



T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ARAÇ ROTALAMA PROBLEMLERİNİN ÇÖZÜMÜ İÇİN YENİ BİR META-
SEZGİSEL YAKLAŞIM: ELEKTROMANYETİK ALGORİTMA

Alkın YURTKURAN

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA-2009



T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ARAÇ ROTALAMA PROBLEMLERİNİN ÇÖZÜMÜ İÇİN YENİ BİR META-
SEZGİSEL YAKLAŞIM: ELEKTROMANYETİK ALGORİTMA

Alkın YURTKURAN

Prof. Dr. Erdal EMEL
(Danışman)

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

BURSA-2009

T.C.
ULUDAĞ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ARAÇ ROTALAMA PROBLEMLERİNİN ÇÖZÜMÜ İÇİN YENİ BİR META-
SEZGİSEL YAKLAŞIM: ELEKTROMANYETİK ALGORİTMA

Alkın YURTKURAN

YÜKSEK LİSANS TEZİ
ENDÜSTRİ MÜHENDİSLİĞİ ANABİLİM DALI

Bu Tez 27/07/2009 tarihinde aşağıdaki jüri tarafından oybirliği ile kabul edilmiştir.

Prof.Dr. Erdal EMEL
Danışman

Prof.Dr. Gökay KAYNAK
Jüri Üyesi

Doç.Dr. H.Cenk ÖZMUTLU
Jüri Üyesi

ÖZET

Dağıtım ve toplama sistemlerinin temelini oluşturan araç rotalama problemlerinin üzerinde özellikle son yıllarda artan sayıda birçok detaylı araştırma yapılmıştır. Bu çalışma kapsamında, kapasite kısıtlı araç rotalama problemlerinin çözümü için kesikli değişkenlerden oluşan eniyileme problemlerinin çözümünde yakın zamanda kullanılmaya başlanan, Elektromanyetik Algoritma isimli yeni bir meta-sezgisel yöntem önerilmiştir. Popülasyon tabanlı olan Elektromanyetik Algoritma, temel olarak elektrik yüklü parçacıkların birbirlerine uyguladıkları itme ve çekme kuvvetlerini esas alır. Bu çalışmada, Elektromanyetik Algoritmanın amaç fonksiyonu hesaplama prosedürüne Rassal Anahtar metodu entegre edilerek , araç rotalama gibi kombinatorik eniyileme problemlerinin çözümü için uygun hale gelmesi sağlanmıştır. Ek olarak, önerilen algoritma yerel bir iyileştirme yöntemi olan İterasyonlu Yer Değiştirme yöntemi ile melezleştirilmiştir. Geliştirilen bu melez algoritma, literatürden alınan test problemleri üzerinde uygulanmış ve diğer meta-sezgisel yöntemler ile karşılaştırılmıştır. Sonuçlar incelendiğinde, Elektromanyetik Algoritmanın kabul edilebilir işlem süreleri içinde, kaliteli çözümler elde ettiği görülmüştür.

Anahtar Kelimeler: Kapasite kısıtlı araç rotalama problemi, Elektromanyetik algoritma, Meta-sezgisel yöntemler.

ABSTRACT

Vehicle routing problems have been extensively analyzed within the last few decades, since they play a central role in optimization of distribution networks. This study presents a new hybrid Electromagnetism-like Algorithm for solving capacitated vehicle routing problems. Electromagnetism-like Algorithm is a population-based algorithm based on attraction-repulsion mechanisms between charged particles. A modified objective function value calculation approach, utilizing the Random-Key procedure makes it possible for Electromagnetism-like Algorithm to solve known combinatorial optimization problems like capacitated vehicle routing problems. Here, the solutions obtained such are improved by a relatively new local search method, Iterated Swap Procedure, and tested on several benchmarking problems. The computational results show that the proposed Hybrid Electromagnetism-like Algorithm gives promising results within acceptable computational times when compared to other novel meta-heuristics.

Key Words: Capacitated vehicle routing problems, Electromagnetism-like algorithm, Meta-heuristics.

İÇİNDEKİLER

	Sayfa
TEZ ONAY SAYFASI	II
ÖZET	III
ABSTRACT	IV
İÇİNDEKİLER	V
KISALTMALAR DİZİNİ	VII
TABLolar DİZİNİ	VIII
ŞEKİLLER DİZİNİ	IX
SİMGELER DİZİNİ	X
GİRİŞ	1
1. KURAMSAL TEMELLER	4
1.1. Araç Rotalama Problemi	4
1.2. Araç Rotalama Problemlerinin Kısa Tarihçesi	6
1.3. Araç Rotalama Problem Tipleri	7
1.3.1. Kapasite Kısıtlı Araç Rotalama Problemi	8
1.3.2. Mesafe Kısıtlı Araç Rotalama Problemi	8
1.3.3. Zaman Pencereli Araç Rotalama Problemi	9
1.3.4. Dağıtım ve Toplamayı İçeren Araç Rotalama Problemi	10
1.3.5. Diğer Araç Rotalama Problemleri	11
2. KAPASİTE KISITLI ARAÇ ROTALAMA PROBLEMİ	13
2.1. Problemin Tanımı	13
2.2. Matematiksel Model	15
2.3. Çözüm Yöntemleri	16
2.3.1. Kesin Çözüm Yöntemleri	16
2.3.2. Sezgisel Çözüm Yöntemleri	17
2.3.2.1. Tur Kurucu Metotlar	17
2.3.2.2. İki Aşamalı Metotlar	17
2.3.2.3. Tur Geliştirici Metotlar	18
2.3.3. Meta-sezgisel Yöntemler	19
2.3.3.1. Tabu Arama	19
2.3.3.2. Tavlama Benzetimi	20
2.3.3.3. Karınca Kolonisi Algoritması	21
2.3.3.4. Genetik Algoritma	21
2.3.3.5. Parçacık Sürüsü Optimizasyonu	22

	Sayfa
3. ELEKTROMANYETİK ALGORİTMA	23
3.1. Giriş	23
3.2. Araç Rotalama Problemleri için EMA	29
3.2.1. Yük-Kuvvet ve Hareket Değerlerinin Hesaplanması	29
3.2.2. Amaç Fonksiyonu Hesaplama Prosedürü	32
3.2.3. Yerel İyileştirme	35
3.2.4. Melez Elektromanyetik Algoritma	36
4. ARAŞTIRMA SONUÇLARI ve YORUMLAR	38
4.1. Christofides Test Problemleri	39
4.2. Golden Test Problemleri	45
SONUÇ	51
KAYNAKLAR	53
ÖZGEÇMİŞ	67
TEŞEKKÜR	69

KISALTMALAR DİZİNİ

ARP	-	Araç Rotalama Problemi
KKARP	-	Kapasite Kısıtlı Araç Rotalama Problemi
ZPARP	-	Zaman Pencere Araç Rotalama Problemi
DTARP	-	Dağıtım ve Toplamayı içeren Araç Rotalama Problemi
MKARP	-	Mesafe Kısıtlı Araç Rotalama Problemi
SDTARP	-	Simultane Dağıtım ve Toplamayı içeren Araç Rotalama Problemi
SARP	-	Stokastik Araç Rotalama Problemi
TA	-	Tabu Arama
TB	-	Tavlama Benzetimi
GA	-	Genetik Algoritma
KKA	-	Karınca Kolonisi Algoritması
PSO	-	Parçacık Sürüsü Optimizasyonu
EMA	-	Elektromanyetik Algoritma
HEMA	-	Melez Elektromanyetik Algoritma
AFH	-	Amaç Fonksiyonu Hesaplama
RA	-	Rassal Anahtar
İYD	-	İterasyonlu Yer Değiştirme

TABLULAR DİZİNİ

		Sayfa
Tablo 1.1	ARP'nin Çözüm Yöntemlerinin Tarihsel Gelişimi	7
Tablo 4.1	HEMA Parametreleri	39
Tablo 4.2	Christofides Test Problemleri	39
Tablo 4.3	EMA ve HEMA Karşılaştırılması	41
Tablo 4.4	Christofides Test Problemleri Sonuçları	43
Tablo 4.5	Christofides Test Problemleri için İşlem Süreleri	45
Tablo 4.6	Golden Test Problemleri	46
Tablo 4.7	Golden Test Problemleri Sonuçları	49
Tablo 4.8	Golden Test Problemleri için İşlem Süreleri	50

