktasından } j \text{ noktasına doğru hareket ediyorsa} \\ 0, \text{ aksi durumda} \end{cases}$

$$\text{Min } Z = \sum_{i=0}^n \sum_{j=0, j \neq i}^n \sum_{k=1}^m d_{ij} x_{ijk} \quad (\text{Amaç fonksiyonu}) \quad (3.1)$$

$$i=0 \text{ için } \sum_{k=1}^m \sum_{j=1}^n x_{ijk} = m \quad (3.2)$$

$$i \in \{1, \dots, n\} \text{ için } \sum_{k=1}^m \sum_{j=0, j \neq i}^n x_{ijk} = 1 \quad (3.3)$$

$$j \in \{1, \dots, n\} \text{ için } \sum_{k=1}^m \sum_{i=0, i \neq j}^n x_{ijk} = 1 \quad (3.4)$$

$$k \in \{1, \dots, m\} \text{ için } \sum_{i=1}^n x_{i0k} \leq 1 \quad (3.5)$$

$$k \in \{1, \dots, m\} \text{ için } \sum_{i=1}^n q_i \sum_{j=0, j \neq i}^n x_{ijk} \leq C \quad (3.6)$$

Amaç fonksiyonu (3.1) numaralı denklem, minimize edilmesi gereken toplam yol mesafesini yani maliyeti minimize etmeyi hedeflemektedir. (3.2) numaralı kısıt denklemi ilgili noktadan hareket edecek olan araç sayısının  $M$  adet olacağını ifade etmektedir. (3.3) numaralı kısıt denklemi, her bir müşterinin en az bir araç tarafından hizmet alması gerektiğini ifade etmektedir. (3.4) numaralı kısıt denklemi, müşteriye giden ve müşteriden ayrılan yolların yalnızca birinin kullanılmasının gerektiğini vurgulamaktadır. (3.5) numaralı kısıt denklemi ise bir aracın sadece bir defa kullanılabileceğini ifade etmektedir. (3.6) numaralı denklem, araçların taşıma kapasitelerini nitelendiren  $C$  değerine eşit ya da daha az yükleme yapılması gerekliliğini ifade etmektedir.

### 3.1.2. Araç Rotalama Problemlerinin Uygulama Alanları

Mal ve hizmetlerin belirli noktalar arasında dağıtımıyla ilgili olan araç rotalama

problemleri, genellikle bir ağ içinde ortaya çıkar. Günümüz dünyasında, ürün dağıtım ve mal/hizmet taşıma zorlukları giderek artmaktadır. Bu bağlamda karşılaşılan bazı sorunlar şunlardır:

- Stok yönetimi ve ürünlerin satış noktalarına düzenli ve verimli bir şekilde ulaştırılması,
- Ana depodan mağazalara ürün sevkiyatı,
- Havayolu şirketleriyle yolcu ve kargo taşımacılığı,
- Para dağıtım, mamul temin ve dağıtım,
- Akaryakıt dağıtım,
- Ürün ve hizmetlerin birden fazla depodan farklı müşteri noktalarına ulaştırılması,
- Hammadde, yarı mamul ve mamullerin ilgili birimler arasında sağlıklı bir şekilde taşınması,
- Stok yönetimi ve ürünlerin satış noktalarına düzenli ve verimli bir şekilde ulaştırılması,
- Online alışverişlerin hızlı ve güvenli bir şekilde teslim edilmesi,
- Posta ve kurye hizmetleri gibi lojistik ve dağıtım problemleri gibi günlük yaşamda oldukça sık karşılaşılan problemler yer almaktadır.

### **3.1.3. Araç Rotalama Problemleri Türleri**

Farklı kısıtların neden olduğu gerçek dünya senaryoları, çeşitli özel durumlardan kaynaklanan araç rotalama problemlerini çeşitli alt kategorilere ayırmaktadır. Örneğin, belirli teslimat saatleri, araç kapasite sınırlamaları, öncelikli teslimatlar gibi faktörler, bu problemlerin çeşitli alt dallarını oluşturmaktadır. Bu problemlerin, her biri farklı zorluklar ve çözüm stratejileri gerektiren geniş bir alanı kapsamaktadır. Bu nedenle, gerçek dünyadaki özel durumlardan kaynaklanan araç rotalama problemleri, araştırma ve uygulama alanlarında geniş bir boyutta incelenmiştir.

#### **3.1.3.1. Klasik Araç Rotalama Problemi**

Genellikle lojistik ve dağıtım alanında en yoğun çalışılan problemlerden biridir. Bu problemde, rotalar genellikle tek bir depodan başlayarak tüm araçların aynı depoya

gelmesi ile depoda sonlanır. Homojen filo ya sahip araçlar, yani aynı kapasiteye sahip araçlar kullanılır. Müşteri talepleri deterministik olarak belirlenir. KARP'nin temel kavramı, her şehirde sadece bir aracın ziyaret etmesi ve bu aracın o şehirdeki tüm talepleri karşılaması gerekliliğidir. Klasik araç rotalama problemi aşağıda özetlenmektedir:

$G = \{V, A\}$ : serim

$V = \{0, \dots, n\}$ : düğüm kümesi

$A$ : ayrıt kümesi

Depo, referans olarak 0 noktasıyla belirtilir. Diğer durumlarda, depo  $n + 1$  düğümü olarak tanımlanmaktadır. Geriye kalan düğüm noktaları ise müşteri noktalarını temsil etmektedir.

### Parametreler

$c_{ij} = i$  şehrinden  $j$  şehrine gitme maliyeti

$d_{ij} = i$  ve  $j$  şehri arasındaki mesafe

$c_{ii} = 0$  olarak kabul edilmektedir.

### Karar Değişkenleri

$X_{ij} = \begin{cases} 1, & \text{eğer } i \text{ şehrinden } j \text{ şehrine gidiliyorsa} \\ 0, & \text{aksi durumda} \end{cases}$

Eğer  $G$  yönünde bir hareket olursa,  $C$  maliyet matrisi asimetrik olur, yani  $c_{ij} \neq c_{ji}$ . Bu noktada oluşan problem, Asimetrik Klasik Araç Rotalama Problemi (AKARP) olarak adlandırılmaktadır. Simetrik Klasik Araç Rotalama Problemi (SKARP) ise  $c_{ij} = c_{ji}$  durumunda oluşan problemdir. Laporte (1992) tarafından geliştirilen yapıda,  $E$  kümesi yönü olmayan kenar kümesini temsil eder ve  $A$  kümesinin rolünü üstlenir.

$$\text{Min } z = \sum_{i \neq j} d_{ij} x_{ij} \quad (3.7)$$

### Kısıtlar:

$$\sum_j x_{ij} = 1, \quad \forall i \in V \quad (3.8)$$

$$\sum_{i,j \in S} x_{ij} \geq S - v(S), \quad \{S: S \in V \setminus \{1\}, S \geq 2\} \quad (3.9)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in E; i \neq j \quad (3.10)$$

$$\sum_{i=1}^N d_{ij} \sum_{j=0, j \neq i}^N x_{ijk} \leq C \quad (3.11)$$

(3.7) numaralı amaç fonksiyonu, toplam mesafeyi en aza indirmeyi hedeflerken, (3.8) numaralı kısıta göre, bir şehirden mutlaka başka bir şehre gidilmesi gerektiği ifade edilmektedir. Kısıt (3.9)'a göre ise, her şehre kesinlikle bir geliş kaynağı bulunmalıdır. (3.10) numaralı kısıt engelleme kısıtıdır yani belirli şehirler arasındaki doğrudan bağlantıyı sınırlamaktadır. (3.11) numaralı kısıt denklemi araç kapasite kısıtı olan C değerine eşit ya da bu değerden daha az yükleme yapılması gerekliliğini belirtir.

### 3.1.3.2. Zaman Pencereci Araç Rotalama Problemi

Zaman Pencereci Araç Rotalama Araç Problemi (ZPRAP), belirlenmiş olan zaman diliminde her müşterinin ihtiyaç duyduğu bir dağıtım problemini ifade eder. Bu durumda, her bir müşteriye, belirlenmiş olan zaman diliminde dağıtım aracının hizmet vermesi gerekmektedir. Bu tür araç rotalama problemlerinde, araçların teslimatları, müşterilerin talep ettiği zaman aralıklarına göre planlanmaktadır. Bu sayede, müşteri memnuniyeti artırılarak dağıtım süreçleri daha verimli hale getirilebilmektedir. ZPARP, bir depodan coğrafi olarak yayılmış depolara en düşük maliyetli rotaların planlandığı bir problemdir. Rotalar aşağıdaki gibi planlanmalıdır:

- Her bir nokta, belirlenen zaman diliminde bir araç tarafından sadece bir kez uğranmalıdır.
- Rotalar, depoda başlayıp orada son bulmalıdır.
- Her bir rota üzerindeki noktaların toplam talebi, o rotaya tahsis edilmiş aracın kapasitesini aşmamalıdır.

Bu problemlerin amacı, sadece gerekli araç sayısının azaltılması değil, ayrıca tüm araçların toplam yol süresi ve toplam yol mesafesinin de en aza indirilmesidir. Bu nedenle, ZPARP çok amaçlı bir optimizasyon problemidir (Braysyy, Gendreau, 2001).

ZPARP,  $(N, A)$  yol ağı içerisinde belirlenir.

$N$ : düğüm kümesi

$C$ : müşteri kümesi

$n$ : müşteri sayısı

$c_{ij}$ : maliyet

$t_{ij}$ : süre

$V$ : özdeş araçlar kümesi

$Q$ : araç kapasitesi

$d_i$ : talep

$[a_i, b_i]$ :  $i$ . müşterinin zaman penceresi

$a_i$ :  $i$ . müşterinin en erken servis zamanı

$b_i$ :  $i$ . müşterinin en geç servis zamanı

$$\text{Min } z = \sum_{k \in V} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \quad (3.12)$$

### Karar Değişkenleri

$$x_{ij}^k = \begin{cases} 1, & \text{eğer } k \text{ aracı } i. \text{ müşteriden gider ise} \\ 0, & \text{aksi durumda} \end{cases}$$

$S_{ik}$ :  $k$  aracının  $i$ . müşteriye servis verdiği zaman

### Kısıtlar

$$\sum_{k \in V} \sum_{j \in N} x_{ij}^k = 1, \quad \forall i \in C \quad (3.13)$$

$$\sum_{i \in C} d_i \sum_{j \in N} x_{ij}^k \leq Q, \quad \forall k \in V \quad (3.14)$$

$$\sum_{j \in N} x_{0j}^k = 1, \quad \forall k \in V \quad (3.15)$$

$$\sum_{i \in N} x_{ih}^k - \sum_{j \in N} x_{hj}^k = 0, \quad \forall h \in C, \forall k \in V \quad (3.16)$$

$$\sum_{i \in N} x_{i,n+1}^k = 1, \quad \forall k \in V \quad (3.17)$$

$$x_{ij}^k (S_i^k + t_{ij} - S_j^k) \leq 0, \quad \forall (i,j) \in A, \forall k \in V \quad (3.18)$$

$$a_i \leq S_i^k \leq b_i, \quad \forall i \in N, \forall k \in V \quad (3.19)$$

$$x_{ij}^k \in \{0,1\} \quad \forall (i,j) \in A, \forall k \in V \quad (3.20)$$

(3.12) numaralı amaç fonksiyonu, maliyetin minimize edilmesini ifade eder. (3.13) numaralı kısıt denklemi müşterilerin sadece bir araca atanmasını sağlar. (3.14) numaralı kısıt denklemi kapasite kısıtıdır. Hiçbir araç, kapasitesinin izin verdiği kadar fazla müşteriye hizmet edemez. (3.15) numaralı kısıt, her bir aracın depodan bir defa çıkmasını zorunlu kılar. (3.16) numaralı kısıt aracın  $h$  müşterisine hizmet vermek üzere ayrılmasını sağlar ve başka müşterilere gitmesine izin vermez. (3.17) numaralı kısıt denklemi her bir aracın merkeze dönmesini sağlamaktadır. (3.18) numaralı kısıt denklemi, eğer  $k$  aracı  $i$  düğümünden  $j$  düğümüne seyahat ediyorsa,  $k$  aracının  $j$  düğümüne  $S_i^k + t_{ij}$  süresi içerisinde varacağını belirtir. (3.19) numaralı kısıt zaman pencerelerine uyulmasını sağlar. (3.20) numaralı kısıt, karar değişkeninin ikili olma şartıdır.

### **3.1.3.3. Heterojen Filolu Araç Rotalama Problemi**

Farklı tipte (yani farklı kapasitelerde veya farklı özelliklere sahip) araçların bir dağıtım veya hizmet rotası üzerinde etkin bir şekilde nasıl kullanılacağını belirleyen optimizasyon problemlerine, Heterojen Filolu Araç Rotalama Problemi (HFARP) denir. Bu problemde, filoya ait araçlar homojen değildir, yani aynı kapasiteye veya özelliklere sahip değildir.

HFARP'nin amacı, verilen bir dizi müşteriye veya hedefe en az maliyetle ve en uygun rotaları kullanarak hizmet veya mal dağıtımını gerçekleştirmektir. Bu problemde, farklı türdeki araçların farklı maliyetleri, kapasiteleri veya diğer özellikleri göz önünde bulundurularak, optimal bir rota planı oluşturulmaya çalışılır.

Bu tür bir problem ile, lojistik, taşımacılık, dağıtım ve hizmet endüstrilerinde sıkça karşılaşılır ve etkin bir çözüm, işletmelerin operasyonel maliyetlerini düşürmelerine ve kaynaklarını verimli bir şekilde kullanmalarına yardımcı olabilmektedir.

### **3.1.3.4. Talebi Belirsiz Araç Rotalama Problemi**

Bu tür problemler, talep miktarının belirli olmadığı durumlarda ortaya çıkan bir araç rotalama problemi olarak adlandırılmakta olup, dağıtım aracı müşteriye ulaştığında talebin netleştirilmesi gerekmektedir. Bu durumda, araç rotalaması planlanırken belirsiz talep durumu dikkate alınmalıdır ve rotalar esneklik sağlayacak şekilde tasarlanmalıdır.

Bu, talep belirlenene kadar araçların en etkin şekilde yönlendirilmesi sağlanarak operasyonel verimlilik artırılabilir.

### 3.1.3.5. Bölünmüş Talebe Sahip Araç Rotalama Problemi

Bölünmüş Talebe Sahip Araç Rotalama Problemi (BTARP), bir noktaya hizmet sağlayabilecek birden fazla aracın bulunduğu bir dağıtım problemidir. Bu problemde, tüm araçlar belirlenmiş bir zaman aralığında müşterilere hizmet sunmak için kullanılır.

### 3.1.3.6. Mesafe Kısıtlı Araç Rotalama Problemi

Mesafe Kısıtlı Araç Rotalama Problemi (MKARP)'inde, her bir aracın rotalarındaki maksimum mesafeyi aşmaması gerekmektedir. Bu kısıt, aracın kapasite kısıtıyla beraber kullanılabilmesi gibi, ayrı olarak da kullanılabilir.

$K$  aracın kullanıldığı araç rotalama problemlerinde, maksimum tur uzunlukları  $L_k$  ile gösterilmektedir. ( $k = 1, \dots, K$ ).  $V$  ve  $K$  müşteri ve araç setini ifade eder.  $x_{ij}^k$  ( $0 - 1$ ) değişkeni ise  $k$ . Aracın kullanıldığı  $(i, j)$  arki ifade eder. Bu kısıtın matematiksel denklemi (3.21) numaralı denklem ile ifade edilmektedir.

$$\sum_{i,j \in V} c_{ij} x_{ij}^k \leq L_k, k \in K \quad (3.21)$$

### 3.1.3.7. Çoklu Depoya Sahip Araç Rotalama Problemi

Müşterilerin yoğunlaştığı bölgelerde ve dağıtım firmalarının birden fazla merkeze sahip olduğu senaryolarda, dağıtım problemi karmaşık hale gelir. Bu durumda, farklı depoların çeşitli müşterilere hizmet vermesi gerektiği şeklinde bir modelleme yaklaşımı benimsenmektedir. Müşterilerin ve depoların konumları birbirleriyle karmaşık bir şekilde entegre olduğunda, yani depoların ve müşterilerin coğrafi dağılımı karmaşık bir yapı oluşturduğunda, çoklu depoya sahip araç rotalama problemi ele alınmalıdır. Bu model, depolar arasındaki mal akışının sağlanmasını ve müşteri taleplerini en etkili şekilde karşılamayı amaçlar.

Ayrıca, depoların konumları, stok seviyeleri, taşıma maliyetleri ve teslimat süreleri gibi faktörler de dikkate alınarak optimizasyon yapılmaktadır. Bu durumda, araçlar belirlenen depolara atanmaktadır. Atanan tüm araçlar, kendi deposundan yola çıkarak müşterilere hizmet etmeye devam eder. Daha sonra başlangıç noktasına dönüş sağlar. Bu şekilde, müşteri ihtiyaçlarına en uygun şekilde yanıt verebilmek için etkin bir dağıtım planı oluşturulur.

### **3.1.3.8. Geri Toplamalı Araç Rotalama Problemi**

Müşterilerin ürünlerle ilgili belirli parçaları, örneğin depozito, ambalaj ve palet gibi, iade edebileceği bir durumu ifade eden problemler Geri Toplamalı Araç Rotalama Problemi (GTARP) olarak nitelendirilmektedir. Bu durumda, müşterilerin iade edecekleri parçalar göz önünde bulundurularak araçların kapasiteleri yeniden hesaplanmalıdır.

### **3.1.3.9. Eş Zamanlı Topla Dağıt Araç Rotalama Problemi**

Tüm müşterilere ait iki önemli parametre bulunmaktadır:  $d_i$  ve  $p_i$ .  $d_i$  dağıtılacak ürünlerin miktarını belirtirken,  $p_i$  toplama yapılacak olan ürünlerin miktarını ifade eder. Bu ürünler genellikle homojen olma özelliği gösterir. Bazı örneklerde, talep miktarı tek bir tane olarak,  $d_i = d_i - p_i$  formülü ile ifade edilebilir. Müşterilerin lokasyonu için, dağıtım işleminin toplama işleminden daha önce gerçekleşmesi gerekmektedir.

### **3.1.3.10. Dinamik Araç Rotalama Problemi**

Çeşitli unsurları içeren bir süreçte meydana gelen değişiklikleri içeren problemler Dinamik Araç Rotalama Problemi (DARP) olarak adlandırılır. Bu değişiklikler, müşterilere ait talep miktarlarındaki değişiklikler, yeni oluşan talep noktalarının ortaya çıkması ve aracın seyahat edeceği yolun durumundaki değişiklikleri içerir.

Bu problem, lojistik operasyonlarda sürekli olarak güncellenen bilgilere dayalı bir yapıdadır. Dinamik araç rotalama stratejileri, operasyonel etkinliği artırmak ve teslimat sürelerini minimize edebilmek için önemlidir.

### **3.1.4. Araç Rotalama Problemleri Çözüm Yöntemleri**

Günümüze kadar süregelen zamanda birçok algoritma ortaya konulup bunların çözümleri için birbirinden farklı metotlar geliştirilmiştir. Araç rotalama problemleri için çözüm yöntemleri üç ana kategoride incelenmektedir: kesin çözüm yöntemleri, klasik sezgisel çözüm yöntemleri ve meta sezgisel çözüm yöntemleri. Optimal çözümü garanti eden kesin çözüm yöntemleri ve optimum çözüme yakın çözümleri daha kısa sürede bulan diğer çözüm metotları bulunmaktadır.

#### **Kesin Çözüm Yöntemleri**

- Dal ve Sınır Algoritması
- Dal ve Kesme Algoritması
- Dinamik Programlama
- Kesme Düzlemi Algoritması

#### **Klasik Sezgisel Çözüm Yöntemleri**

- Dantzig ve Ramser Yöntemi
- Clarke-Wright Tasarruf Algoritması
- En Yakın Komşu Algoritması
- Fisher ve Jaikumar Algoritması
- Süpürme Algoritması
- 2-Opt İyileştirme Operasyonu

#### **Meta Sezgisel Çözüm Yöntemleri**

- Tabu Arama Algoritması
- Karınca Kolonisi Algoritması
- Parçacık Sürü Optimizasyonu
- Yapay Arı Kolonisi Algoritması
- Genetik Algoritma
- Tavlama Benzetimi Algoritması

**Kesin Çözüm Yöntemleri:** Kesin çözüm yöntemleri, Araç Rotalama Problemi (ARP) için Gezgin Satıcı Problemleri (GSP) yöntemlerinin geliştirilmesi ile oluşturulmuştur (Düzakın ve Demircioğlu, 2009). Bu kesin çözüm yöntemleri, en iyi çözümü sağlar ancak çözüm süresi problem büyüklüğüne göre değişkenlik göstermektedir. Genellikle, küçük ve orta ölçekli problemler doğrulanabilir şekilde çözülebilir. Ancak, tüm problemler için kesin bir çözüm yöntemi bulunmamaktadır.

#### **3.1.4.1. Dal ve Sınır Algoritması**

Dal ve sınır ilkesine dayalı bir hesaplama tekniği, tam sayılı programlama problemlerini çözmek için kullanılan stratejidir. Karmaşık problemler, daha küçük alt problemlere bölünerek ele alınmaktadır. Bu bölümlenme aşamasında, tüm elverişli çözümler daha az boyutlu alt kümeler halinde düzenlenir. Dallandırmaya ait adım miktarını minimize edebilmek için sınırlandırma tekniğinden yararlanır. Dallandırma yöntemi ile alt problemlerin oluşturulmasının ardından, bu dalların çözümlerine ait alt ve üst sınırlar belirlenerek modelleme yapılır.

Alt problemlerin dallandırma ile oluşturulmasının ardından çözümlerinin alt ve üst sınırları belirlenerek modelleme yapılır. Sınırlandırma tamamlandığında, tüm alt problemler, en iyi çözümü içermeyecek şekilde sınırlandırıldığında algoritma sona erer. Eğer bir alt kümenin sınırları, en iyi çözümü içermeyecek şekilde belirlenmişse, bu alt küme çıkarılır. Problemin her aşaması, en iyi çözüm için bu yöntemle sistemli bir şekilde gözden geçirilmelidir. Problemin çözümünü en uygun alt sınır belirlemektedir.

#### **3.1.4.2. Dal ve Kesme Algoritması**

Dal ve Sınır algoritmasının bir alternatifi olarak ortaya çıkmıştır. Tam sayılı doğrusal programlama problemlerinde, öncelikle sınırların tam sayı olmasını sağlamak için kullanılır ve gerektiğinde bu sınırların esneklikle değiştirilmesine olanak tanınmaktadır. Sonrasında, problemin çözümü tespit edilir ve eğer ortaya çıkan çözüm tamsayı ise, işlem sonlandırılır.

### 3.1.4.3. Dinamik Programlama

Dal ve Sınır yaklaşımında oluşan alt problemler birbiri ile bağlantılı değildir; ancak, eğer alt problemler bağımlıysa problem çözümünde dinamik programlama yöntemi kullanılmaktadır. Dinamik programlama, her bir alt problemin çözümünün tek seferde çözülüp çözümlerinin bir tabloya kaydedilmesi prensibine dayanır. Aynı alt probleme tekrar rastlanıldığında, problemin tekrar çözülmesi yerine, tablodan bu problemin değeri okunur. Bu şekilde, gereksiz tekrar hesaplamalarının önüne geçilir.

### 3.1.4.4. Kesme Düzlemi Algoritması

Kesme düzlemi algoritmasında temel düşünce, optimal çözüm alanındaki bazı kısıtların kesilerek çıkarılmasıyla tam sayılı olan optimal çözüme ulaşabilmektir. Bu, doğrusal programlama ile tanımlanan uygun çözüm bölgesindeki parçaların simpleks yöntemiyle kesilerek çıkarılması esasına dayanır. Başka bir ifadeyle, tam sayılı sonuca ulaşmak için kısıtlar eklenir. Eğer eklenen kısıtlarla oluşturulan çözüm tam sayılı ise, bu durum optimal sonuca ulaşıldığı anlamına gelmektedir. Ancak, hala elde edilen çözüm kesirli ise tam sayılı bir çözüme erişmek için kısıtlar eklemeye ve çözüm tekrar hesaplanmaya devam edilir.

**Klasik Sezgisel Çözüm Yöntemleri:** Farklı rota kombinasyonlarının değerlendirilmesi, Araç Rotalama Problemlerinin çözümü için en uygun çözümün belirlenmesidir. Çözüme ait bütün kombinasyonların denenmesi, optimal çözüme ulaşmak için etkili olabilir; ancak ölçeği büyük olan problemlerin tüm olasılıklarının sınanması ve optimal çözümün bulunması pratik olmamaktadır. Bu yüzden, sonuçlara daha hızlı bir şekilde ulaşmak için sezgisel çözüm yöntemleri geliştirilmiştir.

Bu yaklaşımlar, mümkün olmayan atamalardan başlar ve her adımda bir kenar eklenerek mümkün olan çözüme doğru ilerler. Çözüm sistemine kenar eklenirken, araç kapasite kısıtlarının dikkate alındığından emin olunmaktadır. Eklenen kısımlar, bazı maliyet avantajlarına göre seçilmektedir. Bu yöntemler, araç rotaları içinde veya arasında kenarları ve köşeleri değiştirerek uygun çözümü geliştirmeye odaklanır. (Toth ve Vigo 2002). Genel olarak, bu yaklaşımlar 1959-1990 yılları arasında geliştirilmiştir.

### 3.1.4.5. Dantzig ve Ramser Yöntemi

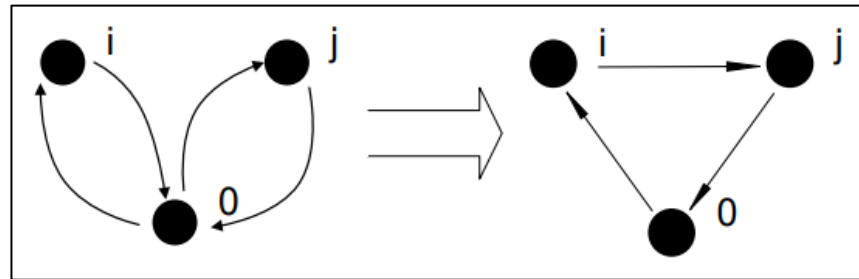
Araç Rotalama Problemi (ARP) ilk olarak 1959 yılında Dantzig ve Ramser öncülüğünde ortaya çıkarılmıştır (Dantzig, Ramser, 1959). Dantzig ve Ramser çalışmalarında, ilk defa, benzin istasyonlarına benzin dağıtımını üzerin odaklanarak, problemin çözümü için matematiksel bir model oluşturmuştur. (Atmaca, 2012).

Elde edilen modelde, belirli kısıtlar altında müşterilerin ihtiyaçlarını karşılayarak taşıma maliyetleri ve kat edilen yol mesafesini minimize eden araç rotaları elde edilmiştir. Elde edilen algoritmaya göre, araçların bir depodan hareket edip yine aynı depoya dönerek belirlenen noktalara hizmet etmesi sağlanmıştır. Algoritma gereğince, herhangi iki noktanın toplam talebi, araç kapasitesinin yarısını aştığı takdirde, bu iki noktanın birbiri ile bağlantısının kurulamayacağı kabul edilmektedir.

### 3.1.4.6. Clarke-Wright Tasarruf Algoritması

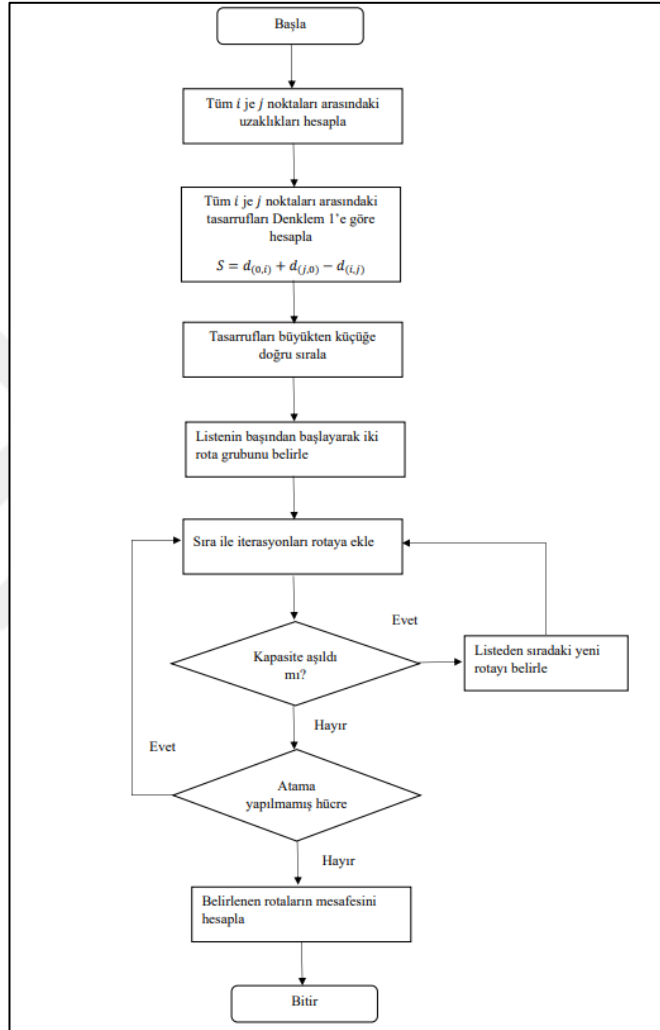
1964 yılında Clarke-Wright öncülüğünde geliştirilen algoritma, Dantzig ve Ramser tarafından yapılan çalışmalardan esinlenilerek tasarlanmıştır. En çok tercih edilen sezgisel yöntemlerden biridir. Eğer herhangi iki nokta birbirine bağlı değilse, araç her noktaya tek tek gidip gelmek zorundadır. Ancak noktalar birbirine bağlandığında, ilk olarak araç birinci noktaya gider, sonra ikinci noktaya geçer ve en son olarak başlangıç noktasına döner. Bu durum, noktalar arasındaki bağlantının varlığına bağlı olarak bir tasarruf oluşturur. Tasarrufun nasıl oluştuğu Şekil 3.2’de gösterilmiştir. Denklem 3.22’de algoritmaya ait formülasyon yer almaktadır.

$$S_{ij} = C_{i0} + C_{0j} - C_{ij} \quad (3.22)$$



Şekil 3. 2. Clarke-Wright tasarruf algoritması şematik gösterimi

Tüm nokta çiftleri için hesaplanan bu tasarruflar, ilk olarak büyükten küçüğe doğru sıralanır ve daha sonra yüksek tasarruf sağlayanlar, kapasite sınırlarını aşmadan birbirlerine ve depoya bağlanmaya çalışılır. Her iki yöntem de hesaplanır ve en iyi sonucu veren yöntem tercih edilerek çözüme devam edilir. Genellikle paralel tasarruf algoritması daha iyi sonuçlar verir. Şekil 3.3.'de algoritmaya ait akış diyagramı yer almaktadır.



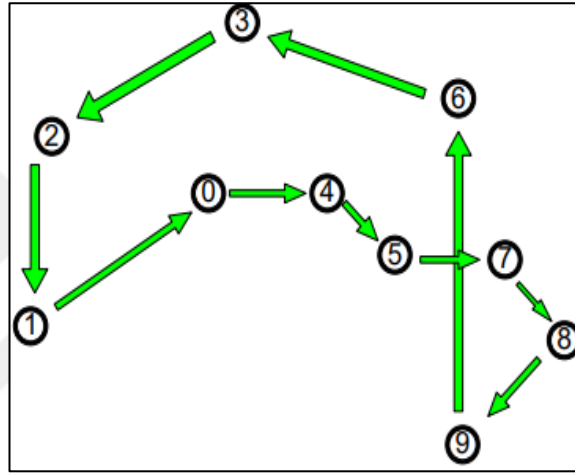
**Şekil 3. 3.** Clarke-Wright tasarruf algoritması akış şeması

Paralel tasarruf algoritmasında yeni bir rota oluşturulabilmesi için sıralamaya göre eklenemeyen kenarlar vardır. Clarke-Wright Tasarruf Algoritmasında ise, yeni bir araç rotası başlatılıp kapasite değeri aşıldığında, oluşturulan yeni araç rotasına eklenmek üzere bağlanmamış kenarlar dikkate alınır. Uygun kenarlar, kapasitenin tamamı kullanılarak rotaya eklenir, fakat aynı anda iki rota oluşturulmaz.

### 3.1.4.7. K-En Yakın Komşu Algoritması

Algoritma, (Bellmore ve Nemhauser, 1966) öncülüğünde modellenen, oldukça basit bir çözüm yöntemi olarak Gezgin Satıcı Probleminin çözümünde kullanılır. Bu yöntemde, başlangıç noktasından en yakın noktaya hareket edilir, ardından süregelen en kısa mesafede yer alan noktaya ulaşılır ve döngünün sonunda problemin başladığı noktaya geri dönülür.

Bu süreç, En Yakın Komşu algoritmasının bir örneğini oluşturur ve Şekil 3.4.'te algoritmanın şematik gösterimi bulunmaktadır.



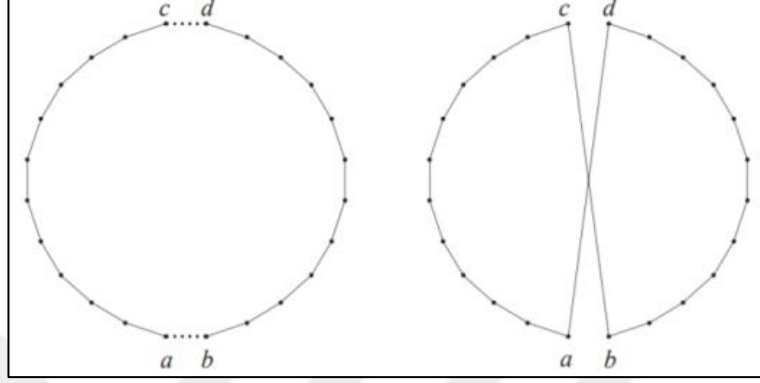
Şekil 3. 4. K-en yakın komşu algoritması şematik gösterimi

K-en Yakın Komşu Algoritması araç rotalama problemlerinin çözümü için adapte edildiğinde, rotaya en yakın komşunun dahil edilmesi mümkün olduğu müddetçe gerçekleşir. Bu durumda kapasite veya zaman kısıtlarını aşmaması gerekmektedir. Eğer kısıtları aşmadan eklemek için uygun bir nokta bulunamazsa, araç depoya dönerek yeni bir rota başlatılır.

### 3.1.4.8. 2-Opt İyileştirme Operasyonu

Croes (1958) tarafından geliştirilen iyileştirme operasyonu, herhangi bir elverişli çözüm alınabileceği gibi basit sezgisel çözüm yöntemleri ile de başlangıç çözümü bulunabilir. 2-opt iyileştirme operasyonu bir GSP turundaki iki kenarı kaldırır. Böylece tur iki parçaya bölünmüş olur. Daha sonra bu iki parçayı farklı kombinasyonlarla birleştirerek yeni bir

GSP turu oluşturur. Yeni tur, eski tura göre daha iyi ise, bu yeni tur çözüm olarak kabul edilir. 2-Opt algoritması, iyileştirme bulana kadar turu değiştirme ve yeniden oluşturma işlemini sürdürür (Johnson ve McGeoch, 1997). Modele ait şematik gösterim Şekil 3.5'te yer bulunmaktadır.