ŞEKİLLER DİZİNİ

	Sayfa	
Şekil 2.1	Örnek bir KKARP	14
Şekil 2.2	KKARP Çözüm Örneği	14
Şekil 3.1	Parçacık i ve Pozisyon Vektörü Bileşenleri	26
Şekil 3.2	Parçacıkların F ile Hareket Ettirilmesi	26
Şekil 3.3	Elektromanyetik Algoritma'nın Temel Adımları	28
Şekil 3.4	Başlangıç Fonksiyonu Sözde Kodu	28
Şekil 3.5	S_Kontrol Fonksiyonu Sözde Kodu	29
Şekil 3.6	Kuvvet, Yük ve Hareket Değerlerinin Hesaplanması	31
Şekil 3.7	Güncellenmiş S_Kontrol Mekanizması Sözde Kodu	32
Şekil 3.8	Güncellenmiş Başlangıç Fonksiyonu Sözde Kodu	32
Şekil 3.9	Rassal Anahtar Yöntemi	33
Şekil 3.10	Güncellenmiş Amaç Fonksiyonu Hesaplama Prosedürü	34
Şekil 3.11	İterasyonlu Yer Değiştirme Metodu	35
Şekil 3.12	Melez Elektromanyetik Algoritma Sözde Kodu	37
Şekil 3.13	Rulet Çemberi Sözde Kodu	37

SİMGELER DİZİNİ

C	-	Araç kapasitesi
c_{ij}	-	(i, j) ark maliyeti
t_{ij}	-	(i, j) arkı gidiş süresi
x_{ij}	-	(0-1) tamsayı karar değişkeni
x_{ij}^k	-	(0-1) tamsayı karar değişkeni
L	-	Maksimum tur uzunluğu
K	-	Araç sayısı/seti
V	-	Düğüm seti
a_i	-	Müşteri i için en erken teslim zamanı
b_i	-	Müşteri i için en geç teslim zamanı
N	-	Müşteri sayısı
d_i	-	Müşteri i 'nin talebi
$f(x_i)$	-	Parçacık i 'nin amaç fonksiyonu değeri
\mathbf{x}_i	-	Parçacık i 'nin pozisyon vektörü
q_i	-	Parçacık i 'nin elektrik yükü
\mathbf{F}_i	-	Parçacık i 'ye uygulanan bileşke kuvvet
U^k	-	k . bileşen için üst sınır
L^k	-	k . bileşen için alt sınır
q_{ij}	-	Parçacık i 'nin parçacık j 'ye göre elektrik yükü
\mathbf{F}_{ij}	-	Parçacık j 'nin parçacık i 'ye uyguladığı kuvvet
U	-	Pozisyon vektörü için üst sınır
L	-	Pozisyon vektörü için alt sınır
m	-	Popülasyon büyüklüğü

GİRİŞ

Lojistik, tedarikçiden müşteriye kadar olan hammadde, yarı mamul ve mamul hareketlerinin ve depolanma faaliyetlerinin yönetimidir. Günümüzde, artan rekabet nedeniyle, sadece fabrika içi süreçlerin değil, aynı zamanda fabrika dışı taşıma ve toplama süreçlerinin detaylı incelenmesi ve uygun bir şekilde çözümlenmesi gerekmektedir. Lojistik sistemlerin temelini oluşturan, malzemelerin dağıtım ve toplama problemlerinin en verimli ve etkin şekilde çözümlenmesi, işletmelerin karlılığı için çok önemli bir rol oynar. Lojistik yönetiminin temel amacı, hammaddenin satın alınmasından, bitmiş ürünün satışına kadar olan dağıtım, toplama ve depolama süreçlerinin mümkün olan en düşük maliyet ile planlanmasıdır. Tüm bu dağıtım ve toplama faaliyetlerinin planlanmasının en zor ayağını ise araç rotalama problemleri oluşturmaktadır.

Araç rotalama problemi, bir veya birden çok depodan alınan malzemelerin, özellikleri belirli araçlar ile belirtilen kısıtlar altında en düşük maliyet ile müşterilere dağıtılması olarak tanımlanabilir. Araç rotalama problemleri sadece üretim sistemleri içerisinde karşılaşılan bir problem değildir. Aynı zamanda, personel veya öğrenci servislerinin rotalarının planlanması, toplu taşıma sistemlerinin planlanması, şehir içi ve dışı kargo sistemlerinin yönetimi, katı atık toplama sistemlerinin yönetimi gibi günümüzde birçok problemin temelini oluşturmaktadır.

Literatürde, araç rotalama problemlerinin farklı tiplerine rastlamak mümkündür. Bu problem tiplerini birbirlerinden ayıran en önemli özellik, temel araç rotalama probleminin üstüne eklenen kısıtlardır. Araç rotalama problemlerinin temeli ve en basiti, kapasite kısıtlı araç rotalama problemidir. Kapasite kısıtlı araç rotalama problemi, kapasiteleri sabit araçlar ile müşterilerin taleplerinin en düşük maliyetle karşılanması için her araca en uygun rotaların

atanması olarak açıklanabilir. Her rotadaki toplam müşteri taleplerinin araç kapasitesini geçmemesi problemin en önemli kısıtını oluşturmaktadır. Araç rotalama problemlerinin temeli olan bu problem tipi “NP-Zor” tipte karmaşıklığa sahiptir. Başka bir deyişle, bu tip problemlerde değişken sayısı arttıkça, karmaşıklık çok daha hızlı büyümektedir. Kapasite kısıtlı araç rotalama problemlerinin daha verimli ve kısa sürede çözümlenebilmesi için araştırmacılar bugüne kadar birçok kesin, sezgisel ve meta-sezgisel çözüm yöntemleri önermişlerdir.

Bu çalışma kapsamında, kapasite kısıtlı araç rotalama problemlerinin çözümü için, kombinatorik problemlerin çözümünde yakın zamanda kullanılmaya başlayan, yeni bir meta-sezgisel yöntem olan, elektromanyetik algoritma önerilmiştir. Elektromanyetik algoritma temel olarak, elektrik yüklü parçacıkların birbirlerine uyguladıkları itme/çekme kuvvetleri analogisini esas alan, popülasyon tabanlı bir meta-sezgisel yöntemdir. Elektromanyetik algoritmada, her parçacık bir çözümü gösterir ve üzerindeki yük ise çözüm kalitesi ile doğru orantılıdır. Parçacıklar üzerlerindeki yükler sayesinde birbirlerine itme/çekme kuvveti uygularlar. Popülasyondaki parçacıkların birbirlerine uyguladıkları kuvvetler sonucunda, parçacıklar hareket ettirilir. Böylece, popülasyonun global bir en iyiye doğru hareket edeceği öngörülmektedir. Önerilen bu algoritma, yerel bir iyileştirme yöntemi ile melezleştirilmiş ve literatürden alınan otuzun üzerinde test probleminin üzerinde uygulanmıştır. Aynı zamanda, çokça kullanılan diğer sezgisel ve meta-sezgisel yöntemler ile çözüm kalitesi ve süre bakımından karşılaştırılmıştır. Sonuçlar incelendiğinde, elektromanyetik algoritmanın diğer temel meta-sezgisel algoritmalar ile karşılaştırılabilir süreler içinde, kaliteli çözümler elde edebildiği görülmektedir.

Çalışmanın bundan sonraki kısımlarında, Bölüm 1’de, genel olarak farklı araç rotalama problemlerinin tipleri ve özellikleri, literatürden faydalanılarak özetlenmiştir. Geliştirilen meta-sezgisel algoritmanın da uygulandığı, kapasite kısıtlı araç rotalama problemi ise Bölüm 2’de detaylı olarak incelenmiştir. Bölüm 3 ve 4 ise, geliştirilen yöntem, uygulama sonuçları ve yorumları içermektedir. Son

bölümde ise, konu hakkında ileride yapılabilecek arařtırmalar ile ilgili öneriler sunulmuřtur.

1. KURAMSAL TEMELLER

1.1 Araç Rotalama Problemleri

Araç Rotalama Problemleri (ARP); kapasitesi belirli (C) araçlar ile belirlenen müşterilere en uygun şekilde hizmet vermek için her araca en uygun rotayı atamayı kapsar. ARP'nin çözümünde oluşturulan rotalar genel olarak aşağıdaki özellikleri sağlamalıdır:

- (1) Her aracın rotası depoda başlayıp, depoda bitmeli,
- (2) Tüm müşteri gereksinimleri (talepleri) karşılanmalı,
- (3) Her müşteriye bir kez uğranmalı,
- (4) Diğer sistem kısıtları sağlanmalı,
- (5) Toplam maliyet en aza indirgenmeli.

ARP birçok farklı bileşenden oluşmaktadır. Bunların en önemlileri; yol ağı, müşteriler, araçlar ve depo(lar) olarak açıklanabilir (Toth ve Vigo 2000).

Yol ağı çoğu zaman bir graf üstünde arklar ve düğümler ile tanımlanır. Düğümler, müşterileri veya depoyu, arklar ise bu düğümleri birleştiren yolları gösterir. Düğümler arasındaki arklar, problemin özelliğine bağlı olarak tek yönlü veya çift yönlü olabilir. Her (i,j) düğüm çiftinin arasındaki arkın belirli bir maliyeti (c_{ij}) vardır. Maliyet genellikle, arkın uzunluğu, aracın veya taşınacak ürünün özelliğine bağlı olarak i 'den j 'ye gidiş süresi olarak tanımlanır.