Şekil 3. 5. 2-opt iyileştirme operasyonu şematik gösterimi

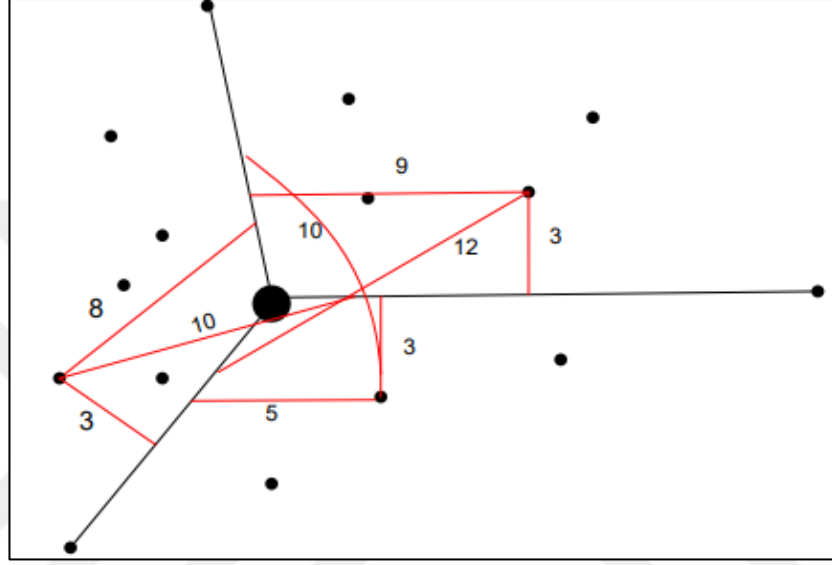
#### 3.1.4.9. Süpürme(Sweep) Algoritması

Süpürme (Sweep) Algoritması, Gillett ve Miller öncülüğünde 1974 yılında geliştirilmiştir. Önerilen algoritma ile büyük ölçekteki klasik araç rotalama problemleri çözülebilmektedir. Polar koordinatlar için noktalar  $i = 1, \dots, n$  için  $r_i, \theta_i$ , ve depo ise  $r_0 = 0$  ve  $\theta_0 = 0$  şeklinde belirtilmektedir. Koordinatlar  $\theta_i$ 'ye göre artan sıra ile düzenlenir. Rotalamada henüz kullanılmayan araç (k) belirlenerek açığı en düşük olan noktadan başlayarak, k aracının kapasitesi dolana kadar noktalar eklenir. Rota üzerindeki tüm noktalar tamamlanana kadar süreç sürdürülür. Bu aşamadan sonra araç rotaları, Gezgin Satıcı Problemi (GSP) çözüm metodlarından biri kullanılarak bulunmaktadır.

#### 3.1.4.10. Fisher ve Jaikumar Algoritması

1981 yılının başlarında, genelleştirilmiş atama metodu olarak çift aşamalı bir yaklaşım olan Fisher ve Jaikumar Algoritması geliştirmiştir. Gruplama işlemi için müşteriler arasındaki uzaklıklar hesaplanmaktadır. Problemden yer alan araç sayısı kadar araç rotası oluşturulacağı varsayılarak, araçların rotaları için bir çekirdek müşteri belirlenmektedir.

Müşteriler arasından seçim yapılırken, birbirlerine olan maksimum uzaklığa sahip olmaları dikkate alınır. Sonrasında, seçilen noktaların ilgili rotalara, yani depo ve nokta arasındaki doğrulara olan uzaklıkları hesaplanmaktadır. Kapasite sınırı göz önünde bulundurularak rotalara noktalar eklenir, çözüm tamamlanır. Şekil 3.6.' da yer alan örnek ile algoritmanın şematik gösterimi ifade edilmiştir. Son aşamada, belirlenen gruplar gezgin satıcı problemi sezgiselleri ile çözülmektedir.



Şekil 3. 6. Fisher ve Jaikumar algoritması şematik gösterimi

**Meta-sezgisel Çözüm Yöntemleri:** Son zamanlarda önemli bir ilerleme kaydetmiştir. Geleneksel sezgisel yöntemlerle karşılaştırıldıklarında, daha verimli bir şekilde en iyi çözümü aramak için uygun olmayan ve alt seviyede hareketlere izin veren bir yaklaşım sunar. Büyük ve karmaşık problemlerin çözümü için kesin çözüm yöntemlerinin gerektirdiği uzun süreyi önemli ölçüde azaltarak, bu tür problemlerin daha makul bir zaman diliminde çözülmesine olanak tanır. Pratik bir şekilde özellikle büyük ve karmaşık problemlerin çözülebilmesi için önemli bir araç olarak bilinmektedir.

Bu nedenle, pratikte önemli bir rol oynamaktadır. Ayrıca, makul bir sürede en iyi çözüme yakın sonuçlar elde edebilmeleri nedeniyle günümüzde yaygın olarak tercih edilmektedir. Bu yöntemler genellikle doğal süreçlerden ilham alarak geliştirilmiştir.

#### **3.1.4.11. Tabu Arama Algoritması**

Glover ve Laguna (1989), Tabu Arama yöntemini geliştirerek optimizasyon alanında önemli bir adım atmışlardır. Tabu Arama yöntemi, Araç Rotalama Problemi (ARP) gibi karmaşık optimizasyon problemlerinde iyi sonuçlar elde etmiştir. Yerel arama algoritmaları genellikle 2-opt kullanır ve sadece pozitif adımlara izin verdiği için yerel optimumdan öteye gidemezler. Ancak tabu arama algoritması, negatif yönlü adımlara izin vererek global optimuma ulaşabilir (Şahin ve Eroğlu, 2014; Nilsson, 2003).

Algoritmanın önceden denenmiş çözümlere dönmesini önleyen ve Tabu Arama yönteminin temelini oluşturan yasaklı liste iyileşen çözüm değerleri bulunduğu güncellenir. Ancak, oluşturulan bir sonuç, yasaklı listede bulursa bile, eğer bu çözüm daha önce elde edilen en iyi çözümden daha iyiyse, kabul edilir ve işlem yeni çözümler üzerinde devam eder. Genellikle, çözümler önce kötüleşir, sonra ise iyileşmeye başlar, bu da algoritmanın zamanla daha iyi sonuçlar elde etmesine olanak sağlar.

Tabu arama algoritması, kısa sürede optimum çözüme yakın bir çözüm bulma yeteneği nedeniyle araç rotalama problemlerinin çeşitli türleri için sıkça başvurulan yöntemlerden biridir.

#### **3.1.4.12. Genetik Algoritma**

Genetik algoritmalar, esasen doğal seçilime dayalı bir arama ve sayısal optimizasyon metodudur ve optimizasyon problemlerinin çözümünde evrimsel hesaplama tekniklerinin bir bileşeni olarak kullanılır. Genetik algoritmalar, biyolojik evrimin süreçlerini taklit ederek belirli doğal olaylarını modellemektedir. Bu yöntem, ilk kez 1975 yılında Holland tarafından geliştirilmiştir. Bu algoritma, bir başlangıç popülasyonundan yola çıkarak çaprazlama, mutasyon ve seçim gibi genetik operatörler kullanarak çözüm alanını araştırır ve en iyi çözümü elde etmeyi hedefler.

Genetik algoritmalar, problemlere çözüm üretmek için evrimsel süreçleri taklit eder. Bu nedenle, genetik algoritmaların pek çok kavramı, genetik bilimindeki karşılıklarıyla benzerlik taşır. Bu paralellik, algoritmanın başlangıç popülasyonu, gen, kromozom,

çaprazlama ve mutasyon gibi terimlerin kullanımında kendini göstermektedir. Genetik algoritmanın temel kavramları aşağıda açıklanmıştır.

**Gen:** Genler, genetik algoritma içindeki bir problemi temsil eden, potansiyel çözüm adaylarını ifade eden genetik yapı birimleridir.

**Kromozom (Birey):** Genetik algoritmalarda, genlerin birleşmesiyle kromozomlar, kromozomların ise birleşmesiyle popülasyon oluşur. Her bir kromozom, belirli bir problemin muhtemel çözümlerini temsil eder. Bu nedenle, kromozomların doğru ve etkili bir şekilde ifade edilmesi, problem çözümü için önemlidir.

**Popülasyon:** Bir dizi kromozomun toplamıdır. Başlangıçta rastgele oluşturulur ve her bir kromozom, potansiyel bir çözüm adayını temsil eder.

**Uygunluk Fonksiyonu ve Değeri:** Her bir kromozomun ne kadar iyi bir çözüm adayı olduğunu değerlendiren ve sıralayan bir fonksiyondur. Genellikle problemin optimize edilmesi gereken hedefine göre belirlenir.

**Seçilim:** Uygunluk değerlerine dayanarak popülasyondan en iyi kromozomları seçme sürecidir. Daha iyi çözümlere sahip kromozomlar daha yüksek seçilme olasılığına sahiptir.

**Çaprazlama:** İki veya daha fazla kromozomun genetik materyallerinin birleştirilmesiyle yeni kromozomlar üretilmesini sağlayan bir işlemdir. Bu işlem, genetik çeşitliliği artırarak daha iyi çözümlerin keşfedilmesini sağlar.

**Mutasyon:** Rastgele olarak seçilen kromozomların genetik materyallerindeki değişiklikleri tanımlayan ve genetik çeşitliliği artıran bir işlemdir. Mutasyon işlemi, yeni ve potansiyel olarak daha iyi çözümlerin keşfedilmesine olanak tanır.

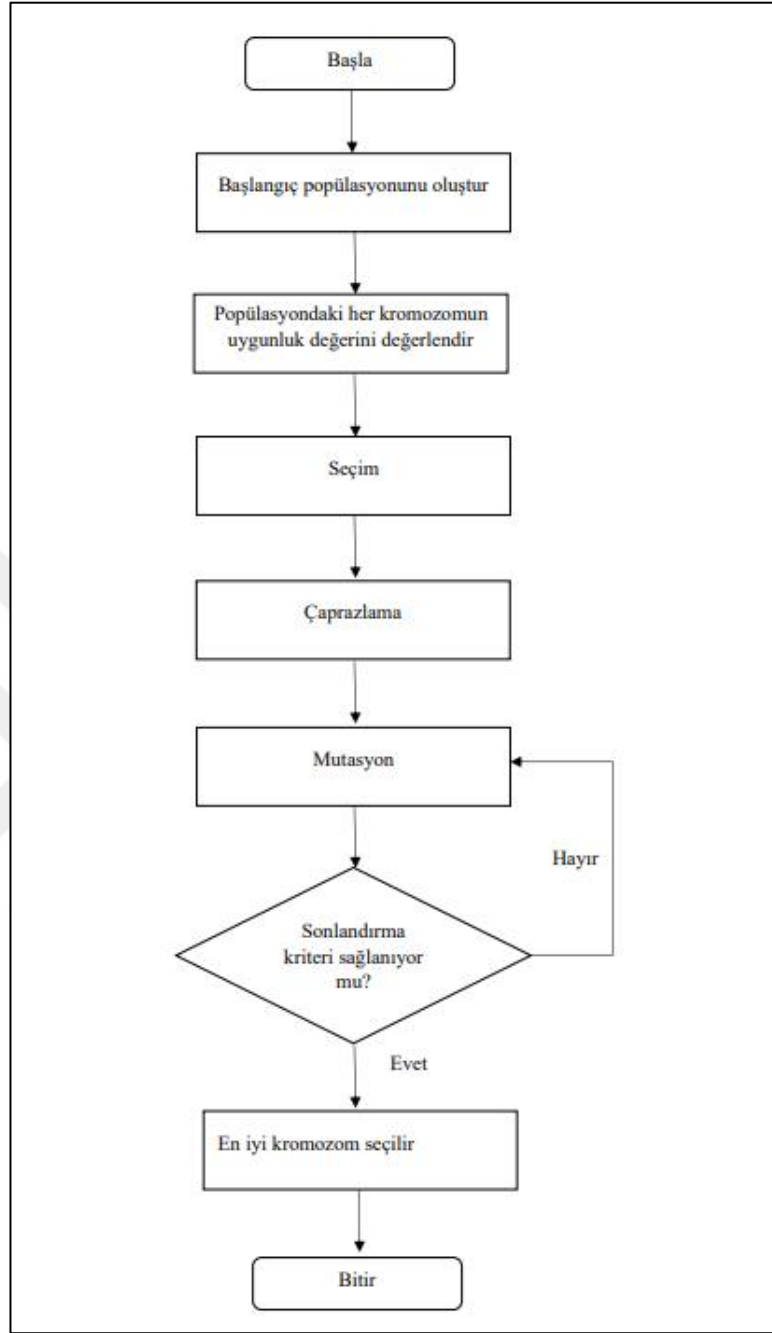
Algoritmanın işleyişi başlangıçta, genellikle basit sezgisel yöntemlerle elde edilen bir çözüm ile problemidir. Bu ilk çözüm başlangıç popülasyonu olarak adlandırılır. Bu başlangıç çözümü, her bir parametrenin bir gen olarak kabul edildiği bir yaklaşımla ele

alınır. Tüm bu genlerin bir araya gelmesiyle bir kromozom elde edilir. Kromozomlar, farklı biçimlerde temsil edilebilir.

Örneğin, ikili olarak (1-1-0-1-0), gerçek değerler olarak (3-4-6-2) veya sıralamalı olarak (2-3-1-4 veya 0-2-3-0-1-0) biçiminde gösterilebilirler. Başlangıçta probleme ait ilk kromozom ikilileri bu yöntemle oluşturulduktan sonra, çeşitli genetik operatörler kullanılarak yeni nesil kromozomlar üretilir. Bu operatörler arasında çaprazlama ve mutasyon bulunur.

Çaprazlama, mevcut iki kromozomun genlerinin birleştirilmesiyle yeni kromozomlar oluşturulmasını sağlar. Mutasyon ise genlerin değerlerinde küçük değişiklikler yaparak çözümün lokal minimumlardan kurtulmasını amaçlar. Yeni nesil kromozomlar üretildikten sonra, her biri bir uygunluk fonksiyonu ile değerlendirilir. Bu fonksiyon, çözümün kalitesini ölçen bir kriterdir. Daha iyi performans gösteren kromozomlar, bir sonraki adıma geçer. Son olarak, seçim adımında, en iyi kromozomlar belirlenir ve bir sonraki nesil için kullanılır. Bu şekilde, her iterasyonda daha iyi çözümler elde edilir ve en iyi çözüme yaklaşmak hedeflenir.

Şekil 3.7.'de genetik algoritma akış şeması yer almaktadır.



**Şekil 3. 7.** Genetik algoritma akış şeması

Genetik algoritmalar, bir optimizasyon probleminde optimal çözüme yakın sonuçlar sağlayabilir, ancak bunu kesin olarak sağlamaz. Başlangıçta çözüm hızlı bir biçimde ilerler, ancak çözüm optimale ulaştıkça bu gelişme azalır (üs şeklinde). Araç rotalama problemlerinin çözümünde Genetik Algoritma sıkça kullanılan bir çözüm yöntemidir.

Son dönemlerde, özellikle klasik sezgisel yöntemlerle birleştirilerek melez stratejiler geliştirilmiştir. Bu melez yaklaşımlar, genetik algoritmaların güçlü taraflarını diğer yöntemlerle birleştirerek daha etkili çözümler elde etmeyi amaçlar.

#### **3.1.4.13. Parçacık Sürü Optimizasyonu**

Parçacık Sürü Optimizasyonu algoritması Kennedy ve Eberhart (1995) öncülüğünde geliştirilmiş, kuş türlerinin tepkisel davranışlarından ilham alarak doğrusal olmayan problemlerin çözümü için kullanılan bir algoritmadır. Algoritmada olası optimal çözümler parçacık adı ile tanımlanmaktadır.

Başlangıçta belirlenen bir sayıda çözümlerle (parçacıkla) başlayan bu algoritma, her iterasyonda parçacıkları güncelleyerek mevcut en iyi çözümleri takip eder ve problem uzayında araştırma yaparak uygun çözümü arar. Parçacıklar, kuş sürülerinin uçuşlarını yönlendiren hız bilgisi benzeri bir bilgiyi kullanır. Her döngüde, parçacık konumları, iki en iyi parçacığa göre yenilenir. Birincisi; şimdiye kadar aynı numaralı parçacıklar arasında en yüksek uygunluk değerine sahip olan parçacıktır. Bu parçacık yerel en iyi ( $p_{best}$ ) olarak nitelendirilmiştir. Diğeri ise, popülasyonda o ana kadar tüm parçacıklar arasında elde edilen en iyi uygunluk değerini sağlayan parçacıktır. Bu parçacık global en iyidir ve ( $g_{best}$ ) ile gösterilir. Bulunan bu değerler hafızada saklanır.

#### **3.1.4.14. Yapay Arı Kolonisi Algoritması**

Karaboğa (2005) tarafından, doğadan esinlenilerek geliştirilen algoritma, arıların yiyecek ve arama davranışlarını modellemektedir. Bu algoritma, görevli ve görevsiz işçi arılar arasında bir ayırım yapar ve yiyecek kaynaklarından gelen olumlu ve olumsuz geri bildirimleri içerir. Belirli kaynaklardan yiyecek toplayan görevli işçi arılar bu kaynakların konumunu ve kalitesini diğer arılara dans alanında ileterek bilgi paylaşımında bulunur.

Kâşif ve gözcü olmak üzere görevi olmayan işçi arılar iki türe ayrılmaktadır. Bu arılar yeni yiyecek kaynaklarını aramaya devam eder. Kâşif türündeki arı, arı kolonisinde yer alan tüm arıların %5-10'undan meydana gelmektedir (Şahin ve Eroğlu, 2014).

### 3.1.4.15. Karınca Kolonisi Algoritması

1991 yılında Dorigo ve arkadaşları öncülüğünde geliştirilen Karınca Kolonisi Algoritması, karıncaların doğal davranışlarını matematiksel olarak modelleyerek oluşturulmuştur. Karıncalar, geçtikleri yollara feromon adı verilen kimyasal bir koku bırakarak izlerini sürer. Feromon izlerini takip ederek arta kalan diğer karıncalar da olası beslenme kaynaklarına ulaşmaktadır. Feromon zamanla buharlaştığı için feromon kokusu, karınca sayısının fazla olduğu yollarda daha etkili bir yoğunluktadır. Bazı karıncalar feromonun az olduğu yolları tercih edebiliyorken, genellikle feromon oranının daha yoğun olduğu yolların tercih edilme oranı daha yüksektir. Bu şekilde, uzun yolların kullanımı zamanla engellenir ve uygun yol karıncalar tarafından işaretlenir.

Algoritma, ilk olarak Gezgin Satıcı Probleminin çözümü için geliştirilmiş olsa da yapılan araştırmalar son yıllarda bu çözüm yöntemine kapasite, birden fazla dağıtım aracı gibi kısıtların eklenmesi ile araç rotalama problemlerinin başarı ile çözüldüğü gözlemlenmiştir.