ARP'nin diğler bir bileşeni müşterilerdir. Müşterilerin tipik karakteristikleri ise aşağıdaki gibi sıralanabilir:

- (1) Talep miktarı (dağıtılması veya toplanması gereken),
- (2) Müşteriye hizmet edilmesi gereken zaman aralığı (Zaman penceresi),
- (3) Taleplerin toplanması veya dağıtılması için müşteride geçen zaman (Servis zamanı),
- (4) Müşterinin konumu (genellikle x-y düzleminde koordinatlar şeklinde)

Müşterilerin taleplerini karşılamak için oluşturulan her rota bir depodan başlar ve belirtilen bir depoda biter. Dağıtım sistemi içinde birden çok depo olabilir. Ek olarak, bu tip durumlarda rotaların başlama ve bitiş noktaları farklı depolarda olabilir. Bazı problem tiplerinde ise depodan başlayıp, uğranan en son müşteride biten rotalara da izin verilmektedir. Her depo, karşılayabileceği talep miktarı (genelde sınırsız alınır) ve pozisyonu ile tanımlanır.

ARP'nin diğler bileşeni ise araçlardır. Kullanılan her araç genelde özdeştir ve sabit hızları ve kapasiteleri vardır. Kapasite genellikle, aracın taşıyabileceği malzeme miktarı olarak tanımlanır.

ARP'nin en önemli bileşeni rotalardır. Problem tiplerini birbirinden ayıran kısıtlar, çoğu zaman uygun rotalar oluşturularak sağlanır. Oluşturulacak rotalar; müşteri, araç veya sistemin karakteristiklerinden oluşan kısıtları sağlamak zorundadır. Uygulanan bazı temel kısıtlar ise; bir tur içerisinde taşınacak malzemenin aracın kapasitesini aşmaması, müşterilere servis edilmesi gereken bir zaman diliminin olması, bazı müşterilerin dağıtım, bazılarının ise toplama ağında bulunması, müşteriler arasında öncelik ilişkisi bulunması vb. şeklinde sıralanabilir.

Her eniyileme probleminde olduğu gibi, ARP'de de amaç fonksiyonu belirlenmelidir. Yukarıda da açıklandığı gibi, ARP'de kullanılan her arkın belirli bir maliyeti (c_{ij}) vardır ve genelde bu maliyet müşteri çiftleri (i, j) arasındaki uzaklık ile ifade edilir. Bazı problem tipleri simetrik ($c_{ij} = c_{ji}$) veya asimetrik ($c_{ij} \neq c_{ji}$) olabilir. Bu maliyet olarak yol uzunluğu yerine müşteri i 'den müşteri j 'ye gidiş

süresi (t_{ij}) olarak da alınabilir. Fakat genellikle ark uzunluğu, harcanan süreye eşittir ($t_{ij} = c_{ij}$). Başka bir deyişle, düğümler arasında oluşturulan süre matrisi ile uzunluk matrisi çoğu zaman simetrik ve birbirlerine eşit kabul edilir. ARP'de kullanılan tipik amaç fonksiyonları aşağıdaki gibi sıralanabilir:

- (1) Toplam rota uzunluğu en küçüklemesi,
- (2) Toplam harcanan zamanın en küçüklemesi,
- (3) Kullanılan araç sayısının en küçüklemesi.

1.2 Araç Rotalama Problemlerinin Kısa Tarihçesi

Araç rotalama ile ilgili ilk çalışma Danzig, Fulkerson ve Johnson (1954) tarafından yapılmıştır. Bu çalışmada, 15 müşterili bir ARP ele alınmış ve çözümlenmiştir. Clarke ve Wright (1964) ilk defa birden çok aracın bulunduğu araç rotalama problemlerini literatüre katmışlardır. Bu çalışma tam anlamıyla ilk araç rotalama çalışması olarak değerlendirilebilir.

“Araç Rotalama” olarak adlandırılan ilk çalışmalar ise, Golden ve diğ. (1972) tarafından yapılmıştır. ARP'ye benzer olan ilk çalışmalar ise; uçak rotalama (Levin 1972), ulaşım ağı tasarımı (O'Conner ve Sussman 1970), hizmet araçlarının rotalanması (Marks ve Stricker 1970), ulaşım sistemlerinin işletimi (Watson-Gandy ve Christofides 1971) ve katı atık toplama sistemlerinin tasarımı (Liebman 1970) olarak sıralanabilir. Solomon (1983), literatürde “Solomon Problemleri” olarak geçen, örnek problem setlerini hazırlamış ve birçoğunu optimal olarak çözmüştür. Daha detaylı bilgi için, Eksioğlu'nun (2009) çalışması incelenebilir.

1980'lerde, işlemci hızlarının günümüzdeki hızlardan çok düşük olması, araştırmacıları daha karmaşık olan ARP'lerden uzak tutmuştur. Fakat 1990'lar ile başlayan ve günümüze kadar uzanan süreçte, ARP üzerinde birçok farklı çalışmalar yapılmıştır. Araştırmacılar, optimal veya optimale yakın çözümleri bulmak amacıyla, kesin çözüm yöntemleri, sezgisel veya meta-sezgisel yöntemler önermişlerdir. ARP'lerin çözüm yöntemlerinin tarihsel gelişimi özet olarak Tablo 1.1'de verilmiştir.

Tablo 1.1 Araç Rotalama Problemlerinin Çözüm Yöntemlerinin Tarihsel Gelişimi

Yıllar	Gelişmeler
1950'ler	ARP tamsayılı programlama olarak formüle edilmiş ve 10-20 müşterili küçük problemler çözülmüştür.
1960'lar	Rota kurma sezgiselleri sunulmuş ve 30-100 müşterili problemler çözülmüştür.
1970'ler:	İki fazlı sezgiseller, interaktif (insan-makine) sezgiseller geliştirilmiş, yaklaşık 50 müşterili problemler optimal metotlarla çözülebilir hale gelmiştir.
1980'ler	Matematiksel programlama esaslı prosedürler literatüre sunulmuştur. Etkileşimli (interaktif: insan-makine) sezgiseller geliştirilmiştir. Optimal yöntemler kullanılarak yaklaşık 50 müşteriye sahip olan bazı problemler çözülmüştür.
1990'lar	Araç rotalama problemlerine meta-sezgiseller uygulanmıştır. 50-100 müşteriye sahip bazı problemler optimal olarak çözülmüştür.
2000'ler	Meta-sezgiseller üzerine yapılan araştırmalar yoğunluk kazanmıştır. Birçok farklı meta-sezgisel ARP'lere uygulanmıştır. Melezleştirme teknikleri geliştirilmiş ve daha karmaşık ARP'ler üzerinde uygulamalar yapılmıştır.

Kaynak: Cricanic, Teodor G. Laporte, Gilbert. 1998. Fleet Management and Logistics. Kluwer Academic Publishers, USA, s:215.

1.3 Araç Rotalama Problem Tipleri

ARP'lerde, gerçek hayatta karşılaşılan karmaşık dağıtım sistemlerinin çözümlenmesi çok zor veya imkânsız olduğundan, bazı önemli kısıtların kullanıldığı, basitleştirilmiş problem tipleri üstünde çalışmalar yapılmaktadır. Günümüze kadar, literatürde birçok farklı tipte ARP incelenmiştir. Özellikle son yıllarda, ARP'ler ile ilgili araştırma sayıları arttıkça farklı problem tipleri de ortaya çıkmaktadır.

Bu çalışma kapsamında, sadece kısıt tiplerine göre deęişen araç rotalama problemleri incelenmiştir. Çalışma kapsamında geliştirilen algoritma, üzerinde en çok çalışılan ve ARP ailesinin en basit problemi olan Kapasite Kısıtlı Araç Rotalama Problemi'nin (KKARP) çözümde kullanılmıştır. Birçok ARP, KKARP'nin üzerine farklı kısıtlar eklenerek geliştirilmiştir. Kısıtlarına göre araç rotalama problemleri incelendiğinde, KKARP dışında temel problem tipleri; Zaman Pencereli Araç Rotalama Problemi (ZPARP), Dağıtım ve Toplamayı içeren Araç Rotalama Problemi (DTARP), ve Mesafe Kısıtlı Araç Rotalama Problemi (MKARP) olarak sıralanabilir (Toth ve Vigo 2000). KKARP Bölüm 2'de detaylı bir şekilde incelenmektedir.

1.3.1 Kapasite Kısıtlı Araç Rotalama Problemi

KKARP, ARP'lerin temelini oluşturan bir problem tipidir. Basit olarak, oluşturulacak rotaların araç kapasitelerini geçmemesi gerekmektedir. Bu problem tipinde, birbirinden farklı ve belirli taleplere sahip müşterilere, kapasiteleri sabit olan araçlar ile hizmet verilmelidir. KKARP, çalışmanın ileriki bölümlerinde detaylı olarak incelenmiştir (Bkz. Bölüm 2). Bu sebeple bu bölüm içerisinde incelenmesine gerek duyulmamıştır.

1.3.2 Mesafe Kısıtlı Araç Rotalama Problemi

Bu problem tipinde, rotalara atanan her aracın gidebileceği maksimum mesafe kısıtı bulunmaktadır. Bu mesafe kısıtları, araçların kapasite kısıtları ile birlikte kullanılabilirliği gibi, tek başlarına da kullanılabilirler.

Literatürde farklı tiplerde MKARP'ler bulunmaktadır. Örneğin farklı tipteki araçlar için, farklı mesafe kısıtları oluşturulabilir. Aynı şekilde, mesafe kısıtları yerine, toplam seyir süreleri de kısıt olarak alınabilir.

Basit olarak, her (i, j) müşteri çiftinin arasındaki arka negatif olmayan bir uzunluk, c_{ij} atanırsa, bir aracın rotasındaki tüm arkların toplam uzunluğu, maksimum tur uzunluğundan (L) büyük olmaması gerekmektedir. Farklı K aracın

kullanıldığı problemlerde, maksimum tur uzunlukları L_k ile gösterilebilir ($k = 1, \dots, K$). V ve K sırasıyla müşteri ve araç setini, x_{ij}^k (0-1 değişkeni) ise k . aracın kullandığı (i, j) arkını gösterdiği kabul edilirse, bu kısıt matematiksel olarak Denklem 1.1'deki gibi ifade edilebilir.

$$\sum_{i,j \in V} c_{ij} x_{ij}^k \leq L_k, \quad k \in K. \quad (1.1)$$

Her ARP'de olduğu gibi, bu problem tipinde de amaç, toplam maliyeti (uzunluk veya süre) en küçükmektir (Toth ve Vigo 2000).

1.3.3 Zaman Pencereci Araç Rotalama Problemi

KKARP'nin bir diğer türevi ise, zaman pencereci araç rotalama problemleridir (ZPARP). Pek çok dağıtım sisteminde, her müşteri, talep miktarının yanında bu dağıtımın yapılması gereken zaman aralığını da belirler. Bu zaman aralığına "Zaman Penceresi" denir. Bu tip problemlerde, müşteri i 'ye yapılacak olan hizmet, en erken (a_i) ve en geç (b_i) zamanlar arasında yapılmalıdır.

Zaman pencereci araç rotalama problemi şöyle tanımlanabilir (Hombberger ve Gehring 1999): N adet müşteriye, kapasiteleri sabit ve C olan araçlar ile hizmet verilmektedir. Müşteri talepleri ($d_i, i = 1, \dots, N$), sabit bir depodan ($i = 0$), müşteriler tarafından tanımlanan zaman penceresi $[a_i, b_i]$ içerisinde karşılanmalıdır. Amaç, toplam yolu veya süreyi en küçükleyerek uygun bir çözüm bulmaktır.

Aracın müşteriye geliş zamanı belirli bir zaman aralığı içerisinde olmalıdır. Eğer araç müşteriye daha erken ulaşmış ise, aracın en erken servis başlama zamanına (a_i) kadar beklemesi gerekmektedir. Eğer araç müşteriye en geç servis zamanından (b_i) daha geç giderse, bulunan çözüm uygun olmayan bir çözüm halini alır. Bazı ZPARP'lerde, hizmetin geç verilmesi durumunda, amaç fonksiyonuna ceza da eklenebilir. ZPARP'de, genellikle önce müşterilere istedikleri zaman aralığında hizmet vermek, sonra rotaların eniyilenmesi yoluyla çözüme ulaşılır (Erol 2006).

Lenstra ve R. Kan (1981) yaptıkları çalışma ile ZPARP'nin NP-zor karmaşıklığına sahip olduğunu kanıtlamışlardır. Başka bir deyişle, geçerli bir süre içerisinde kesin optimal çözüm bulmak çoğu zaman imkansızdır. ZPARP, KKARP'ye göre çok daha karmaşıktır (Solomon ve Desrosiers 1988). Bu yüzden araştırmacıların çoğu, sezgisel ve meta-sezgisel çözüm yöntemlerini kullanarak, yakın-optimal çözümler bulmaya çalışmışlardır. ZPARP'nin çözümünde kullanılan kesin çözüm yöntemlerinin detaylı incelenmesi için, Kallehauge'nin (2008) çalışmasını incelenebilir.

Literatürde, ZPARP'nin çözümünde birçok meta-sezgisel yöntem kullanılmıştır. Tabu arama (Potvin ve diğ. 1997, Taillard ve diğ. 1997, Chiang ve Russel 1997, Schulze ve Fahle 1999), tavlama benzetimi (Chiang ve Russell 1996), genetik algoritma (Thangiah ve diğ. 1991, Potvin ve Bengio, 1996), karınca kolonisi algoritması (Gambardella ve diğ. 1999) bu çalışmalardan bazılarıdır.

1.3.4 Dağıtım ve Toplamayı İçeren Araç Rotalama Problemi

Bu tip problemlerde, müşterilere dağıtılan malların bir kısmı depoya gönderilmesi amacıyla geri verilebilir. Dağıtım ve toplamayı içeren araç rotalama problemlerinde (DTARP), müşterilerin geri gönderecekleri malların araçlara sığabilmesi koşulu dikkate alınmalıdır. Bu kısıt, problemi planlamayı ve çözmeyi çok daha karmaşık hale getirmektedir. Toplama kısıtından dolayı araç kapasiteleri iyi kullanılamayabilir ve araç sayıları veya kat edilen toplam mesafe uzayabilir.

Problemi daha basite indirgeyebilmek için, müşteriler arasında mal akışının olmadığı kabul edilir. Yani, tüm mallar depodan dağıtılır ve müşteriler tarafından geri gönderilen mallar depoya geri döner. Başka bir basitleştirme işlemi, önce toplamanın, daha sonra dağıtımın olduğunu varsaymaktadır. Bu basitleştirmenin uygulanmadığı problemlere, Simültane Dağıtım ve Toplama Problemleri (SDTARP) denir.

DTARP üzerindeki çalışmalar 1980'li yıllarda başlamış ve günümüze kadar artarak devam etmiştir. Bu tip problemler üzerinde yapılan, detaylı bir literatür

taraması olarak Bodin ve diğ.'nin (1983) çalışması gösterilebilir. Toth ve Vigo (1997) ve Minzoggi ve diğ. (1999), problemin çözümü için farklı kesin çözüm metotları önermişlerdir. Araştırmacılar, DTARP'nin çözümünde daha çok meta-sezgisel metotlara yoğunlaşmışlardır. Bu çalışmalardan bazıları; kümeleme ve araya sokma algoritması (Salhi ve Nagy 1999), tabu arama (Brandao 2006), komşuluk arama (Gendreau ve diğ. 2006) ve "Unified Heuristic" adı ile geliştirilen sezgisel yöntem (Ropke ve Pisinger 2006) olarak sıralanabilir.

STDARP ilk olarak Min (1989) tarafından, kütüphane içi dağıtım ve toplama sistemlerinin çözümü için ortaya atılmıştır. STDARP'nin çözümü için; Nagy ve Salhi (2005), Dethloff (2001), Crispim ve Brandao (2005), Bianchessi ve Righini (2007), Montane ve Galvao (2006) gibi çalışmaları incelenebilir.

1.3.5 Diğer Araç Rotalama Problemleri

Önceki bölümlerde incelenen temel ARP'lerden başka, literatürde birçok farklı ARP mevcuttur. Fakat bu çalışma kapsamında detaylı bir şekilde incelenmelerine gerek duyulmamıştır. Bunlardan ilki; çok depolu araç rotalama problemleridir. Bu problem, ilk olarak Tillman ve diğ. (1971) tarafından 1970'lerin başlarında, çözümü için bir sezgisel önerilerek incelenmiştir. Bu problem tipinde bir depodan dağıtım yerine, birden çok depodan dağıtımın gerçekleştirilmesi göz önüne alınır. Literatürde problemin çözümü için farklı sezgisel ve meta-sezgisel yöntemler önerilmiştir (Cordeau 1997, Nagy ve Salhi 2005, Pisinger ve Ropke 2007).

Alan Bağımlı Araç Rotalama Problemlerinde, farklı tipte araçlar mevcut olup, her müşteri sadece bir tipteki araç ile hizmet almaktadır. Bu tip problemler, ilk olarak Nag ve diğ. (1988) tarafından 1980'lerin sonunda ortaya atılmıştır. Cordeu ve Laporte (2001) zaman kısıtlı ve alan bağımlı problemlerin çözümü için bir tabu arama algoritması önermişlerdir.

Bir diğer problem tipi ise, Açık Araç Rotalama Problemleridir. Bu tip problemlerde, rotaların depoda sonlanma koşulu ortadan kalkar. Depodan

bařlayan rotalar, en son uęranan müşteride son bulmaktadır. Bu tip problemler içinde farklı sezgisel yöntemler literatürde mevcuttur (Sarıkliş ve Powell 2000, Brandao 2004).