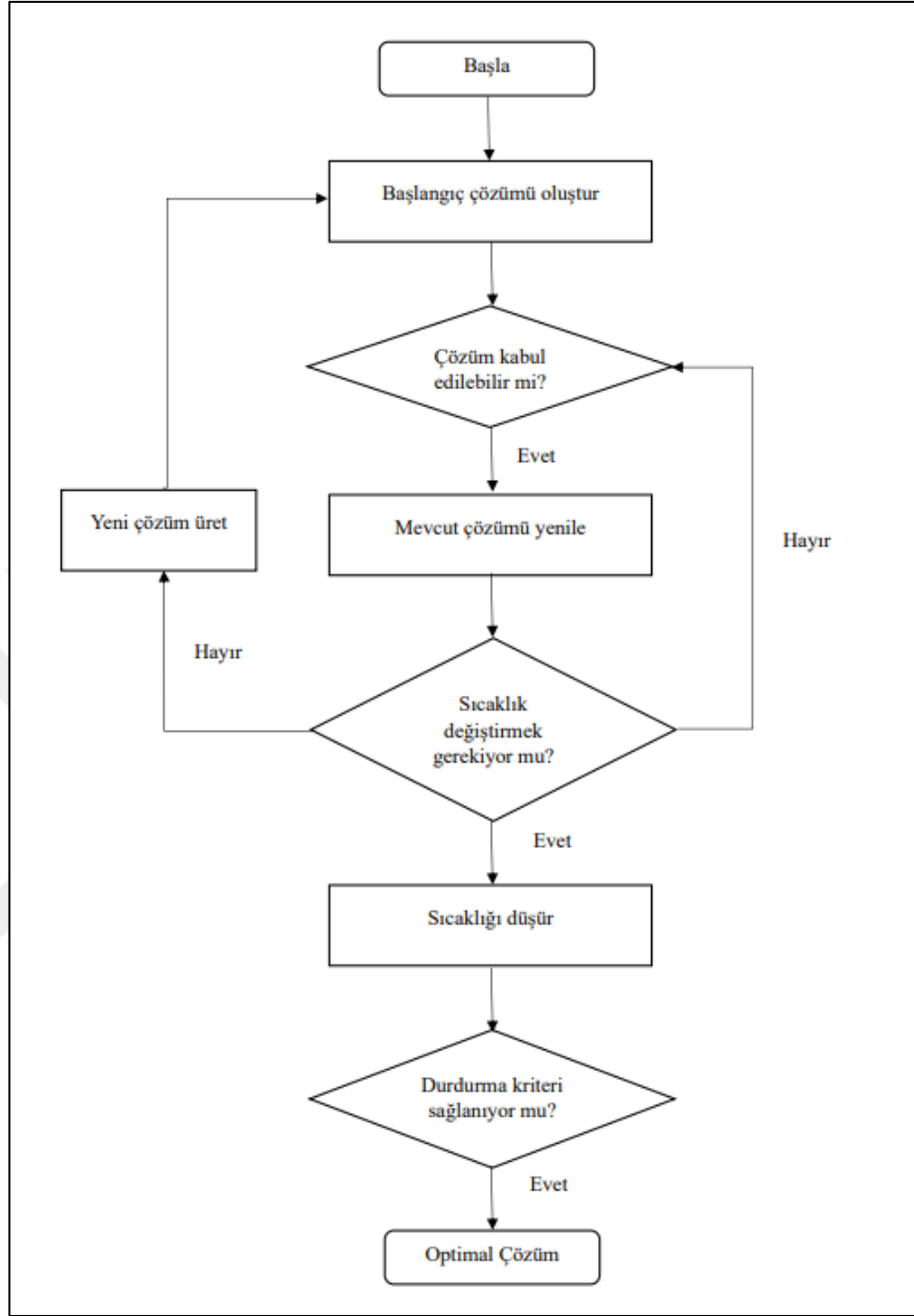
### 3.1.4.16. Tavlama Benzetimi Algoritması

Tavlama Benzetimi Algoritması, 1983 yılında Kirkpatrick ve arkadaşları öncülüğünde geliştirilen bir algoritmadır (Kirkpatrick ve ark., 1983). Temel mantık, olumlu ilerleme yerine negatif ilerlemenin seçilme olasılığının, tekrar sayısının artmasıyla giderek azalmasıdır. Bu sayede, başlangıçta uygun çözüm bölgeleri arasında atlamalar meydana gelirken, optimum çözüme yaklaşıırken bahsedilen olasılık değeri sıfıra yaklaşır ve uygun çözüm alanı azalır. Bu şekilde, global optimum çözüm yerine yerel optimumlardan çıkamama olasılığı giderek azalır.

(3.23) numaralı denklem olasılık kriteridir. Burada,  $\Delta$  (enerji) fonksiyonundaki değişim, T sıcaklık parametresi, k ise Boltzmann sabitini ifade etmektedir.

$$P = e^{\left(\frac{-\Delta}{kT}\right)} \quad (3.23)$$

Şekil 3.8'de Tavlama Benzetimi Algoritma'sı akış şeması yer almaktadır.



Şekil 3. 8. Tavlama benzetimi algoritması akış şeması

### 3.2. Doğrusal Regresyon

Doğrusal regresyon, bir özellik vektöründeki giriş değişkenlerinin ağırlıklarını kullanarak bir hedef değişkenini belirli bir hatayla tahmin etmeyi amaçlar. Doğrusal Regresyon varsayımı aşağıdaki (3.24) numaralı denklemle temsil edilmektedir.

$$Y = W_0 + W_1x + E_y \quad (3.24)$$

$W$  ağırlıkları,  $X$  verinin özellik vektörünü,  $E$  hatayı ve  $Y$  değişkeni ifade eder. Her veri örneği,  $E$  hatasına oranla bir doğru denklemine uzaklık gösterir. Çözüm sırasında hata değerleri çok küçük olduğu varsayımıyla ihmal edilir.  $W$  ağırlık vektörünün belirlenmesi,  $X$  ve  $Y$  değişkenlerine sahip veri kümesi kullanılarak doğrusal regresyon analizinin başlıca amacını oluşturur.

Doğrusal regresyon, istatistiksel bir analiz yöntemidir ve değişkenler arasındaki ilişkiyi anlamak, bu ilişkiyi modellemek ve tahminler yapmak amacıyla kullanılır. Genel olarak bağımlı değişkenin nasıl daha fazla bağımsız değişken tarafından etkilendiğini anlamak için kullanılır.

Örneğin, gelir seviyesinin ev büyüklüğü, eğitim seviyesi, yaş gibi faktörlere bağlı olarak nasıl değiştiğini anlamak için doğrusal regresyon kullanılabilir. Doğrusal regresyon, bağımlı değişkenin tahmin edilen değerinin bir doğrusal kombinasyonunu oluşturarak çalışır. Bu doğrusal kombinasyon, bağımsız değişkenlerin katsayıları ve bir sabitin (veya kat sayısız terim) toplamıdır. Bu model, bağımsız değişkenlerin bağımlı değişken üzerindeki etkisini ölçmek için kullanılır.

Doğrusal regresyon modelleri iki tür olarak ele alınmaktadır. Bu modeller; Basit Doğrusal regresyon ve Çoklu Doğrusal regresyondur.

**Basit Doğrusal Regresyon Yöntemi:** Bir girdi bağımsız değişkeniyle ilişkili olan bir çıktı değişkenini bulmak amacıyla kullanılır. Bağımsız değişken girdi ve ilgili bağımlı değişken ise çıktıdır.

**Çoklu Doğrusal Regresyon Yöntemi:** Bağımlı değişken ve iki veya daha fazla bağımsız değişkenin arasındaki ilişkiyi belirlemek amacıyla kullanılır.

Özel bir çoklu doğrusal regresyon yöntemi olan polinom regresyonu da bulunmaktadır. Basit bir doğrusal regresyon modeli, sadece tek bir bağımsız değişkenden meydana gelir.

Çoklu bir doğrusal regresyon modelinin ise iki veya daha fazla bağımsız değişkeni bulunabilir. Karmaşık veri analizi için literatürde farklı doğrusal olmayan regresyon metotları da mevcuttur.

### **3.3. Otomotiv Sektöründe Araç Rotalama Problemi Uygulaması**

Fabrikanın mevcut taşıma sistemine ilişkin verilerin analizinden sonra, milk-run dağıtım rotası belirlemek için araç rotalama problemini sezgisel yöntemler kullanılarak çözülecektir. Bu süreçte, dağıtım maliyetlerinin azaltılması öncelikli hedef olacaktır. En uygun dağıtım rotasını bulmak amacıyla, Clarke-Wright Tasarruf Algoritması, Genetik Algoritma ve Tavlama Benzetimi Algoritması çözülecektir.

Problem belirlendikten sonra ele alınan yöntemler uygulanacaktır. Bu problemler, belirli bir zaman çerçevesi içinde gerçekleştirilmesi gereken taşıma işlemlerini içerir ve verimli bir şekilde yönetilmesi gereken lojistik operasyonlarında önemli bir rol oynar. Manuel çözüm, problem karmaşık olduğu ve tedarikçi sayısının fazla olması nedeniyle mümkün değildir.

#### **3.3.1. Matematiksel Model**

Yapılan çalışmada, otomotiv fabrikası ve tedarikçiler arasında kullanılan milk-run taşıma araçlarının toplam mesafe ve taşıma maliyetlerini minimize etmek amacıyla araç rotalama problemi ele alınarak çözüm önerisi oluşturulmuştur. Problemden bir ana üretim tesisi ve 10 tedarikçi fabrika bulunmaktadır.

- Her tedarikçiye kesinlikle bir defa uğranmalıdır.
- Araç rotaları boyunca tedarikçilerin taleplerinin toplamı, araç kapasitesini aşmamalıdır.
- Tüm tedarikçilere belirlenen zaman aralıklarında hizmet verilmelidir.
- Tüm tedarikçilere zaman pencereleri içerisinde hizmet verilmelidir.
- Her rota, belirlenen maksimum süre içerisinde tamamlanmalıdır.
- Teslimat sürecinde her kilometre için geçen süre sabittir ve dış etkenlerden etkilenmez.

- Araçların geri dönüşleri sırasında boş kasalar organize edilmektedir. Boş kasa çevrimi, proje açısından bir kısıtlama oluşturmamaktadır.
- Tedarikçilerin çalışma saatleri aynı kabul edilmiştir.

### 3.3.2. Problemin ve Kısıtların Tanımlanması

#### Parametreler

$I$	toplam tedarikçi sayısı
$Q$	araç kapasitesi(kg)
$D_j$	tedarikçiden toplanılacak yük miktarı
$t_{ij}$	$i$ ve $j$ noktaları arasındaki ulaşım süresi
$m_{ij}$	mesafe
$k$	araçlar kümesi
$K$	araç kümesi

#### Karar Değişkenleri

$x_{ij}$	Araç $i$ noktasından çıkıp $j$ noktasına geliyorsa 1, aksi halde 0 değerini almaktadır.
$p_{ij}$	Aracın $i$ noktasından sonraki yük miktarı
$i, j = 1, \dots, I; \quad i \neq j$	

#### Amaç Fonksiyonu

$$\text{Min } Z = t. C \sum_{i=1}^I \sum_{j=1}^I (m_{ij} x_{ij}) \quad (3.24)$$

#### Kısıtlar

$$\sum_{i=1, i \neq j}^N x_{ij} = 1, \quad \forall j, \quad j \in \{1, \dots, N\} \quad (3.25)$$

$$\sum_{j=1, j \neq i}^N x_{ij} = 1, \quad \forall i, \quad i \in \{1, \dots, N\} \quad (3.26)$$

$$\sum_{i,j=0}^N \sum_{i \neq j}^N x_{ij} + x_{ji} \leq 1, \quad (3.27)$$

$$\sum_{i,j=0}^N q_i \sum_{j=0, i \neq j}^N x_{ij} \leq Q \quad (3.28)$$

$$p_{jk} \leq Q \quad (3.29)$$

Amaç fonksiyonu (3.24) numaralı kısıt ile ifade edilmektedir. Toplam gidilecek yol mesafesinin minimize edilmesini hedeflemektedir. (3.25) ve (3.26) numaralı kısıtlar, bir aracın kesinlikle bir tedarikçi tarafından ziyaret edilmesi gerektiğini ve aracın o tedarikçiden ayrılması gerektiğini vurgular. (3.27) numaralı kısıt denklemi, depoda başlayıp depoda sonlanmayan turları elemek için kullanılır. Bu kısıt, her turun depo başlangıcı ve depo sonu ile sonlanmasını sağlayarak alt turları ortadan kaldırır. (3.28) numaralı kısıt denklemi ise yüklenen talep miktarlarının, araç kapasitesi  $Q$  değerini aşmaması gerektiğini belirtmektedir. Araç kapasitesi rota boyunca (3.29) numaralı kısıttan kontrol edilmektedir.

### 3.3.3. Metot

Bu çalışma, bir otomotiv fabrikasının tedarik zinciri yönetimindeki kritik bir nokta olan milk-run tedarik süreçlerinin optimizasyonunu hedeflemektedir. Gerçek veriler ile desteklenen çalışma, fabrika ile tedarikçiler arasındaki verimli ve maliyet-etkin rotaları belirleyerek, zaman ve kaynakların etkin kullanımını sağlamayı amaçlamaktadır. Tedarik zinciri yönetimi alanında bu tür analizler, işletmelerin günümüzün karmaşık pazar ortamında başarılı olmalarına destek sağlayacaktır.

Çalışma, klasik sezgisel çözüm yöntemlerinden Clarke-Wright Tasarruf Algoritması'nı, meta sezgisel çözüm yöntemlerinden Tavlama Benzetimi Algoritması ve Genetik Algoritmayı temel alarak araç rotalama problemine odaklanmıştır. Clarke-Wright Tasarruf Algoritması Excel aracılığıyla başarılı bir şekilde çözülmüştür, elde edilen çözüm algoritmanın işleyişini ve ayrıntılarını detaylı bir şekilde sunmaktadır. Operasyonel süreçlerdeki verimliliği artırmak için kritik bir adım olan, Doğrusal Regresyon yöntemi ile tedarikçilerin iki haftalık süreçteki talep miktarları tahminleri doğrultusunda araç rotalaması gerçekleştirilmiştir. Bu yaklaşımlara ek olarak, Tavlama Benzetimi Algoritması ve Genetik Algoritma ile araç rotalama problemi Matlab üzerinde kapsamlı bir şekilde test edilmiş ve araç rotalama problemine yenilikçi bir bakış açısı getirilmiştir. Elde edilen sonuçlar karşılaştırılarak en uygun çözümün belirlenmesini sağlamıştır.

Bu yöntemlerin, tedarik zinciri yönetiminde verimliliği artırmak ve operasyonel mükemmelliği sağlamak için bir araçtır.

### 3.3.3.1. Clarke-Wright Tasarruf Algoritması Uygulaması

Clarke-Wright Tasarruf Algoritması, araç rotalama problemlerinin çözümünde kullanılan klasik sezgisel çözüm yöntemlerinden biridir. Bu algoritma, bir depodan farklı müşterilere yapılan teslimat rotalarını iyileştirmeye odaklanır. Kazanç kavramı, iki ayrı rotanın birleştirilmesinin farklılıklarına göre sağlayacağı avantajı ifade etmek için kullanılır. Bu, sezgisel bir yöntem olmasına rağmen hesaplama süresi bakımından avantajlıdır. Uygulanan Clarke-Wright Algoritması ile oluşturulan çözüm yöntemi, araç rotasının belirlenmesini sağlamıştır.

Çizelge 3.1’de fabrikanın ve tedarikçilerin konumları listelenmektedir. Bu çizelgede, tedarikçilerin ve fabrikanın coğrafi konumları Öklid mesafesi, kuş uçuşu olarak hesaplanmış ve çizelge formatına dönüştürülmüştür. Her bir tedarikçinin konumu ve buna bağlı talep miktarı, araç rotalama probleminde yer alan tedarikçiler ve fabrika arasındaki rotaların oluşturulmasında kullanılmıştır.

**Çizelge 3. 1.** Fabrika ve tedarikçilere ait konum tablosu

	<b>X Koordinatı</b>	<b>Y Koordinatı</b>
<b>Fabrika</b>	50,2	46,4
<b>T1</b>	42,3	48,2
<b>T2</b>	20,7	24,6
<b>T3</b>	16,6	14,7
<b>T4</b>	17,4	28,8
<b>T5</b>	17,6	27,5
<b>T6</b>	24,8	7,8
<b>T7</b>	26,9	48,4
<b>T8</b>	16,0	17,5
<b>T9</b>	24,0	27,0
<b>T10</b>	12,4	18,6

$$d(A, B) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (3.30)$$

Öklid uzaklığı formülasyonu (3.30) numaralı denklem ile gösterilmiştir.

Tedarikçilerden fabrikaya taşınan malzemeler, milk-run kamyonları aracılığıyla fabrikaya taşınmaktadır. Bu taşıma sürecinde her bir kamyonun taşıma kapasitesi 20 ton'dur. Bu şekilde, kamyonlar güvenilir ve etkili bir şekilde tedarikçilerden gelen ürünleri fabrikaya taşıyarak, üretim sürecinin devam etmesini sağlar.

Fabrikaya tedarikçilerden günlük olarak temin edilecek malzeme miktarları iki haftalık süreç boyunca analiz edilmiştir. Malzeme miktarları Çizelge 3.2'de detaylı bir şekilde listelenmektedir. Bu çizelgede, her bir tedarikçiden günlük olarak talep edilen malzeme yük miktarlarının iki haftalık ortalaması alınarak problemde kullanılmıştır. Söz konusu bilgiler, fabrikanın tedarik zinciri planlaması ve günlük operasyonlarının yönetimi için kritik öneme sahiptir. Doğru ve zamanında malzeme temini, üretim süreçlerinin kesintisiz bir şekilde ilerlemesini sağlamaktadır.

**Çizelge 3. 2.** Tedarikçi günlük malzeme talep miktarları

Tedarikçi	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	Günlük Malzeme Yüğü (ton)
<b>T1</b>	2	1,5	2	1	1,5	1	2	4	3	1	2
<b>T2</b>	1	2	2,5	3,5	3	4	3	5	3	3	3
<b>T3</b>	1,5	2,5	3	1	2	4	2,5	1,5	1,5	0,5	2
<b>T4</b>	4,5	3	5	4,5	3	4	5	3	5	3	4
<b>T5</b>	1,5	2	0,25	1	2	0,75	0,5	1	0,5	0,5	1
<b>T6</b>	0,25	0,75	0,5	0,25	0,75	1,5	0,25	0,25	0,25	0,25	0,5
<b>T7</b>	3	4	5	4,5	5	4,5	3	2,5	3	5,5	4
<b>T8</b>	5	5,5	8	5,5	3	6	7	6,5	6	7,5	6
<b>T9</b>	1	1,5	0,5	1,5	2,5	3	4	2,5	2,5	1	2
<b>T10</b>	2,5	3	2,5	3	4	3,5	3	3,5	2,5	2,5	3

Çizelge 3.3'te fabrika ve tedarikçiler arasındaki coğrafi konumların dikkate alınarak oluşturulduğu bir matris yer almaktadır. Bu matris, her bir tedarikçinin diğer tedarikçilere ve fabrikaya olan uzaklıklarını kilometre cinsinden içermektedir. Bu bilgiler, lojistik yönetimi ve tedarik zinciri planlaması açısından oldukça yüksek öneme sahiptir.

Fabrika ile tedarikçiler arasındaki mesafelerin net bir şekilde belirlenmesi, dağıtım rotalarının verimliliğinin artırılması için temel bir bilgi kaynağıdır. Ayrıca, bu matris, karar vericilere doğru ve zamanında lojistik stratejiler geliştirmelerine yardımcı olmak için kritik bir araçtır.

**Çizelge 3. 3.** Tedarikçiler ve fabrika arasındaki uzaklıklar(km) matrisi

	<b>Fabrika</b>	<b>T1</b>	<b>T2</b>	<b>T3</b>	<b>T4</b>	<b>T5</b>	<b>T6</b>	<b>T7</b>	<b>T8</b>	<b>T9</b>	<b>T10</b>
<b>Fabrika</b>	0,0	7,9	29,5	33,6	32,8	32,6	25,4	23,3	34,2	26,2	37,8
<b>T1</b>		0,0	21,6	25,7	24,9	24,7	17,5	15,4	26,3	18,3	29,9
<b>T2</b>			0,0	4,1	3,3	3,1	4,1	6,2	4,7	3,3	8,3
<b>T3</b>				0,0	0,8	1,0	8,2	10,3	0,6	7,4	4,2
<b>T4</b>					0,0	0,2	7,4	9,5	1,4	6,6	5,0
<b>T5</b>						0,0	7,2	9,3	1,6	6,4	5,2
<b>T6</b>							0,0	2,1	8,8	0,8	12,4
<b>T7</b>								0,0	10,9	2,9	14,5
<b>T8</b>									0,0	8,0	3,6
<b>T9</b>										0,0	11,6
<b>T10</b>											0,0

Problemi Clarke-Wright Tasarruf yöntemiyle çözmek için öncelikle tasarruf miktarları hesaplanmıştır. Bu yöntemde, tasarruf miktarlarının doğru bir şekilde belirlenmesi, çözüm sürecinin ilk ve en kritik adımıdır. Hesaplanan tasarruf miktarları, tedarikçiler arasındaki mesafeler ve taşıma kapasiteleri gibi faktörlere dayanılarak belirlenmiştir. Bu hesaplamalar sonucunda Çizelge 3.4'te yer alan tasarruf miktarları matrisi elde edilmiştir. Elde edilen bu matris, sonraki aşamalarda araç rotalarının oluşturulmasında temel bir rehber olarak kullanılmıştır.

Tasarruf matrisinin oluşturulması sürecinde, taşıma maliyetlerinin minimize edilmesi, yolculuk sürelerinin azaltılması ve araç kapasitelerinin etkin bir şekilde kullanılması hedeflenmiştir.

Clarke-Wright Tasarruf yöntemi, bu hedeflere ulaşmada kullanılan etkili bir yöntemdir ve lojistik operasyonlarının optimizasyonunda önemli bir rol oynamaktadır.

**Çizelge 3. 4.** Tasarruf miktarları matrisi

	<b>T2</b>	<b>T3</b>	<b>T4</b>	<b>T5</b>	<b>T6</b>	<b>T7</b>	<b>T8</b>	<b>T9</b>	<b>T10</b>
<b>T1</b>	15,8	7,9	7,9	7,9	7,9	7,9	7,9	7,9	7,9
<b>T2</b>		29,5	29,5	29,5	21,3	17,1	29,5	22,9	29,5
<b>T3</b>			32	31,6	17,2	13	33,6	18,8	33,6
<b>T4</b>				32,4	18	13,8	32,8	19,6	32,8
<b>T5</b>					18,2	14	32,6	19,8	32,6
<b>T6</b>						21,2	25,4	25,4	25,4
<b>T7</b>							23,3	23,3	23,3
<b>T8</b>								18,2	34,2
<b>T9</b>									26,2
<b>T10</b>									

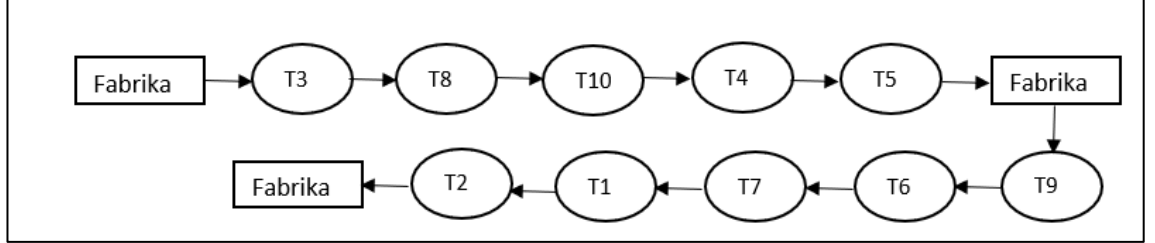
Bu adım, tedarik zinciri rotalarının daha verimli bir hale getirilmesi ve lojistik maliyetlerin iyileştirilmesi için önemli bir adımdır. Matrisin her hücresindeki değerler, ilgili tedarikçi ve alıcı noktaları arasındaki taşıma maliyetlerinin tasarruf edilmesiyle ilgilidir. Her hücre, tedarikçiler arasındaki taşıma maliyeti için tasarruf miktarını temsil etmektedir. Tasarruf miktarlarının doğru belirlenmesi, verimli ve etkili araç rotalarının oluşturulması için temel oluşturur.

**Çizelge 3. 5.** Hesaplanan tasarruf miktarları ve rotalar

Sıra	Tasarruflar	Tedarikçi 1	Tedarikçi 2	Toplam Talep	Hareketler
1	34,2	T8	T10	9	T8 ve T10
2	33,6	T3	T10	5	T3 , T8 ve T10
3	33,6	T3	T8	11	T3 , T8 ve T10
4	32,8	T4	T10	15	T3 , T8, T10, T4
5	32,8	T4	T8	15	T3 , T8, T10, T4
6	32,6	T5	T8	16	T3 , T8, T10, T4, T5,T2
7	32,6	T5	T10	16	T3 , T8, T10, T4, T5,T2
8	32,4	T4	T5	16	T3 , T8, T10, T4, T5,T2
9	32,0	T3	T4	16	T3 , T8, T10, T4, T5,T2
10	31,6	T3	T5	16	T3 , T8, T10, T4, T5,T2
11	29,5	T2	T3	19	T3 , T8, T10, T4, T5,T2
12	29,5	T2	T4	19	T3 , T8, T10, T4, T5,T2
13	29,5	T2	T5	19	T3 , T8, T10, T4, T5,T2
14	29,5	T2	T8	19	T3 , T8, T10, T4, T5,T2
15	29,5	T2	T10	19	T3 , T8, T10, T4, T5,T2
16	26,2	T9	T10	5	T9-T10
17	25,4	T6	T8	6,5	T9-T10-T6
18	25,4	T6	T9	2,5	T9-T10-T6
19	25,4	T6	T10	3,5	T9-T10-T6
20	23,3	T7	T8	10	T9-T10-T6-T7
21	23,3	T7	T9	6	T9-T10-T6-T7
22	23,3	T7	T10	8,5	T9-T10-T6-T7
23	22,9	T2	T9	11,5	T9-T10-T6-T7-T2
24	21,3	T1	T6	13,5	T9-T10-T6-T7-T2-T1

Malzeme taşınması için gerçekleştirilen rotalama sürecinde, tedarikçilerden fabrikaya malzeme akışını sağlamak amacıyla talep miktarları ve araç kapasiteleri detaylı bir şekilde incelenmiştir.

Bu analiz sonucunda, T8 ile T10 arasında belirlenen rotanın en büyük tasarruf potansiyeline sahip olduğu tespit edilmiştir. Tam bir optimizasyon için yapılan tüm iterasyonlar neticesinde, (F-T3 - T8 - T10 - T4 - T5 - T2-F-T9 - T6 - T7 - T1-T2-F) rotası elde edilmiştir. Bu rotalama işleminde milk-run aracı 2 defa kullanılmıştır.



**Şekil 3. 9.** Clarke-Wright tasarruf algoritması ile elde edilen araç rotası

Her iki güzergahta da tedarikçilerden gereken tüm malzemelerin zamanında ve verimli bir şekilde fabrikaya ulaştırılması hedeflenmektedir. Rotalama süreci, sadece en büyük tasarrufu sağlamakla kalmayıp, aynı zamanda lojistik verimliliğini de artırmak için önemle iyileştirilmiştir.

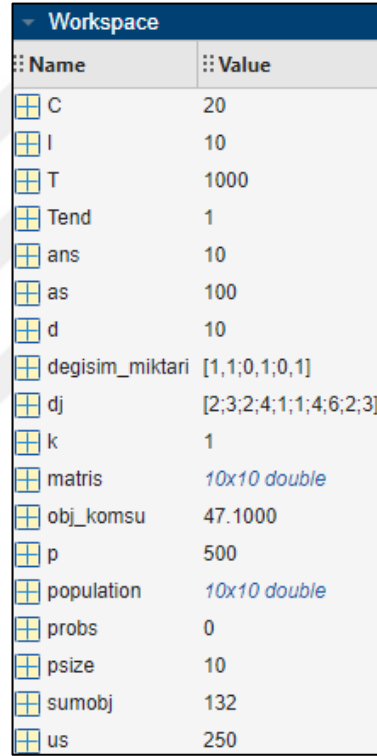
### 3.3.3.2. Tavlama Benzetimi Algoritması Uygulaması

Ele alınan araç rotalama probleminin Tavlama Benzetimi Algoritması ile çözümü Matlab’da sağlanmıştır. Tavlama Benzetimi Algoritması (TB), bir denge durumuna (donma durumu) yaklaşıncaya kadar, soğutma işlemi uygulanan bir sistemdeki enerji değişikliklerini simüle eden bir stokastik arama algoritmasıdır. TB algoritmasında, her turda çözüm alanının taranması sağlanarak bir öncekinden daha iyi bir çözüm seçilmesi hedeflenir. Sıcaklık değeri, bu benzetimde bulunan en iyi çözümün kabul edilme olasılığını belirlemek için daha kötü çözümlerin seçilme olasılığını belirler.

Matlab’da Tavlama Benzetimi Algoritması uygulanarak araç sayısı, araç kapasiteleri, tedarikçiler arasındaki uzaklık, tedarikçiler ve fabrika arasındaki uzaklık matrisleri, fabrika ve tedarikçilerin coğrafi konumları ve tedarikçi yük miktarları kullanılarak bir model oluşturulmuştur. Bu rotalama ile elde edilen sonuçlar gerçek veriler ile karşılaştırılmıştır. Şekil 3.10’da Matlab’da oluşturulan model değişkenleri yer

almaktadır. Burada C araç kapasitesini, I tedarikçi sayısını, matris tedarikçiler ve fabrika arasındaki mesafeyi,  $d_j$  talep miktarlarını, k soğutma katsayısını, Tend son sıcaklık değişkenini, as alt sınır, üs üst sınırı ifade etmektedir.

Tavlama Benzetimi algoritmasıyla çözülen araç rotalama probleminde, 250 üst iterasyon ve 100 alt iterasyon olmak üzere toplamda 25.000 rota kombinasyonu test edilmiştir. İlk sıcaklık değeri 0 olarak belirlenmiş ve sıcaklık düşürme oranı %97 olarak ayarlanmıştır. Kullanılan araçların kapasitesi 20 ton olarak belirlenmiş, müşteri talepleri ise iki haftalık ortalama talep verileri üzerinden hesaplanmıştır.



Name	Value
C	20
I	10
T	1000
Tend	1
ans	10
as	100
d	10
degisim_miktari	[1,1;0,1;0,1]
dj	[2;3;2;4;1;1;4;6;2;3]
k	1
matris	10x10 double
obj_komsu	47.1000
p	500
population	10x10 double
probs	0
psize	10
sumobj	132
us	250

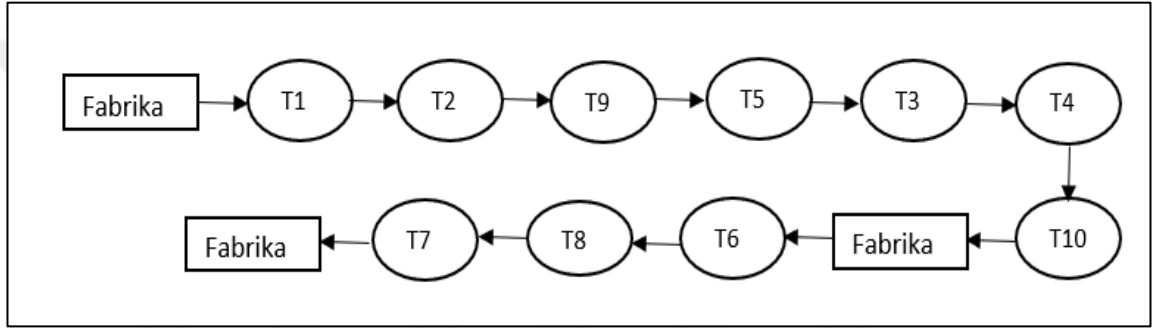
**Şekil 3. 10.** Matlab'da oluşturulan tavlama benzetimi algoritması değişkenleri

EK-1'de Matlab Tavlama Benzetimi Algoritması(TB) kodu yer almaktadır.

Tavlama Benzetimi Algoritması ile elde edilen araç rotalarına göre en kısa yolu veren rota (FT1-T2-T9-T5-T3-T4-T10-F-T6-T8-T7-F) olmuştur bu rotaya ait diyagram Şekil 3.11'de verilmiştir. Bu rotada talep kısıtından dolayı toplamda 2 milk-run aracı

kullanılmıştır. Fabrika taleplerinin milk-run aracı ile karşılandığı araç rotalamanın sonuçları diğer algoritma çıktılarıyla karşılaştırılacaktır.

Matlab kullanılarak gerçekleştirilen Tavlama Benzetimi meta sezgisel algoritmasıyla elde edilen toplam taşıma mesafesi, fabrikanın mevcut araç rotalama mesafesinden daha kısadır. TB algoritmasıyla oluşturulan modelin, gerçek verilerle yapılan testler sonucunda, bilinen bir veri kümesindeki rotalardan daha iyi sonuçlar elde edebileceği gösterilmiştir. Bu, optimuma yakın problem çözümlerinin elde edilmesinde tavlama dayalı algoritmanın etkinliği vurgulanmaktadır.



**Şekil 3. 11.** Tavlama benzetimi algoritması ile elde edilen araç rota diyagramı

Rota maliyetlerinden bu şekilde kazanç elde edilecektir. Oluşturulan model ve kullanılan metodoloji literatüre katkı sağlamaktadır. İleriki çalışmalarda çeşitli yaklaşımlar kullanılarak araştırmalar genişletilebilir. Ek olarak, şirketlere özgü farklı problemlere yönelik ARP modelleri geliştirilerek birbirinden farklı bulgular elde edilebilir.

### 3.3.3.3. Genetik Algoritma Uygulaması

Ele alınan araç rotalama probleminin Genetik Algoritma ile çözümü Matlab'da sağlanmıştır. Toplam milk-run rota mesafesini kısaltabilmek için araç rotaları belirlenmiştir.

Genetik algoritma kullanılarak çözülen problemin popülasyon büyüklüğü 50 olarak belirlenmiştir. Popülasyon büyüklüğü, her bir iterasyonda üretilen potansiyel çözüm adaylarının sayısını ifade etmektedir. Dağıtım rotasındaki tedarikçiler ve fabrika genleri oluşturmaktadır. Araç rotaları, problemin olası çözümünü temsil eden kromozomları

oluşturur. Rastgele belirlenen araç rotalarının kümesi, başlangıç popülasyonunu oluşturur. Kromozomlar, çözüme ait bütün bilgileri içerisinde bulundurulur. Çizelge 3.6'da probleme ait başlangıç popülasyonu ve kromozom yapısı ifade edilmektedir.

Popülasyonda bireylerin gen sıralaması rassal olarak oluşturulmuştur. Popülasyonda bulunan tüm bireylerin uygunluk değerleri hesaplandıktan sonra seçim yapılır. Uygunluk değeri sayesinde olurlu olmayan çözümlerin dikkate alınmaması sağlanır.

**Çizelge 3. 6.** Probleme ait başlangıç popülasyonu ve kromozom yapısı

1. Dizilim	F	T1	T3	T9	T2	T5	T8	T4	T6	T10	T7
2. Dizilim	F	T1	T6	T3	T4	T7	T9	T2	T5	T10	T8
3. Dizilim	F	T2	T1	T4	T9	T6	T3	T5	T7	T8	T10

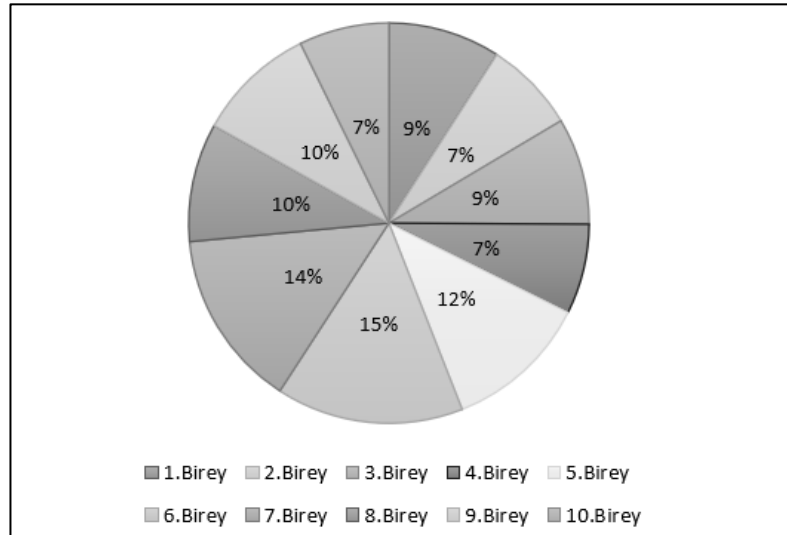
10.Dizilim	F	T2	T1	T3	T9	T4	T6	T7	T10	T8	T5
------------	---	----	----	----	----	----	----	----	-----	----	----

Kromozomlar oluşturulduktan sonra problemin uygunluk değeri hesaplanır. Bu değer, kromozomdaki ardışık her iki nokta arasındaki mesafelerin toplamıyla elde edilir. Problemin amacı, en kısa toplam mesafeye sahip araç rotasını bulmaktır. Bu nedenle, uygunluk değeri ne kadar küçük olursa bulunan çözüm o kadar iyidir. Çizelge 3.7.'de uygunluk değerleri yer almaktadır.