Son yıllarda üzerinde sıkça çalışılan bir dięer problem tipi ise, bir veya birden çok bileşenin rassal olarak oluşturulduęu Stokastik Araç Rotalama problemidir (SARP). Bu bileşenler; müşteriler, servis zamanları, yol uzunlukları veya müşteri talepleri olabilir. SARP üzerindeki arařtırmalar, 1990'lı yıllarla hız kazanmıştır. Literatürde birçok farklı çözüm yöntemi bulunmaktadır (Roberts ve Hadjiconstantinou 1998, Park ve Hong 2003). Detaylı kaynak arařtırmaları için; Gendreau ve dię. (1996) ve Yaohuang ve dię. (2002) çalışmaları incelenebilir.

2. Kapasite Kısıtlı Araç Rotalama Problemi

2.1 Problemin Tanımı

KKARP bir graf üzerinden şöyle tanımlanabilir: $G = (V, A)$ 'nin bütün bir graf olduğu varsayılarak, $V = \{0, \dots, N\}$ 'nin düğüm setini, A 'nın ise ark setini gösterdiği kabul edilir. Düğümler, $i = 1, \dots, N$ müşterileri, düğüm 0 ise depoyu gösterir. Negatif olmayan c_{ij} maliyeti, her $(i, j) \in A$ arkı ile ilişkilendirilir. Maliyet c_{ij} , düğüm i 'den düğüm j 'ye gidiş maliyetini gösterir ve (i, j) düğümleri arasındaki mesafeye eşit olduğu kabul edilir. Çoğu zaman, $c_{ii} = \infty$ tanımlanarak, (i, i) arkına izin verilmez. $(i, j) \in A$ 'dan oluşan maliyet matrislerinin simetrik olduğu ($c_{ij} = c_{ji}$) kabul edilir. Çoğu KKARP'de üçgen eşitsizlikleri sağlanır: $c_{ik} + c_{kj} \geq c_{ij}$, $\forall i, j, k \in V$. Bunlara ek olarak, eğer (i, j) düğüm çiftlerinin arasında ark yok ise, $c_{ij} = \infty$ alınarak direkt bağlantı kurulması da engellenmiş olur (Toth ve Vigo 2000). KKARP'nin temel notasyonunda, V , düğüm setini, d , müşteri talepleri vektörünü, k , toplam rota sayısını (toplam araç sayısı) ve C ise araç kapasitesini gösterir.

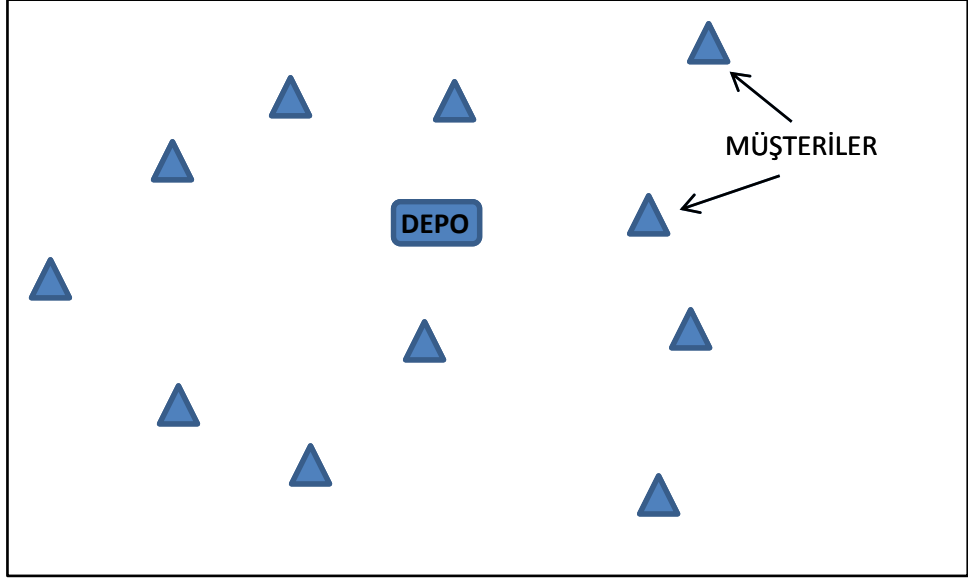
Detaylı incelendiğinde, KKARP'nin, iki karmaşık kombinatorik problemin birleşiminden oluştuğu görülmektedir. Eğer tek aracın kapasitesi tüm müşterilerin talep toplamlarına eşit ise, $C = \sum_{i \in V, i \neq 0} d_i$, KKARP Gezgin Satıcı Problemine (GSP) dönüşmektedir. Eğer tüm $(i, j) \in V$ yay maliyetleri 0 ise, $c_{ij} = 0, \forall i, j \in V$, KKARP Sırt Çantası Problemine dönüşmektedir.

KKARP için amaç fonksiyonu tanımlanacak olursa; G grafının içinde en düşük maliyete sahip öyle k adet rota (her araç için bir rota), bulunabilir ki:

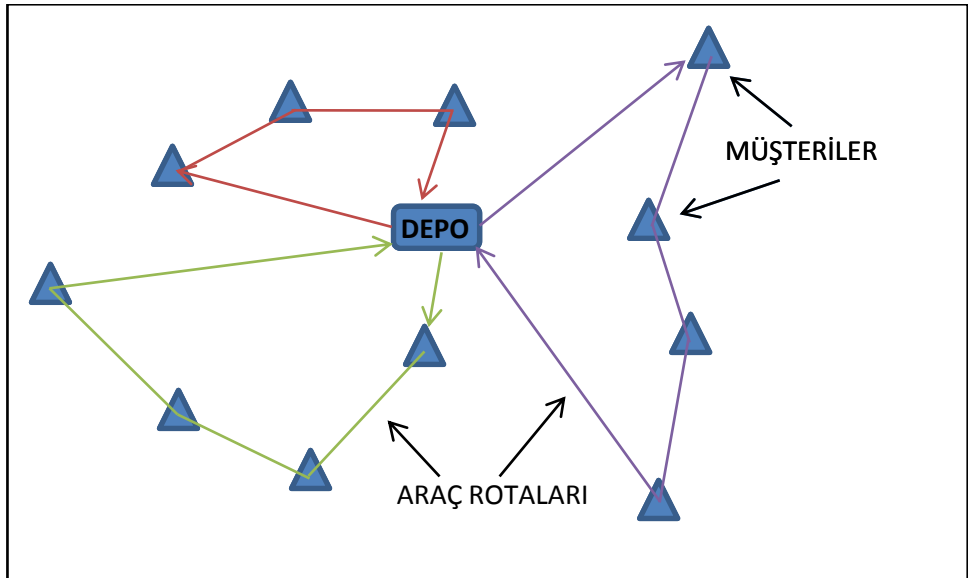
- (1) her müşteriye tek bir araç hizmet verecek,
- (2) her araç depodan çıkacak ve depoya dönecek,

(3) her bir rotada hizmet verilen müşterilerin toplam talepleri araç kapasitesi C 'den küçük veya eşit olacaktır.

Şekil 2.1'de, bir KKARP'de müşteri noktaları ve depo gösterilmiştir. Bu probleme ait örnek çözüm ise, Şekil 2.2'de görülebilir.



Şekil 2.1 Örnek bir KKARP.



Şekil 2.2 KKARP Çözüm Örneği.

2.2 Matematiksel Model

KKARP'ler için literatürde birçok farklı tamsayılı ve karışık tamsayılı programlama modeli bulunmaktadır. $V = \{0,1, \dots, N\}$ 'nin tüm düğüm setini ve $V_+ = \{1, 2, \dots, N\}$ 'nin pozitif düğüm setini tanımladığı, böyle bir matematiksel model aşağıda verilmiştir.

Karar Değişkenleri:

$$x_{ij} = \begin{cases} 1, & i' \text{den } j' \text{ye gidiş var ise} \\ 0, & \text{aksi takdirde.} \end{cases}, \quad i, j \in V, i \neq j.$$

u_i = Taşıma miktarı.

Parametreler:

c_{ij} i düğümünden j düğümüne gidiş maliyeti.

K Araç sayısı.

C Araç kapasitesi

d_i Müşteri i 'nin talep miktarı.

Amaç Fonksiyonu:

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} \quad (2.1)$$

Kısıtlar:

$$\sum_{i \in V_+} x_{ij} = 1, \quad j \in V_+, i \neq j. \quad (2.2)$$

$$\sum_{j \in V_+} x_{ij} = 1, \quad i \in V_+, i \neq j. \quad (2.3)$$

$$\sum_{i \in V} x_{i0} = K, \quad i \neq j. \quad (2.4)$$

$$\sum_{j \in V} x_{0j} = K, \quad i \neq j. \quad (2.5)$$

$$u_j - u_i + C(1 - x_{ij}) \geq d_j, \quad i, j \in V_+, i \neq j. \quad (2.6)$$

$$d_i \leq u_i \leq C, \quad i \in V_+. \quad (2.7)$$

$$x_{ij} \in \{0,1\}, \quad i, j \in V. \quad (2.8)$$

$$u_i \geq 0, \quad i \in V_+. \quad (2.9)$$

Kısıt 2.1 toplam maliyeti minimize edecek amaç fonksiyonunu göstermektedir. Kısıt 2.2 ve 2.3 bir düğüme sadece bir kere girilmesini ve bir kere çıkılmasını sağlar. Kısıt 2.4 ve 2.5 K adet aracın depodan çıkıp, depoya dönmesini sağlar. Kısıt 2.6 ve 2.7 ise akış kısıtlarıdır ve alt turların oluşumunu engeller. Son iki kısıt ise, karar değişkenlerinin kısıtlarıdır.