**Çizelge 3. 7.** Uygunluk değerleri

Birey	Uygunluk Değeri
1.Birey	84,7
2.Birey	69,1
3.Birey	80,6
4.Birey	68
5.Birey	109
6.Birey	141,4
7.Birey	134,1
8.Birey	89,5
9.Birey	89,9
10.Birey	68

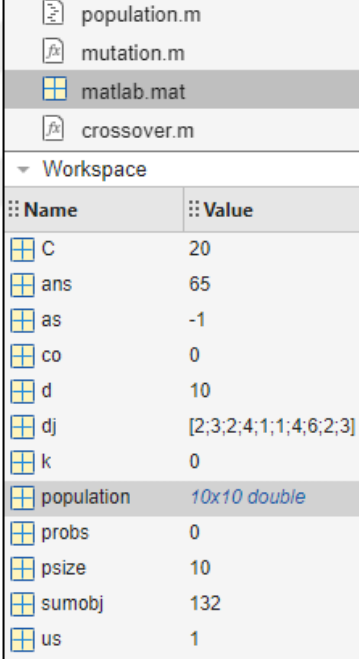
Seçim işleminde rulet tekerleği seçimi yöntemi kullanılmaktadır. Bu süreçte, popülasyondaki en güçlü adaylar, yani daha iyi uygunluk değerine sahip adaylar, bir sonraki popülasyona aktarılma olasılığına sahiptir. Rulet tekerleği seçim yönteminde, her bireyin seçilme olasılığı, uygunluk değeri toplam uygunluk değerine oranıyla belirlenir. Bu yöntemle bireylerin seçilme olasılıkları Şekil 3.12’de gösterilmiştir. Problemin amacı, en kısa mesafeli araç rotasını bulmaktır, bu nedenle çözüm kümesinde en kısa mesafenin en yüksek orana sahip olması sağlanmaktadır.



**Şekil 3. 12.** Rulet çemberi tekniği ile birey seçilme olasılıkları

Sonrasında bu oranların kümülatif toplamları hesaplanır. Yüksek uygunluk değerine sahip bireylerin oranı daha yüksek olduğundan, bir sonraki popülasyona aktarılma olasılığı da yüksek olacaktır. Ardından popülasyon büyüklüğü kadar [0,1] arasında rasgele değerler üretilerek seçim yapılır. Rasgele seçilen iki genin yer değiştirilmesi sağlanır. Çaprazlama sonrasında çözüm kümesi kadar yeni birey elde edilir.

Mutasyon aşamasında, popülasyon içerisindeki çeşitliliğin artırılması hedeflenir. Problem çözümünde çaprazlama oranı 0,60 ve mutasyon oranı=0,1 olarak ele alınmıştır. Bu oranlar, literatürde kullanılan genel değerlerden dikkate alınarak kullanılmıştır. Bu değerler problem çözümü boyunca sabit kalmaktadır. Araç kapasitesi 20 tondur. Sonlandırma kriteri olarak maksimum yineleme sayısı belirlenmiştir. Şekil 3.13'te Matlab'da oluşturulan model değişkenleri yer almaktadır.

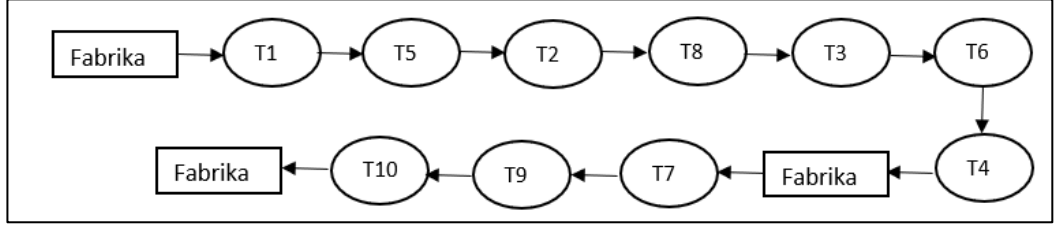


Name	Value
C	20
ans	65
as	-1
co	0
d	10
dj	[2;3;2;4;1;1;4;6;2;3]
k	0
population	10x10 double
probs	0
psize	10
sumobj	132
us	1

**Şekil 3. 13.** Matlab'da oluşturulan genetik algoritma model değişkenleri

Problemde maksimum nesil sayısı 500 olarak belirlenmiştir ve genetik algoritma bu sayı kadar tekrar edecektir. Problemin çözümü, saptanan maksimum yineleme sayısına kadar gerçekleştirilen nesil oluşturma işlemlerinden sonra elde edilen en düşük mesafeli birey olarak kabul edilmiştir. EK-2'de Matlab Genetik Algoritma (GA) kodu yer almaktadır.

Genetik algoritma kullanılarak hesaplanan araç rotalarına göre en kısa yolu veren rota (F-T1-T5-T2-T8-T3-T6-T4-F-T7-T9-T10-F) olmuştur. Oluşturulan rotaya ait diyagram Şekil 3.14’te yer almaktadır. Bu rotada talep kısıtından dolayı toplamda 2 milk-run aracı kullanılmıştır.



Şekil 3. 14. Genetik algoritma ile elde edilen araç rota diyagramı

#### 3.3.3.4. Doğrusal Regresyon ile Geliştirilmiş Çözüm Uygulaması

Doğrusal regresyon, bağımsız değişkenler ve bağımlı değişken arasındaki ilişkiyi modellemek için kullanılan bir regresyon yöntemidir. Bu teknik ile, veri setindeki değişkenler arasındaki ilişkiyi analiz etmek ve bağımlı değişkenin bağımsız değişkenler arasında nasıl etkilendiğini ve veri setindeki değişkenler arasındaki ilişkiyi analiz etmek için kullanılır.

Araç rotalama problemi bağlamında, doğrusal regresyon modeli, geçmiş verileri kullanarak fabrika taleplerinin tahmin edilmesinde kullanılabilir. Bu bağlamda tedarikçi fabrikaların gelecek dönemlerdeki yük miktarlarını tahmin etmek için doğrusal regresyon ile bir model kurulmuştur. Ardından, bu tahminlerin temel alındığı Python ortamında araç rotalama problemi çözümlenmiş ve en verimli dağıtım rotaları oluşturulmuştur. Böylece, tedarik zinciri yönetimi süreçlerinde hem talep tahmini hem de dağıtım planlaması daha etkili bir şekilde gerçekleştirilmiştir.

Bu yöntem, hem tedarik zinciri planlamasını iyileştirerek lojistik operasyonlarının etkinliğini artırmayı hem de araç rotalama problemlerinin çözümüne etki ederek maliyetleri düşürmeyi hedeflemektedir.

Bu bütünsel yaklaşım, tedarikçi fabrikaların taleplerini doğru bir şekilde karşılamak ve lojistik operasyonlarını daha verimli hale getirmek için kritik bir rol oynamaktadır.

Tedarikçilerin gelecek aylarda gerçekleşecek gibi talep miktarlarını tahmin etmek için, geçmiş verilere dayalı olarak aylık bazda talep miktarlarını hesaplayan bir model oluşturulmuştur. Bu model, geçmiş dönemlerdeki talep eğilimlerini analiz ederek ve ilgili faktörleri dikkate alarak gelecekteki talep miktarlarını tahmin etmek için kullanılmıştır. Bu yaklaşım, tedarik zinciri planlamasını güçlendirmek ve üretim kapasitesi planlarını iyileştirmek için önemli bir amaç sağlamaktadır.

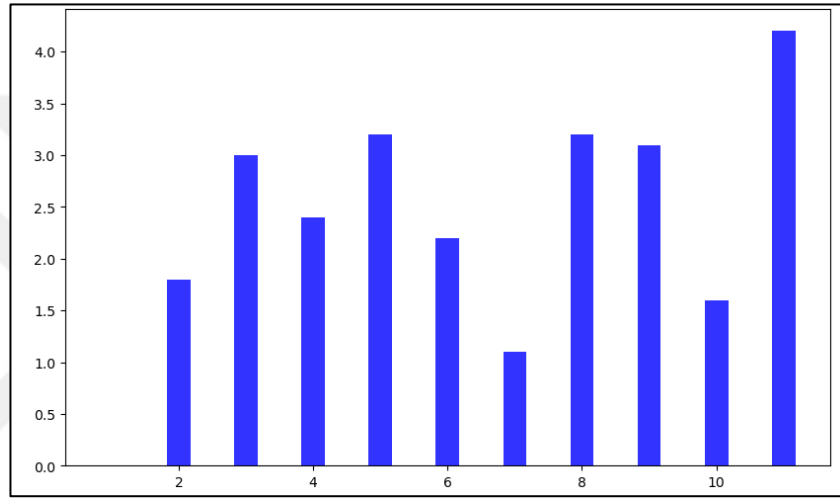
Araç rotalama probleminde yer alan fabrika taleplerini tahminleme yaparken bu makine öğrenmesi yöntemlerine başvurulmuştur. Bu sayede, gelecekteki taleplerin daha doğru ve güvenilir bir şekilde tahmin edilmesi hedeflenmiştir. Bu analiz, geçmiş verilere dayalı olarak yapılan doğru öngörülerle, ileriye dönük 2 haftalık süreç için talep miktarlarını tahminlemeye olanak sağlamıştır. Ayrıca, doğrusal regresyon ile elde edilen tahminler, tedarik zinciri planlama süreçlerinde daha etkin ve verimli kararlar alınmasını desteklemiştir. Çizelge 3.8’de tedarikçilerin bir haftalık talep miktarlarını gösteren tablo yer almaktadır.

**Çizelge 3. 8.** Tedarikçilerin bir haftalık talep miktarları

	<b>G1</b>	<b>G2</b>	<b>G3</b>	<b>G4</b>	<b>G5</b>
<b>T1</b>	2	1	2	3	1,5
<b>T2</b>	3	4	3	2	2,5
<b>T3</b>	2	3	2	3	2,5
<b>T4</b>	4	2	4	2	3
<b>T5</b>	1	4	1	4	1,5
<b>T6</b>	0,5	2	0,5	2	0,5
<b>T7</b>	4	2	4	2	3
<b>T8</b>	6	0,5	3	2	2
<b>T9</b>	2	1	2	1	4
<b>T10</b>	3	4	3	4	2

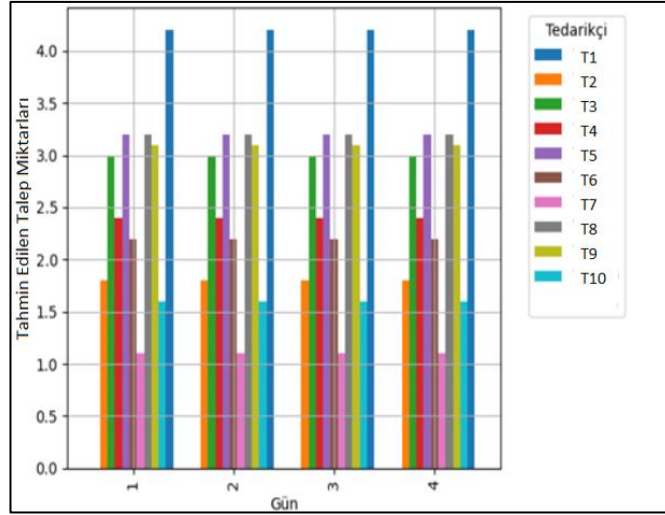
Tedarik zinciri yönetiminde verimliliği artırmak amacıyla, Python'da doğrusal regresyon modeli kullanılarak tedarikçilerin bir haftalık talep miktarları verisi analiz edilerek bu veriler temel alınarak ileriye dönük iki haftalık talep miktarları tahminlenmiştir.

Bu analiz, geçmiş verilere dayalı doğru öngörülerle, ileriye dönük süreç için yük miktarlarını tahminleme olanağı sağlamıştır. Bu sayede, tedarik zincirindeki operasyonel süreçler daha etkin bir şekilde planlanıp ve kaynaklar daha verimli bir şekilde yönetilebilecektir.



**Şekil 3. 15.** Doğrusal regresyon ile tahminleme yapılan talep miktarları grafiği

Doğrusal regresyon modeli kullanılarak, 10 farklı tedarikçinin 2 haftalık süreçteki yük miktarları başarılı bir şekilde tahmin edilmiştir. Bu tahminler, fabrikada dönemsel olarak araç rotalaması yapılmasını sağlayarak, lojistik yönetiminde daha öngörülebilir ve etkili bir yaklaşım benimsenmesine olanak sağlayacaktır. Bu da fabrikanın operasyonel verimliliğini artırmasına fayda sağlayacaktır.



**Şekil 3. 16.** Doğrusal regresyon ile tahminleme yapılan talep miktarları

Tedarikçilere ait yük tahminlerinden yola çıkılarak, fabrikanın iki haftalık milk-run sürecindeki araç rotalaması için bir çözüm geliştirilmiştir. Bu çözüm, Python ortamında OR-TOOLS kütüphanesi kullanılarak oluşturulmuştur. OR-TOOLS, optimize edilmiş araç rotalaması gibi karmaşık problemleri çözmek için son derece etkili bir araçtır. Bu sayede, tedarik zinciri operasyonlarının verimliliği artırılarak taşıma maliyetleri minimize edilmiştir. Analize ait tüm Python kodları EK-3'te yer almaktadır.

Bu çalışma ile, tedarik zinciri dağıtım problemlerinin çözümünde klasik sezgisel ve metal sezgisel algoritmaların uygulamaları, lojistik operasyonlarının verimliliğinin artırılmasına katkı sağlanmıştır.

Tedarikçiler ve fabrika arasındaki mesafelerin hesaplanmasıyla birlikte matris oluşturularak araç rotalaması için çözüm geliştirilmiştir. Şekil 3.17'de tedarikçiler arası mesafe matrisine ait Python ekran görüntüsü yer almaktadır.

Tedarikçiler arasındaki mesafeler:						
Nokta	Fabrika	T1	T2	T3	T4	T5
Nokta						
Fabrika	0.000000	8.102469	36.680921	46.193614	37.223648	37.682489
T1	8.102469	0.000000	31.992499	42.222506	31.565329	32.227007
T2	36.680921	31.992499	0.000000	10.715409	5.341348	4.244997
T3	46.193614	42.222506	10.715409	0.000000	14.122677	12.839003
T4	37.223648	31.565329	5.341348	14.122677	0.000000	1.315295
T5	37.682489	32.227007	4.244997	12.839003	1.315295	0.000000
T6	46.207359	44.027378	17.293062	10.716809	22.265669	20.974508
T7	23.385679	15.401299	24.594308	35.238899	21.780955	22.875751
T8	44.775551	40.424992	8.514693	2.863564	11.386395	10.127191
T9	32.600613	28.005892	4.080441	14.354442	6.841053	6.419502
T10	46.922063	42.073388	10.241582	5.731492	11.359577	10.307764
Nokta						
Nokta						
Fabrika	46.207359	23.385679	44.775551	32.600613	46.922063	
T1	44.027378	15.401299	40.424992	28.005892	42.073388	
T2	17.293062	24.594308	8.514693	4.080441	10.241582	
T3	10.716809	35.238899	2.863564	14.354442	5.731492	
T4	22.265669	21.780955	11.386395	6.841053	11.359577	
T5	20.974508	22.875751	10.127191	6.419502	10.307764	
T6	0.000000	40.654274	13.096946	19.216659	16.443844	
T7	40.654274	0.000000	32.766141	21.595601	33.140459	
T8	13.096946	32.766141	0.000000	12.419742	3.764306	
T9	19.216659	21.595601	12.419742	0.000000	14.322011	
T10	16.443844	33.140459	3.764306	14.322011	0.000000	

**Şekil 3. 17.** Fabrika ve tedarikçiler arasındaki mesafe matrisi

Bu çözüm, her bir tedarikçiden alınacak yük miktarlarını ve bu yükleri fabrikaya taşımak için en verimli rota planını belirlemeyi amaçlamaktadır. Matematiksel modellemeler ve optimizasyon teknikleri kullanılarak, lojistik maliyetlerin minimize edilmesi ve operasyonel verimliliğin artırılması hedeflenmiştir. Tedarikçiler ve fabrika arasındaki konumlar hesaplanmıştır. Bu konumlar arasındaki ilişkiler belirginleştirilerek verilerin daha net bir şekilde anlaşılması sağlanmıştır

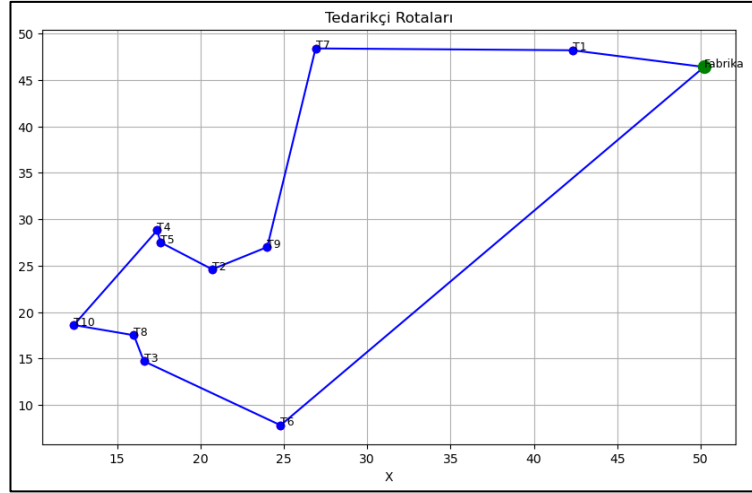
Araç rotalama işlemini gerçekleştirebilmek için tüm olası rotaları bulabilmek adına K-NN En Yakın Komşuluk Algoritması kullanılarak problem çözümüne katkı sağlanmıştır. Her bir noktanın diğer noktalara olan uzaklıkları öklid mesafesi ile elde edilerek tüm uzaklıklar hesaplanmıştır. Daha sonra her bir noktanın en yakın komşusu bulunarak rotalar elde edilmiştir. Başlangıç noktası olan fabrikadan yola çıkılarak en yakın komşuluk durumlarına göre rotalar oluşturulmuştur.

**Çizelge 3. 9.** Talep tahminlemesi sonucu oluşturulan araç rotaları

											Mesafe
<b>ROTA 1</b>	T1	T7	T9	T2	T5	T4	T10	T8	T3	T6	130
<b>ROTA 2</b>	T1	T7	T9	T3	T2	T10	T4	T8	T5	T6	158
<b>ROTA 3</b>	T1	T9	T7	T5	T2	T4	T10	T6	T3	T8	135
<b>ROTA 4</b>	T1	T9	T4	T2	T5	T4	T10	T8	T3	T6	138
<b>ROTA 5</b>	T1	T9	T7	T2	T5	T4	T10	T8	T3	T6	134
<b>ROTA 6</b>	T1	T7	T9	T4	T2	T5	T8	T10	T3	T6	146
<b>ROTA 7</b>	T1	T7	T9	T3	T5	T4	T10	T8	T3	T6	154
<b>ROTA 8</b>	T1	T7	T9	T4	T2	T5	T8	T10	T3	T6	130
<b>ROTA 9</b>	T1	T7	T9	T2	T4	T10	T8	T5	T3	T6	152
<b>ROTA 10</b>	T1	T7	T9	T4	T5	T2	T8	T10	T3	T6	145

Toplamda 10 günlük talep verileri kullanılarak ilgili günlerde planlanması gereken araç rotaları elde edilmiştir. Bu araç rotalarından Rota 1 ile en kısa yol mesafesi elde edilmiştir. Bu rotaya göre, toplamda 129.65 birimlik bir mesafe kat edilmiştir. Bu rotanın, tedarikçiler ve fabrika arasındaki en verimli rota olduğunu gösterir. Talep tahminlerine göre elde edilen rotalar, lojistik maliyetlerinin minimize edilmesini ve operasyonel verimliliği artırmaya yardımcı olacaktır.

Tedarik zinciri yönetiminde bu tür elde edilen rotalar, zaman ve kaynak tasarrufu sağlayarak operasyonel süreçlerin daha etkin bir şekilde yönetilmesine olanak tanır. Bu çerçevede 2 haftalık süre için yapılan araç rotalaması, 10 günlük talep verilerini temel olarak yapılmış ve tedarik zinciri operasyonlarını yönlendirmeye yardımcı olmuştur. Şekil 3.18’de optimal rota için grafik oluşturulmuştur.



Şekil 3. 18. Optimal rotaya ait grafik

#### 4. BULGULAR

Bu bölümde klasik sezgisel çözüm yöntemlerinden Clarke-Wright, meta sezgisel çözüm yöntemlerinden Tavlama Benzetimi ve Genetik Algoritma yöntemleri kullanılarak çözülen araç rotalama problemine ait elde edilen bulgular sunularak tartışılmıştır.

Clarke-Wright Tasarruf Algoritmasının uygulanmasıyla, otomotiv fabrikasının milk-run sürecindeki tedarikçiler ve fabrika arasındaki araç rotalama problemi çözümlenmiştir. Araçların 20 ton kapasitesi göz önünde bulundurularak araç rotalaması (F-T3-T8-T10-T4-T5-F-T9-T6-T7-T1-T2-F) olarak elde edilmiştir. Bu rotalamanın toplam mesafesi 120,1 km olarak hesaplanmıştır. Araç kapasite kısıtlarını aşan yükler için ikinci bir milk-run aracı kullanılmıştır. Mevcut durumda milk-run kamyonetinin aldığı yol 138,4 km olarak hesaplanan toplam araç rotası 120,1 km'ye indirgenerek toplamda 18.3 km'lik bir tasarruf elde edilmiştir. Bu tasarruf aracın harcadığı maliyeti düşürerek işletme için olumlu bir etki yaratacaktır.

Her ne kadar tasarruf miktarı küçük görünse de büyük ölçekli dağıtım süreçlerinde uygulandığında algoritmanın daha büyük tasarruflar sağlayabileceği düşünülmektedir. Literatürde bulunan diğer araç rotalama problemleri algoritmalarıyla farklı çözüm yöntemleri geliştirilebilir ve çeşitli stratejiler ile çözüm yolları çeşitlendirilebilir.

Matlab'da, arama uzayında 25000 adet rota oluşturularak tavlama benzetimi algoritmasıyla ele alınan araç rotalama problemi için, milk-run sürecinde fabrikanın mevcut kullandığı rotanın iyileştirilmesi hedeflenmiştir. İki milk-run aracı kullanılarak elde edilen en iyi araç rotasının toplam mesafesi 128,9 km olarak belirlenmiştir. Bu, fabrikanın mevcut rotalamasına göre 9,5 km'lik bir iyileştirme sağlamaktadır.

Ele alınan araç rotalama problemi için genetik algoritma 500 kez çalıştırılmıştır ve sonuca ulaşılmıştır. Genetik algoritma ile çözümlenen araç rotalama probleminde milk-run sürecinde fabrikanın hali hazırda kullandığı rota optimize edilerek daha verimli bir sonuç elde edilmiştir. İki milk-run aracı kullanılarak genetik algoritma ile elde edilen en iyi araç

rotasının toplam mesafesi 127,2 km olarak bulunmuştur. Bu durumda mevcut durumda 138,4 km olarak hesaplanan araç rotasında 11,2 km'lik bir iyileştirme sağlanmaktadır.

Araç rotalama problemi ile alakalı çözülen tüm algoritmalar karşılaştırıldığında en iyi çözümü veren algoritmanın Clarke-Wright tasarruf algoritması olduğu saptanmıştır. Clarke-Wright algoritması ile 18,3 km'lik yol tasarrufu sağlanmıştır. Bu tasarruf ile toplam alınan yol mesafesinde %13,22'lik bir kazanç elde edilmiştir. Algoritmalara ait çözümler Çizelge 4.1'de bulunmaktadır.

**Çizelge 4. 1.** Çözülen algoritmaların sonuçları

	<b>Clarke Wright Algoritması</b>	<b>Genetik Algoritma</b>	<b>Tavlama Benzetimi Algoritması</b>
<b>Optimum araç rotası</b>	F-T3-T8-T10-T4-T5-F-T9-T6-T7-T1-T2-F	F-T1-T5-T2-T8-T3-T6-T4-F-T7-T9-T10-F	F-T1-T2-T9-T5-T3-T4-T10-F-T6-T8-T7-F
<b>Toplam yol mesafesi</b>	120,1	127,2	128,9
<b>Toplam araç sayısı</b>	2	2	2
<b>Toplam alınan yol kazancı</b>	18,3	11,2	9,5
<b>Kazanç yüzdesi</b>	13,22%	8,09%	6,86%

Çözüm sonuçları incelendiğinde Clarke-Wright tasarruf algoritmasının bu dağıtım probleminde en iyi sonucu verdiği saptanmıştır. Bu algoritma, birçok dağıtım problemi için uygulanabilir ve genellikle optimale yakın sonuçlar verir. Bundan dolayı, milk-run dağıtım süreçlerinde operasyonel verimliliği artırmak için Clarke-Wright algoritmasının tercih edilmesi önerilir.

Doğrusal Regresyon yaklaşımı kullanılarak tedarikçilere ait talep miktarlarının tahminlemesi sonucunda elde edilen veriler, tedarik zinciri yönetiminde önemli bir rol oynamaktadır. İki haftalık(on günlük) süreç için yapılan talep tahminlemesi, tedarikçilerin belirlenmiş rotalara en uygun şekilde dağıtılması için temel bir rehber sağlamaktadır.

Bu yöntem, veri odaklı ve analitik bir yaklaşımı benimseyerek, tedarik zinciri süreçlerinin iyileştirilmesine ve kaynakların verimli bir şekilde kullanılmasına olanak tanır. Talep tahmini sonuçlarıyla elde edilen optimum araç rotaları, tedarikçilerin ve fabrikanın gereksinimlerine daha etkili ve verimli bir şekilde yanıt verilmesini sağlar. Bu, işletmelerin rekabet avantajını artırmak ve maliyetleri minimize etmek için kritik bir faktördür. Dolayısıyla, bu veri odaklı yaklaşımın başarıyla uygulanması, şirketin operasyonel verimliliğini artırabilir ve karar alma süreçlerine pozitif anlamda etki edecektir.



## 5. TARTIŞMA ve SONUÇ

Çalışmada otomotiv fabrikasının milk-run sürecinde yer alan tedarikçiler ile fabrika arasındaki araç rotaları hesaplanmıştır.

Araç kapasitesinin 20 ton olduğu durumda Clarke-Wright Tasarruf algoritması ile elde edilen dağıtım rotası (T3-T8-T10-T4-T5-T2-F-T9-T6-T7-T1) olarak hesaplanmıştır. Bu algoritma ile elde edilen toplam tasarruf 105,8 km olarak hesaplanmıştır. Mevcut durumda milk-run kamyonetinin aldığı yol 138,4 km olarak hesaplanan toplam rota 120,1 km'ye indirgenerek toplamda 18,3 km'lik bir tasarruf elde edilmiştir. Bu durumda aracın harcayacağı yakıt maliyeti miktarında indirgeme sağlanmıştır. Kamyonetin 100 km'de 25 litre yakıt tükettiği ve yakıt litre fiyatının mayıs ayı itibari ile 45 TL olduğu dikkate alındığında, 18,3 km'de araç yaklaşık 4,5 litre yakıt tüketimi gerçekleştirecektir. Bu yakıt miktarının mali değeri ise günlük olarak 205,8 TL'ye tekabül etmektedir. İşletme açısından bakıldığında, bu tür bir tasarrufun milk-run sürecinde önemli bir rol oynadığı görülmektedir. Bir ay 30 gün olarak dikkate alındığında aylık bazda toplam tasarruf miktarı 6176,25 TL ve yıllık bazda ise 74115 TL'lik bir tasarruf sağlanmış olacaktır. Bu tür maliyet tasarrufları, işletmenin genel verimliliğini artırmak ve maliyetleri kontrol altında tutmak için önemli bir faktör olarak dikkate alınmaktadır.

Doğrusal regresyon yöntemi kullanılarak, tedarikçilere ait 2 haftalık talep miktarlarından yola çıkılarak ileriye dönük araç rotalaması oluşturulmuştur. Bu yaklaşım, araç rotalama problemi çözümünde kullanılan algoritmalara ek olarak probleme tahminleme ile talep miktarlarının önceden tahmin edilebilmesi ile farklı bir bakış açısı kazandırmıştır. Bu çözüm yöntemiyle birlikte, tedarik zinciri süreçlerinde yer alan dağıtım problemleri için daha öngörülebilir araç rotalarının oluşturulması sağlanmıştır.

Tedarik zinciri yönetiminde bu tür elde edilen rotalar, zaman ve kaynak tasarrufu sağlayarak operasyonel süreçlerin daha etkin bir şekilde yönetilmesine olanak tanır. Bu çerçevede, tahminleme yöntemleriyle desteklenen araç rota planlaması, tedarik zinciri operasyonlarında daha verimli ve öngörülebilir bir yapı oluşturmak amacıyla önemli bir adımı temsil eder.

Bu çalışma ile tedarik zinciri dağıtım süreçlerindeki araç rotalama problemi analiz edilerek yapılan talep miktarı tahminlemesi ile de uygun araç rotaları elde edilmiştir. Bu sayede, fabrika ve tedarikçiler arasındaki tedarik zinciri ağının etkinliği artmıştır. Bu, işletmelerin rekabet avantajını artırmak ve maliyetleri minimize etmek için kritik bir faktördür. Dolayısıyla, bu veri odaklı yaklaşımın başarıyla uygulanması, şirketin operasyonel verimliliğini artırabilir ve karar alma süreçlerine pozitif anlamda etki edecektir.

Ayrıca, bu çalışmada kullanılan klasik, meta sezgisel çözüm yöntemleri ve doğrusal regresyon ile talep miktarı tahminlemesi, tedarik zinciri yönetimine yenilikçi bir bakış açısı kazandırmıştır. Talep miktarlarının doğru tahmin edilmesi, tedarik zinciri operasyonlarının planlanmasında büyük bir avantaj sağlamış, kaynakların daha verimli kullanımını ve lojistik süreçlerin iyileştirilmesi mümkün olmuştur. Bu sayede maliyetlerin minimize edilmesinin yanında teslimat süreleri kısalarak süreç daha verimli hale getirilmiştir.

Sonuç olarak, bu çalışma, tedarik zinciri yönetiminde veri odaklı yaklaşımların önemini bir kez daha vurgulamaktadır. Gelecekte yapılacak çalışmalar, bu yöntemlerin farklı sektörlerde ve daha geniş ölçekli uygulamalarla test edilmesini sağlayarak, tedarik zinciri yönetiminde daha kapsamlı çözümler sunabilir. Bu sayede, işletmelerin sürdürülebilir rekabet avantajı elde etmesi ve operasyonel anlamda üst düzeye ulaşması mümkün olacaktır.

Bu çalışma ile otomotiv sektöründe yer alan tedarik zinciri milk-run sürecindeki araç rotalama probleminin, klasik sezgisel ve meta sezgisel çözüm yolları ile çözülebileceğini ve talep miktarlarının doğrusal regresyon ile tahmin edilerek ileriye dönük araç rotalaması yapılabileceği gösterilmiştir.

## KAYNAKLAR

- Dantzig, G.B., Ramser J.B. (1959). The Truck Dispatching Problem. *Management Sci.* 6, 80-91.
- Clarke, G., Wright, J.W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, 12, s. 568-581.
- Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI.
- Fisher, M.L., Jaikumar, R. (1981). A Generalized Assignment Heuristic For Vehicle Routing. *Networks* 11, 109-124.
- Glover, F., Laguna, M. (1989). Tabu Search-Part I. *ORSA Journal on Computing*, 1 (3): 190-206.
- Dorigo, M., Maniezzo, V., Colomi, A. (1991). Positive Feedback as a Search Strategy. Technical Report N. 91-016, Politecnico di Milano.
- Laporte, G. (1992). The Vehicle Routing Problem: An overview of exact and approximate algorithms. *European Journal of Operational Research*, 59, 345-358.
- Kennedy, J., Eberhart, C., (1995). Particle Swarm Optimization. *Proc. of IEEE International Conference on Neural Network*, Piscataway, NJ. s. 1942-1948.
- Johnson, D. S. and McGeoch, L. A. (1997). "The Traveling Salesman Problem: A Case Study", *Local Search in Combinatorial Optimization*, 215-310, John Wiley & Sons.
- Eryavuz, M., Gencer, C. (2001). Araç Rotalama Problemine Ait Bir Uygulama, Süleyman Demirel Üniversitesi, İktisadi ve İdari Bilimler Fakültesi, Vol. 6, No. 1, pp. 139-155.
- Braysy, O., Gendreau, M., "Route Construction and Local Search Algorithms for the Vehicle Routing Problem with Time Windows", *Sintef Report, STF42 A01024*, (2001).
- Toth, P., Vigo, D. (2002). The Vehicle Routing Problem. *Windows and Transshipment. Inform*, vol. 44, no. 217-227, ISSN:0315-5986.
- Nilsson, C. (2003). *Heuristics for the Traveling Salesman Problem*. Linköping University.
- Karaboğa, D. (2005). *An Idea Based on Honey Bee Swarm For Numerical Optimization*.
- Kirkpatrick, S., Gelatt C.D. Jr., Vecchi, (1983). Optimization by simulated annealing. *M.P. Science*, 220, (4598): 671-680.

Düzakın, E., Demircioğlu, M. (2009). Araç Rotalama Problemleri ve Çözüm Yöntemleri. Çukurova Üniversitesi İİBF Dergisi Cilt:13. Sayı:1, ss.68-87.

Demir, Ö. (2010). Bir Otomotiv Firmasının İhracat Dağıtım Merkezinde Milk Run Sisteminin Kurulması, Yüksek Lisans Tezi, Kocaeli Üniversitesi Fen Bilimleri Enstitüsü.

Kesen, E.S. (2012). Bütünleşik Üretim ve Dağıtım Problemleri İçin Yeni Bir Çözüm Yaklaşımı: Matematiksel Modelleme, Selçuk Üniversitesi, Mühendislik-Mimarlık Fakültesi, Endüstri Mühendisliği Bölümü. Derg., c.27, s.3, 2012, 139-155, ISSN: 1300-5200.

Balcı, H. Işıklı Esener, İ., Kurban, M. (2012). Regresyon analizi kullanılarak kısa dönem yük tahmini. ELECO-2012 Elektrik - Elektronik ve Bilgisayar Mühendisliği Sempozyumu.

Atmaca, E. (2012). Bir kargo şirketinde araç rotalama problemi ve uygulaması. TÜBAV Bilim Dergisi, 5(2), 12-27.

Doulabi, S.H.H., Seifi, A. (2013). Lower and upper bounds for location-arc routing problems with vehicle capacity constraints. European Journal of Operational Research, 224, s.189–208.

Şahin, Y., Eroğlu, A. (2014). Kapasite Kısıtlı Araç Rotalama Problemi İçin Metasezgisel Yöntemler: Bilimsel Yazın Taraması. Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi C.19, S.4, s.337-355.

Wei, Y., Zhou, L. (2014). Soft time windows associated vehicles routing problems of logistics distribution center using genetic simulated annealing algorithm. Journal of Computing and Information Technology - CIT 22, Special Issue on LISS, s. 31–39.

Yazgan, H., Ercan, S., Arslan, C. (2014). Talep ve Kapasite Kısıtlı Optimizasyon Problemi İçin Yeni Bir Melez Algoritma. Sakarya Üniversitesi, Endüstri Mühendisliği Dergisi. 25(1): 16-28.