2.3 Çözüm Yöntemleri

KKARP'nin iki zor kombinatorik problemden oluştuğu Bölüm 2.2'de açıklanmıştır. Hem sırt çantası problemi, hem de gezgin satıcı problemi "NP-Zor" tipinde problemlerdir (Falkenauer 1996, Wolsey 1998). Sonuç olarak, KKARP'nin de "NP-Zor" tipinde karmaşıklığa sahip olduğu ve kabul edilebilir sürelerde optimal olarak çözülmesinin çok zor olduğu söylenebilir (Lawrence ve Bruce 1981, Lee ve diğ. 2003). Literatürde küçük veya orta büyüklüğe sahip problemler için kesin çözüm yöntemleri başarılı olarak uygulamış olsa da, son 10 yıldır araştırmacılar çoğunlukla büyük KKARP'lerin çözümünde sezgisel veya meta-sezgisel metotlardan faydalanmışlardır. Bu çalışmada, genel ağırlık meta-sezgiseller üzerinde olduğu için, kesin çözüm metotlarının detaylı olarak incelenmesine gerek duyulmamıştır.

2.3.1 Kesin Çözüm Yöntemleri

Kapasiteli araç rotalama problemleri için en başarılı kesin çözüm yöntemlerinden biri Fisher'in (1994) 71 müşterili KKARP'yi çözmeyi başarabilen K-Tree metodudur. Fisher'in, K-Tree algoritması dışında, birçok farklı kesin çözüm yöntemleri literatürde mevcuttur. Detaylı kaynak araştırmaları ve çözüm yöntemleri için Ball ve diğ. (1995), Laporte (1992, 1997), Toth ve Vigo (2002) çalışmaları incelenebilir. Bu yöntemlerden bazıları, dinamik programlama (Hadjiconstantinou ve diğ. 1996), sütun oluşturma ve küme bölme yaklaşımları (Agarwal ve diğ. 1989) olarak sıralanabilir.

2.3.2 Sezgisel Çözüm Yöntemleri

KKARP çözümünde arařtırmalar özellikle 1990'lı yıllardan sonra sezgisel ve meta-sezgisel yöntemlerin kullanımına yoğunlařmıştır. Sezgisel metotlar genel olarak miyop metotlar olarak adlandırılır. Bunun sebebi, iterasyonun her zaman daha iyi çözüme doğru ilerlemesidir ve kötü olan çözüm hiçbir zaman yeni çözüm olarak kabul edilmez. Sezgisel yöntemlerde global en iyi çözüme gitmek yerine, çoğu zaman yerel en iyi çözümlerde kalınır. Fakat, sezgisel yöntemler kısa işlem sürelerinde yakın-optimal sonuçlar verebilirler. Sezgisel yöntemler genel olarak 3 grupta incelenebilir (Laporte ve Semet 2000):

- (1) Tur kurucu metotlar,
- (2) Tur geliştirici metotlar,
- (3) İki-ařamalı metotlar.

2.3.2.1 Tur Kurucu Metotlar

Tur kurucu metotlar genel olarak, her ařamada toplam maliyeti en küçükleyecek şekilde arkların birbirine eklenmesi sonucu araç rotalarının oluşturulmasını kapsar. Bu tip yöntemlerden en bilinen Tasarruf (Savings) (Clarke ve Wright 1964) algoritmasıdır.

Tasarruf algoritması bir başlangıç noktasının depo olarak ele alınmasıyla başlar. Başlangıçta tüm durakların depodan ziyaret edildiđi varsayılır. Daha sonra, iki alt arkın birleřtirilmesiyle elde edilen tasarruflar hesaplanır. En büyük tasarruf deđerinden başlayarak arklar birleřtirilir. Tasarruf algoritmasının birçok farklı çeřiti mevcuttur (Desrochers ve Verhoog 1989, Altinkemer ve Gavish 1991).

2.3.2.2 İki Ařamalı Metotlar

İki-ařamalı yöntemler genel olarak, ilk etapta müşterilerin kümelenmesini ve daha sonra rotaların oluşturulmasını kapsar (önce kümele, sonra rotala)

(<http://neo.lcc.uma.es/radi-aeb/WebVRP/>, Eriřim: 2009). Bu tipteki yntemlerin en nemlileri, Fisher ve Jaikumar Algoritması (Fisher ve Jaikumar 1981), Petal Algoritması (Ryan ve diđ. 1993), Sprme (Sweep) Algoritması (Gillet ve Miller 1974) ve Taillard Algoritması (Taillard 1993) olarak sıralanabilir.

2.3.2.3 Tur Geliřtirici Metotlar

Tur geliřtirici yntemlerin temel amacı, uygun olan bir zm zerinde, arkların veya turlar arasında mřterilerin yer deđiřtirilmesinin sađlanması ve daha iyi zmler elde edilmesi olarak aıklanabilir (<http://neo.lcc.uma.es/radi-aeb/WebVRP/>, Eriřim: 2009). Thompson ve Psaraftis (1993), Van Breedam (1994) ve Kinderwater ve Savelsbergh (1997) bu tipteki yntemleri ieren alıřmalardır.

2.3.3 Meta-Sezgisel Yöntemler

Meta-sezgisel yöntemler sezgisel yöntemlere göre çok daha özelleşmiş ve daha zeki bir şekilde global en iyiyi arayan metotlardır. Ancak, işlem sürelerinin tam olarak kestirilmesi güçtür. Bu yöntemler, global en iyi çözüme giderken sezgisel yöntemlere göre daha çok süre harcaabilirler (Gendreau ve diğ. 1999). KKARP'lerin çözümü için araştırmacılar birçok meta-sezgiseller üzerine araştırmalar yapmışlardır. Tabu arama (TA), tavlama benzetimi (TB), genetik algoritmalar (GA), karınca kolonisi algoritması (KKA) ve parçacık sürüsü eniyilemesi (PSO) KKARP'lerin çözümünde kullanılan temel ve kabul edilmiş meta-sezgisellerdir.

TA ve TB komşuluk stratejisi kullanarak ilerleyen yöntemlerdir. Başka bir deyişle, bunlar popülasyon tabanlı olmayıp, yeni çözümlerin bir önceki çözümden üretildiği mekanizmalara sahiptir. GA, her ne kadar popülasyon tabanlı olsa da, çaprazlama ve mutasyon gibi bir veya iki bireyin kullanıldığı operatörler, bir sonraki jenerasyona aktarılan bilgilerin kısıtlı bir yapıdan gelmesine sebep olurlar. GA'nın getirdiği en önemli yenilik, yeni bireylerin seçiminde popülasyondaki tüm bireylerin bilgilerine başvurulmasıdır. KKA ve PSO ise, bu çalışmada kullanılan Elektromanyetik Algoritma'nın mekanizmasına benzer meta-sezgiseller olarak sınıflandırılabilir. Bu yöntemler de yeni çözümler bir önceki jenerasyondaki tüm bireylerin bilgilerine başvurularak oluşturulur. Böylece, daha çok bilgi kullanılarak oluşturulan yeni jenerasyonlar, daha geniş bir alanda arama yaparken, algoritmanın global optimum noktasına gidişini de hızlandırır.

2.3.3.1 Tabu Arama

TA tekniğine ait ilk çalışmalar Glover'a (1986, 1989, 1990) aittir. TA yerel en iyinin ötesindeki çözüm uzayını keşfetmek için yerel bir sezgisel arama prosedürüne kılavuzluk eden bir meta-sezgiseldir. TA bu kılavuzluğu, hafızasında aramanın hikâyesini tutarak, yani bazı adaylara "yasak" koyarak, aramayı sınırlandırması ve yerel en iyiden kurtarmasıyla yerine getirmektedir.

TA'nın işleyişi basitçe şöyledir: Bir başlangıç çözümü ele alınır, bu seçimin komşuları bir komşuluk stratejisi ile belirlenir, bir amaç fonksiyonuna göre komşular değerlendirilir. Değerlendirilen komşu tabu listesinde yoksa, bu çözüm şimdiye kadar bulunan en iyi çözümle karşılaştırılır. Eğer çözüm en iyi çözümden daha iyi ise sonraki arama için yeni başlangıç çözümü olarak alınır ve tabu listesine eklenir. Bunlara ek olarak, bulunan komşu çözüm tabu listesinde olmasına rağmen en iyi çözümden daha iyi ise, belirli bir kritere göre kabul edilir. Bu kritere aspirasyon kriteri denir. Bu kriterle kabul edilen bir hareketin tabu olma durumu kaldırılmaz ve hareket tabu listesinden çıkarılmaz. Aksi durumda, iterasyonlara en son çözümden devam edilir (Şeker 2007).

KKARP'lerin çözümü için birçok farklı stratejilere ve mekanizmalara sahip TA uygulamaları bulunmaktadır (Gendreau ve diğ. 1999, Osman 1993). Daha detaylı kaynak taraması için Gendreau ve diğ.'nin (1999) çalışması incelenebilir.

2.3.3.2 Tavlama Benzetimi

Tavlama benzetimi ismi demirin tavlansından alınmıştır. Demir çok yüksek sıcaklıklarda çok yüksek enerjilere sahip olabilen bir metaldir. Bununla birlikte işlenen demirin sıcaklığı nispeten düşük olmasına rağmen enerjisi yüksek durumlarla karşılaşılabilir (Kirkpatrick ve diğ. 1983, Cerny 1985). Temel olarak, TB demirin bu özelliğinden faydalanarak en iyiye gitmeyi hedefler. TB'de TA'daki gibi, yeni çözümler, komşuluk oluşturma stratejisi kullanılarak elde edilir. Fakat bu algorithmada yasak listesi yerine, stokastik bir yapı kullanılır.

Bir en küçükleme problemi için, TB basitçe aşağıda açıklanan mantıkta çalışmaktadır: Mevcut çözüm X iken komşuluk stratejisi ile üretilen komşu çözüm X' olarak alınırsa ve eğer $delta, \Delta = f(X) - f(X')$ ise X' yeni çözüm ($X = X'$) olarak kabul edilir. Burada $f(X)$, amaç fonksiyonunu ifade eder. Eğer delta sıfırdan büyük ise ($\Delta > 0$), X' , $e^{-\Delta/T} \leq 0.5$ olduğu durumlarda kabul edilir, diğer durumlarda ise yeni çözüm olarak kabul edilmez. T , sıcaklık parametresi olup, her iterasyonda belirli bir oranda düşürülür. Birçok araştırmacı TB'nin farklı türlerini

kullanarak KKARP için çözüm üretmeye çalışmışlardır (Osman 1993, Tavakkoli - Moghaddam ve diğ. 2006).

2.3.3.3 Karınca Kolonisi Algoritması

Karıncalar, yuvalarından bir gıda kaynağına giden en kısa yolu, herhangi görsel ipucu kullanmadan bulma yetisine sahiptirler. Ayrıca, çevrelerindeki değişikliklere çok iyi uyum sağlama özellikleri vardır. Örnek vermek gerekirse, bir gıdaya giden yolda herhangi bir problem meydana gelmesi (bir engelin ortaya çıkması gibi) ve yolun kullanılamaz olması durumunda, yeniden en kısa yolu bulurlar (Şeker 2007).

Karıncaların bu yolu bulmak için kullandıkları araç feromonlardır (pheromone). Feromon, bazı hayvanların kendi cinslerinden diğer hayvanları etkilemek için kullandıkları bir tür kimyasal salgıdır. Karıncalar ilerlerken, belirli bir miktar feromon depo ederler ve olasılığa dayanan bir yöntemle fenomenin daha çok olduğu yolu az olduğu yola tercih ederler.

KKARP için ilk karınca kolonisi algoritması 1997 yılında Bullnheimer tarafından ortaya atılmıştır. Bullnheimer'in tasarımında iki temel karınca sistemi adımı vardır: (1) Araç rotalarının oluşturulması, (2) iz güncellemesi. Algoritma kurulduktan sonra bu iki adım her iterasyon için tekrarlanır. Yapay karıncaların ilk yerleşimleri dikkate alındığında, iterasyon başlangıcında, her müşteriye karınca sayısının eşit olması gerektiği sonucuna ulaşılmıştır. Bullnheimer'in çalışması dışında, KKARP için farklı KKA araştırmaları da vardır (Hartl ve Strauss 1999 Yu B. ve diğ. 2008, Doerner ve diğ. 2002).

2.3.3.4 Genetik Algoritma

Genetik algoritmaların temel ilkeleri ilk kez Michigan Üniversitesi'nde John Holland tarafından ortaya atılmıştır. Holland 1975 yılında yaptığı çalışmaları "Adaptation Natural and Artificial Systems" adlı kitabında bir araya getirmiştir. Genetik algoritmalar doğada geçerli olan en iyinin yaşaması kuralına dayanarak

sürekli iyileşen çözümler üretir. Bunun için “iyi”nin ne olduğunu belirleyen bir uygunluk fonksiyonu ve yeni çözümler üretmek için yeniden kopyalama, değiştirme çaprazlama gibi operatörler kullanılır (Goldberg 1989). Genetik algoritmaların bir diğer önemli özelliği de grup çözümlerle uğraşmasıdır. Bu sayede çok sayıda çözümün içinden iyileri seçilip kötülerini elenebilir. KKARP üzerine yapılmış birçok GA çalışması literatürde mevcuttur. Bunların en önemlileri; Baker ve Ayechev (2003), Osman ve diğ. (2005) ve Prins (2004) olarak sıralanabilir.

2.3.3.5 Parçacık Sürüsü Optimizasyonu

Parçacık sürüsü optimizasyonu (PSO) ilk olarak, Kennedy ve Eberhart (1995) tarafından ortaya atılmıştır. PSO, bir grup organizmanın davranışlarını taklit eden, popülasyon tabanlı bir arama yöntemidir. PSO’da, bir çözüm çok boyutlu bir düzlemde bir nokta ile gösterilir. Her iterasyonda, her organizmanın pozisyonu, organizmanın hızı ile bağlantılı bir şekilde güncellenmektedir. Böylece, tüm sürünün global bir en iyi noktasına doğru hareket edeceği öngörülmektedir. PSO mekanizmasının detaylı incelemesi ve uygulamaları Kennedy ve Eberhart (2001) ve Clerc (2006)’in çalışmalarında incelenebilir. PSO, KKARP’ye son yıllarda uygulanmaya başlanmıştır (Chen ve diğ. 2006, Ai ve Kachitvichyanukul 2007, 2009).

3. Elektromanyetik Algoritma

3.1 Giriş

Elektromanyetik Algoritma (EMA) ilk olarak Birbil ve Fang (2003) tarafından sınırlı karar değişkenleri kullanılarak, sürekli eniyileme problemlerinin çözümünde kullanılmıştır. EMA, diğer temel meta-sezgisel algoritmalara kıyasla oldukça yeni sayılabilecek, popülasyon tabanlı bir meta-sezgisel yöntemdir. EMA mekanizması, yüklü parçacıkların bir elektrik alan içerisinde, birbirlerine uyguladıkları itme/çekme kuvvetlerini esas alır. Algoritmada, her parçacık bir çözümü gösterir ve belirli bir miktarda elektrik yükü taşır. Her parçacığın taşıdığı yük, o parçacığın gösterdiği çözümün kalitesi ile doğru orantılıdır. Başka bir deyişle, parçacığın gösterdiği çözüm ne kadar iyi ise, yükü de o kadar fazla olacaktır.

EMA'da çözümler, her parçacığın çok boyutlu bir düzlemdeki pozisyonunu gösteren, "pozisyon vektörleri" ile tanımlanır. Parçacığın gösterdiği çözümün değeri de, o parçacığın pozisyon vektörü kullanılarak hesaplanır. EMA'da her parçacık, diğer popülasyon bireyleri üzerine itme veya çekme kuvveti uygular. Bir parçacık üzerine uygulanan kuvvetlerin bileşkesinden faydalanılarak, parçacık düzlem içerisinde hareket ettirilir. EMA mekanizmasındaki temel felsefe; bu uygulanan bileşke kuvvetler ile, parçacıkların bir global en iyi noktaya doğru hareket ettirilmesidir. EMA'da; GA, TA ve TB gibi geleneksel meta-sezgisel yöntemlerde olduğu gibi popülasyon bireylerinin birbirleri arasında bilgi alışverişinde bulunmaları yerine, PSO veya KKA'daki gibi, bir popülasyon bireyi, popülasyondaki tüm bireylerin bilgilerinden faydalanır.