Göçken, T., Yaktubay, M., Kılıç, F. (2018). Zaman pencereci araç rotalama problemi çözümü için çok amaçlı genetik algoritma yaklaşımı. Gazi Üniversitesi Fen Bilimleri Dergisi. GU J Sci, Part C, 6(4): 774-786.

Tanyaş, M., Karlı H. (2020). Lojistik Yönetiminin Dijital Dönüşümü: Akıllı Lojistik Üzerine Sistemik Literatür Haritalaması, Optimum Ekonomi ve Yönetim Bilimleri Dergisi, 2020, 7(2), 613-632.

Bozkurt Keser, S., Sarıççek, İ., Yazıcı, A. (2020). İç lojistikte otonom robotlar için görev planlaması. Eskişehir Osmangazi Üniversitesi, Mühendislik-Mimarlık Fakültesi Dergisi. 2020, 28(2), 117-127.

Yazgan, H.R, Ercan Cömert, S., Kılıç, E.N. (2020). Araç rotalama probleminin sezgisel algoritmalar ile çözülmesi: Bir boya fabrikasında uygulama. Journal of Turkish Operations Management, JTOM(4)2, 549-563.

Emeksiz, C., Demir, İ., (2021). Farklı regresyon teknikleri kullanılarak rüzgar hızına etkiyen meteorolojik parametrelerin incelenmesi. ISSN: 2146-8168. 10(3): 100-110.

Şahin, Y., Karagül, K., Aydemir, E., (2021). Heterojen filolu yeşil araç rotalama probleminin tavlama benzetimi yöntemi ile çözümü, Düzce Üniversitesi Bilim ve Teknoloji Dergisi, Sayı:9, (2021) 65-82.

Şehitoğlu, A., İşleyen, Ş., (2023). Yüzüncü Yıl Üniversitesi Fen Bilimleri Enstitüsü Dergisi, Cilt 28, Sayı 1 (Nisan), 252-270.

Meydan, B. (2023). Araç rotalama problemlerinde sezgisel yöntemlerin kullanımı ve bir uygulama, Dumlupınar Üniversitesi İİBF Dergisi, Sayı 12, ISSN: 2791-9579.

Sanlı, Ö., Kartal, Z. (2024). Kapasiteli Araç Rotalama Problemi İçin Makine Öğrenmesi ve Matematiksel Programlama Temelli Hibrid Bir Çözüm Önerisi, Journal of the Faculty of Engineering and Architecture of Gazi University 39:2 (2024) 741-755.

## EKLER

### EK-1 Tavlama Benzetimi Algoritması Matlab Kodları

```
function [cozum obj degisim_miktari obj_komsu] = tavlama(as,us,d,delta,T,Tend,sk)
tedarikcisayisi=10;
tedarikcisayisi=size(matris,1);
cozum=randperm(tedarikcisayisi);
obj=0;
for i=1 tedarikcisayisi-1
tedarikci1=cozum(i);
tedarikci2=cozum(i+2);
obj=obj+matris(sehir1, sehir2);
end
tedarikci1=cozum(end);
tedarikci=cozum(end);
obj=obj+matris(tedarikci1,tedarikci2);
cozum=unifrnd(as,us,[1,d])
obj=sum(cozum.^2);
degisim_miktari=unifrnd(-(us-as)*delta/2,(us-as)*delta/2,[1,d]);
komsu=cozum+degisim_miktari;
delta=0,10
obj_komsu=sum(komsu.^2);
T=1000;
Tend=1;
sk=0,9;
while (T<Tend)
if(obj_komsu<obj)
    cozum=komsu;
    obj=obj_komsu;
    iterasyon=1;
    objit=obj;
    cozumeniyi=cozum;
```

```

objeniyi=obj;
else      /(Geldiğim yer kötü ise kabul olasılığı hesapla)
    de=obj_komsu-obj;
    pa=exp(-de/T);
    rs=unifrnd(0,1);
    if
        rs<pa
            cozum=komsu;
            obj=obj_komsu;
        end
    end
T=T*sk;
iterasyon=iterasyon+1;
if (obj<min(objit))
    objit(iterasyon)=obj;
else
    objit(iterasyon)=objit(iterasyon-1)
    objit(iterasyon)=obj;
    if (objit(iterasyon)<objeniyi)
        cozumeniyi=cozum;
        objeniyi=obj;
    end
end
end

```

## EK-2 Genetik Algoritma Matlab Kodları

```
%population
function Y=population(n,nc)
% nc = number of cities
% n = pop size
B=zeros(n,nc);
for j=1:n
A=1:1:nc;
[x y]=size(A);
for i = 1:nc
r = randi(y);
B(j,i)=A(r);
A(:,r) = [];
[x y]=size(A);
end
end
Y=B;
%crossover
function YY = crossover(X,n)
% X = population
% n = Number of chromosomes to be crossed
[x1 y1] = size(X);
Y = zeros(n,y1);
for z = 1:n
    B = X(randi(x1),:);
    r1 = 1 + randi(y1-1);
    C = B(1,1:r1);
    B(:,1:r1) = []; % cut
    [x3 y3] = size(B);
    B(1,y3+1:y1) = C;
    Y(z,:) = B;
end
```

```

YY = Y;
%mutation
function Y = mutation(X,n)
% X = population
% n = number of chromosomes to be mutated
[x1 y1]=size(X);
Y=zeros(n,y1);
for z=1:n
    A=X(randi(x1),:);
    r1=1+randi(y1-1,1,2);
    while r1(1)==r1(2)
        r1=1+randi(y1-1,1,2);
    end
    B=A(1,r1(1));
    A(1,r1(1))=A(1,r1(2));
    A(1,r1(2))=B;
    Y(z,:)=A;
end
YY = Y;
%Evaluation
%P popülasyon büyüklüğü
%data tedarikçi konumları
function YY = evalutaion(P,data)
[x0 y0]=size(P);
for i =1:x0
    A=P(i,:); %kromozom
    B=zeros(size(A));
    for j1 = 1:y0-1
        [x1 y1]=find(Data(:,1)==A(1,j1));
        [x2 y2]=find(Data(:,1)==A(1,j1+1));
        B(1,j1)=sqrt((Data(x1,2)-Data(x2,2))^2+(Data(x1,3)-Data(x2,3))^2);
    end
end

```

```

[x1 y1]=find(Data(:1)==A(1,j1));
[x2 y2]=find(Data(:1)==A(1,j1+1));
B(1,j1)=sqrt((Data(x1,2)-Data(x2,2))^2+(Data(x1,3)-Data(x2,3))^2);
Y(1,i)=1/sum(B);
end
YY=Y;
%repair
function Y = repair(X,Fc,Lc)
[x y]=size(X);
for i =1:x
    B = x(i,:);
[x1 y1] = find(B== Fc);
B(:,y1)=[];
[x2 y2] = find(B== Lc);
B(:,y2)=[];
A(i,1)=Fc;
A(i,2:y-1)=B;
A(i,y)=Lc;
end
Y=A;
%selection
function [YY1 YY2] = selection(P1,B,p)
[r1 c1]=find(B==max(B));
Y1(1,:)=P1(max(c1),:);
Fn(1,1)=1/B(1,max(c1));
C=sum(B);
D=B/C;
E=cumsum(D);
N=rand(1);
D1=1;
D2=1
While d2<= p-1

```

```

if N<=E(d1)
Y1(d2+1,:)=P1(d1,:);
Fn(1,d2+1);
N=rand(1);
D2=d2+1;
D1=d1+1;
end
end
P=50 %popülasyon_sayısı
C=25 %kromozom_sayısı
M=25; %jenerasyon_sayısı
Fc:2 %ilk_tedarikçi
Data= 50,2 46,4
      42,3 48,2
      20,7 24,6
      16,6 14,7
      17,4 28,8
      17,6 27,5
      24,8 7,8
      26,9 48,4
      16,0 17,5
      24,0 27,0
      12,4 18,6

```

### EK-3 Python Kodları

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import networkx as nx
yuk_matrisi = np.array([
    [0, 0, 0, 0, 0],
    [2, 1, 2, 3, 1],
    [3, 4, 3, 2, 3],
    [2, 3, 2, 3, 2],
    [4, 2, 4, 2, 4],
    [1, 4, 1, 4, 1],
    [0.5, 2, 0.5, 2, 0.5],
    [4, 2, 4, 2, 4],
    [6, 0.5, 3, 2, 4],
    [2, 1, 2, 1, 2],
    [3, 4, 3, 4, 7]
])
son_5_gun_ortalamasi = np.mean(yuk_matrisi[:, -5:], axis=1)
tahmin_edilen_yuk = np.tile(son_5_gun_ortalamasi, (4, 1)).T
print("Tahmin edilen yük miktarları:")
print(tahmin_edilen_yuk)
nokta_sayisi = len(son_5_gun_ortalamasi)
x = np.arange(1, nokta_sayisi + 1)
bar_width = 0.35
```

```

opacity = 0.8

plt.figure(figsize=(10, 6))

plt.bar(x, son_5_gun_ortalamasi, bar_width, alpha=opacity, color='b', label='Son 5 Gün
Ortalaması')

plt.bar(x + bar_width, tahmin_edilen_yuk.flatten(), bar_width, alpha=opacity, color='r',
label='Tahmin Edilen Yük')

tahmin_df = pd.DataFrame(tahmin_edilen_yuk.T, columns=[f'Yük {i+1}' for i in
range(len(tahmin_edilen_yuk))])

tahmin_df.index = np.arange(1, len(tahmin_df) + 1) # günler indeks

print("Tahmin edilen yük miktarları (ilk 10 gün):")

from scipy.spatial import distance_matrix

from ortools.constraint_solver import routing_enums_pb2

from ortools.constraint_solver import pywrapcp

data = {

'Nokta': ['Fabrika', 'T1', 'T2', 'T3', 'T4', 'T5', 'T6', 'T7', 'T8', 'T9', 'T10'],

'Fabrika': [0.000000, 8.102469, 36.680921, 46.193614, 37.223648, 37.682489,
46.207359, 23.385679, 44.775551, 32.600613, 46.922063],

'T1': [8.102469, 0.000000, 31.992499, 42.222506, 31.565329, 32.227007, 44.027378,
15.401299, 40.424992, 28.005892, 42.073388],

'T2': [36.680921, 31.992499, 0.000000, 10.715409, 5.341348, 4.244997, 17.293062,
24.594308, 8.514693, 4.080441, 10.241582],

'T3': [46.193614, 42.222506, 10.715409, 0.000000, 14.122677, 12.839003, 10.716809,
35.238899, 2.863564, 14.354442, 5.731492],

'T4': [37.223648, 31.565329, 5.341348, 14.122677, 0.000000, 1.315295, 22.265669,
21.780955, 11.386395, 6.841053, 11.359577],

'T5': [37.682489, 32.227007, 4.244997, 12.839003, 1.315295, 0.000000, 20.974508,
22.875751, 10.127191, 6.419502, 10.307764],

'T6': [46.207359, 44.027378, 17.293062, 10.716809, 22.265669, 20.974508, 0.000000,
40.654274, 13.096946, 19.216659, 16.443844],

'T7': [23.385679, 15.401299, 24.594308, 35.238899, 21.780955, 22.875751,
40.654274, 0.000000, 32.766141, 21.595601, 33.140459],

```

```

'T8': [44.775551, 40.424992, 8.514693, 2.863564, 11.386395, 10.127191, 13.096946,
       32.766141, 0.000000, 12.419742, 3.764306],

'T9': [32.600613, 28.005892, 4.080441, 14.354442, 6.841053, 6.419502, 19.216659,
       21.595601, 12.419742, 0.000000, 14.322011],

'T10': [46.922063, 42.073388, 10.241582, 5.731492, 11.359577, 10.307764,
        16.443844, 33.140459, 3.764306, 14.322011, 0.000000]

}

df = pd.DataFrame(data)

dist_matrix = distance_matrix(df[['X', 'Y']], df[['X', 'Y']])

dist_df = pd.DataFrame(dist_matrix, index=df['Nokta'], columns=df['Nokta'])

print("Tedarikçiler arasındaki mesafeler:")

print(dist_df)

df.set_index('Nokta', inplace=True)

plt.figure(figsize=(10, 8))

sns.heatmap(df, annot=True, cmap='YlGnBu', fmt=".2f")

plt.title('Tedarikçiler Arasındaki Mesafeler')

plt.xlabel('Nokta')

plt.ylabel('Nokta')

plt.show()

G = nx.Graph()

nodes = df.index.values.tolist()

edges = [(row, col, {'weight': df[col][row]}) for row in df.index for col in df.columns if
         col != row] # Kenarları ve ağırlıklarını ekleyelim

G.add_nodes_from(nodes)

G.add_edges_from(edges)

plt.figure(figsize=(12, 8))

pos = nx.spring_layout(G)

from ortools.constraint_solver import routing_enums_pb2

```

```

from ortools.constraint_solver import pywrapcp

def arac_rotalamasi(mesafe_matrisi, yuk_miktarlari, arac_kapasitesi):

    manager = pywrapcp.RoutingIndexManager(len(mesafe_matrisi), 1, 0)

    routing = pywrapcp.RoutingModel(manager)

    def mesafe_callback(from_index, to_index):

        from_node = manager.IndexToNode(from_index)

        to_node = manager.IndexToNode(to_index)

        return int(mesafe_matrisi[from_node][to_node])

    transit_callback_index = routing.RegisterTransitCallback(mesafe_callback)

    routing.SetArcCostEvaluatorOfAllVehicles(transit_callback_index)

    def demand_callback(from_index):

        from_node = manager.IndexToNode(from_index)

        return int(yuk_miktarlari[from_node])

    demand_callback_index = routing.RegisterUnaryTransitCallback(demand_callback)

    routing.AddDimensionWithVehicleCapacity(

        demand_callback_index,

        0,

        [arac_kapasitesi] * len(veri),

        True,

        'Capacity')

    search_parameters = pywrapcp.DefaultRoutingSearchParameters()

    search_parameters.first_solution_strategy = (

        routing_enums_pb2.FirstSolutionStrategy.PATH_CHEAPEST_ARC)

    solution = routing.SolveWithParameters(search_parameters)

    if solution:

        rotalar = []

```

```

for arac_index in range(routing.vehicles()):
    route = []
    index = routing.Start(arac_index)
    while not routing.IsEnd(index):
        node_index = manager.IndexToNode(index)
        route.append(node_index)
        index = solution.Value(routing.NextVar(index))
        node_index = manager.IndexToNode(index)
        route.append(node_index)
    rotalar.append(route)
return rotalar
else:
    return None
rotalar = arac_rotalamasi(mesafe_matrisi, veri['Yük_1'], 20)
print(rotalar)
points = {
    'Fabrika': (50.2, 46.4),
    'T1': (42.3, 48.2),
    'T2': (20.7, 24.6),
    'T3': (16.6, 14.7),
    'T4': (17.4, 28.8),
    'T5': (17.6, 27.5),
    'T6': (24.8, 7.8),
    'T7': (26.9, 48.4),
    'T8': (16.0, 17.5),
    'T9': (24.0, 27.0),

```

```

    'T10': (12.4, 18.6)
}

def calculate_distance(point1, point2):
    return np.sqrt((point1[0] - point2[0]) ** 2 + (point1[1] - point2[1]) ** 2)

def calculate_distances(points):
    distances = {}

    for point1 in points:
        for point2 in points:
            if point1 != point2:
                distance = calculate_distance(points[point1], points[point2])
                distances[(point1, point2)] = distance

    return distances

def find_nearest_neighbor(distances):
    nearest_neighbors = {}

    for point1 in points:
        min_distance = float('inf')
        nearest_neighbor = None

        for point2 in points:
            if point1 != point2 and distances[(point1, point2)] < min_distance:
                min_distance = distances[(point1, point2)]
                nearest_neighbor = point2

        nearest_neighbors[point1] = nearest_neighbor

    return nearest_neighbors

distances = calculate_distances(points)
nearest_neighbors = find_nearest_neighbor(distances)

for point, neighbor in nearest_neighbors.items():

```

```

    print(f"{point} -> {neighbor}")

def create_route(start_point, nearest_neighbors):

    route = [start_point]

    current_point = start_point

    while True:

        next_point = nearest_neighbors[current_point]

        route.append(next_point)

        if next_point == start_point:

            break

        current_point = next_point

    return route

start_point = 'Fabrika'

route = create_route(start_point, nearest_neighbors)

print("Oluşturulan Rota:")

print(" -> ".join(route))

def calculate_total_distance(route, distances):

    total_distance = 0

    for i in range(len(route) - 1):

        total_distance += distances[(route[i], route[i+1])]

    return total_distance

total_distance = calculate_total_distance(route, distances)

print("Oluşturulan Rotanın Toplam Mesafesi:", total_distance)

data = {

df = pd.DataFrame(data)

plt.figure(figsize=(10, 6))

plt.scatter(df["X"], df["Y"], color='blue')

```

```

for i, txt in enumerate(df["Nokta"]):
    plt.annotate(txt, (df["X"][i], df["Y"][i]), textcoords="offset points", xytext=(0,10),
                  ha='center')

plt.title("Konumlar")

plt.xlabel("X")

plt.ylabel("Y")

plt.grid(True)

plt.show()

print("Rotalar:", rotalar)

print("Toplam Mesafe:", toplam_mesafe)

rotalar_x = [tedarikciler[tedarikci][0] for tedarikci in rotalar]
rotalar_y = [tedarikciler[tedarikci][1] for tedarikci in rotalar]

tedarikciler_x = [koordinat[0] for koordinat in tedarikciler.values()]
tedarikciler_y = [koordinat[1] for koordinat in tedarikciler.values()]

plt.figure(figsize=(10, 6))

plt.plot(rotalar_x, rotalar_y, marker='o', linestyle='-', color='b')

plt.plot(rotalar_x[0], rotalar_y[0], marker='o', markersize=10, color='g') # Başlangıç
noktasını yeşil olarak işaretle

plt.scatter(tedarikciler_x, tedarikciler_y, color='r')

for tedarikci, koordinat in tedarikciler.items():
    plt.text(koordinat[0], koordinat[1], tedarikci, fontsize=9)

plt.xlabel('X')

plt.ylabel('Y')

plt.title('Tedarikçi Rotaları')

```