EMA yeni bir meta-sezgisel yöntem olduğu için, uygulama açısından kısa bir geçmişe sahiptir. EMA, geçmişte bulanık mantık ilişkilerinin çözümünde (Birbil ve

Feyzioglu 2003) ve yapay sinir ağıları eğitiminde (Wu ve diğ. 2004) kullanılmıştır. Debels ve diğ. (2006) EMA ile yerel bir arama metodunu birleştirerek kaynak kısıtlı çizelgeleme problemlerine çözüm aramışlardır. Tsou ve Kao (2007), EMA'yı çok amaçlı eniyileme problemlerinin çözümde kullanmışlardır. Maenhout ve Vanhoucke(2007), hemşire çizelgeleme problemine EMA'yı kullanarak çözüm getirmişlerdir. Yakın zamanda EMA, diğer mühendislik alanlarındaki eniyileme problemlerinin çözümünde de kullanılmaya başlanmıştır (Jhang ve Lee 2008). Chang ve diğ. (2008), EMA'yı genetik algoritma ile melezleyerek, tek makineli erken bitme/geç kalma çizelgeleme probleminin çözümünde kullanmışlardır. Aynı zamanda, bu çalışmada "rassal anahtar" yöntemini kullanarak, pozisyon vektörlerinin uygun birer iş sırasını göstermesini sağlamışlardır. Wu ve diğ. (2007) rassal anahtar yöntemi ile EMA'yı birleştirerek gezgin satıcı problemlerinin etkin bir şekilde çözümlemeyi başarmışlardır. EMA ilk kez kapasite kısıtlı araç rotalama problemlerinin çözümünde bu çalışma kapsamında kullanılmıştır.

Bu çalışma kapsamında, melezleştirilmiş KKARP'lerin çözümü için melezleştirilmiş EMA (HEMA) yaklaşımı benimsenerek, bu çerçevede güncellenmiş bir amaç fonksiyonu hesaplama (AFH) prosedürü kullanılmıştır. Bu AFH prosedürü kapsamında, rassal anahtar (RA) yöntemi algoritmaya uyarlanarak, EMA; KKARP gibi kesikli değişkenlerden oluşan eniyileme problemlerini çözebilecek hale getirilmiştir. Reel sayılar ile kodlanmış pozisyon vektörlerinin, RA ile çözümlenerek, uygun birer rota setini göstermeleri sağlanmıştır.

RA yaklaşımı ilk olarak Bean (1994) tarafından, gerçel sayılarla kodlanmış bir popülasyonda genetik algoritma ile bir sıralama probleminin çözümünde kullanılmıştır. Synder ve Daskin (2006), GA'da çaprazlama sonucunda ortaya çıkan uygun olmayan çözümlerden kurtulmak için RA'yı kullanmışlardır. Valente ve Goncelvez (2009) yine RA'yı GA ile birlikte kullanarak tek makineli iş sıralama ve çizelgeleme problemine çözüm aramışlardır. Günümüze kadar birçok araştırmacı birçok eniyileme probleminin çözümünde RA'dan faydalanmışlardır (Norman ve Bean 2000, Mendes ve diğ. 2005, Tasgetiren ve diğ. 2007).

Çok boyutlu bir düzlemde, yüklü parçacıkların birbirlerine uyguladıkları itme/çekme kuvvetleri, EMA yaklaşımının arkasındaki temel felsefeyi oluşturur. Her yeni iterasyonda, parçacıkların bu düzlem üzerindeki yeni pozisyonları, o parçacığın üzerine etki eden kuvvetlerin bileşkesi kullanılarak güncellenir. Elektromanyetik teorisine göre, bir elektrik yüklü parçacığın diğer yüklü parçacığa uyguladığı kuvvet, parçacıkların arasındaki mesafe ile ters orantılı, yüklerinin çarpımları ile doğru orantılıdır. EMA yönteminde, her parçacık bir çözüm gösterir ve parçacık i 'nin elektrik yükü, o parçacığın amaç fonksiyonu değeri $f(x_i)$ kullanılarak hesaplanır. Parçacığın amaç fonksiyon değeri ise, doğrudan parçacığın pozisyon vektörü ile bağlantılıdır. Parçacık i 'nin yükü q_i , aşağıdaki denklem kullanılarak hesaplanır.

$$q_i = \exp\left(-n \frac{f(\mathbf{x}_i) - f(\mathbf{x}_{best})}{\sum_{j=1}^m [f(\mathbf{x}_j) - f(\mathbf{x}_{best})]}\right), \forall i. \quad (3.1)$$

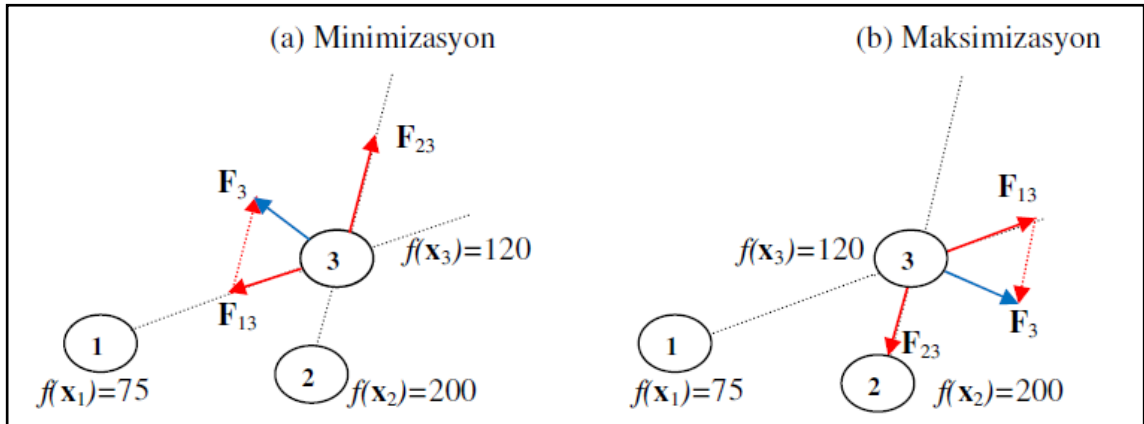
Denklem (3.1)'de $f(\mathbf{x}_{best})$, popülasyondaki en iyi çözüm değerini, m popülasyon büyüklüğünü ve n pozisyon vektörünün uzunluğunu gösterir. Burada n , elektrik yükü değerlerini düşürerek, işlem hacmini azaltmak amacı ile kullanılır. Denklem 3.1, bir em küçükleme problemi için verilmiştir. Başka bir deyişle, $\exp()$ operatörü nedeni ile, parçacık i 'nin yükü her zaman pozitif değerler alacaktır. Parçacık i 'nin yükü, parçacık i 'nin gösterdiği çözüm değerinin, en iyi çözüme yakınlığı ile ters, popülasyondaki çözüm değerleri arasındaki varyasyonun büyüklüğü ile doğru orantılıdır.

EMA, ilk olarak sürekli fonksiyonların eniyilenmesi için kullanıldığından, pozisyon vektörleri ile gösterilen koordinat değerleri doğrudan amaç fonksiyonu hesaplamalarında kullanılmıştır. Parçacık i ve onun amaç fonksiyonu hesaplamalarında kullanılan pozisyon vektörü bileşenleri, $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5]$ Şekil 3.1'de örnek olarak gösterilmiştir. Şekil 3.1'de k , bileşen indisini (bit numarası), bileşen değeri ise, sınırlı değişkenlerin değerlerini göstermektedir.

Bileşen indisi, k	1	2	3	4	5
k . bileşen değışkeni	x_i^1	x_i^2	x_i^3	x_i^4	x_i^5
k . bileşen değeri	0.23	0.45	1.55	3.42	-4.10

Şekil 3.1 Parçacık i ve onun Pozisyon Vektörü Bileşenleri

Daha önce belirtildiği gibi EMA'da yüklere herhangi bir işaret verilmez. Uygulanacak kuvvetin itme veya çekme olacağına, kuvvet hesaplama adımında karar verilir. EMA yaklaşımının en önemli noktası, parçacıkların, kendinden iyi çözümler gösteren bir kısım parçacıklar üzerinde itme, kendinden kötü çözüm gösteren diğer parçacıklar üzerinde ise çekme kuvveti uygulamasıdır. Şekil 3.2-(a)'da, bir en küçükleme problemi için, parçacık 3 üstüne uygulanan bileşke kuvvet (\mathbf{F}_3) gösterilmiştir. $f(x_1) < f(x_3)$ olduğundan, parçacık 1, parçacık 3'ü çeker. Diğer taraftan, parçacık 2 ile karşılaştırıldığında, parçacık 3 daha iyi bir çözümü gösterdiğinden ($f(x_3) < f(x_2)$), parçacık 2 parçacık 3'ü iter. Sonuç olarak, parçacık 3, bileşke kuvvet \mathbf{F}_3 doğrultusunda hareket eder. Şekil 3.2-(b)'de verilen en büyükleme probleminde ise, bu durumun tam tersi geçerlidir.



Şekil 3.2 Parçacıkların \mathbf{F} ile hareket ettirilmesi

Parçacık i üzerinde ki bileşke kuvvet \mathbf{F}_i Denklem 3.2'deki gibi hesaplanır.

$$\mathbf{F}_i = \sum_{j=1, j \neq i}^m \begin{cases} (\mathbf{x}_j - \mathbf{x}_i) \frac{q_i q_j}{\|\mathbf{x}_j - \mathbf{x}_i\|^2}, & \text{eğer } f(\mathbf{x}_j) < f(\mathbf{x}_i) \\ (\mathbf{x}_i - \mathbf{x}_j) \frac{q_i q_j}{\|\mathbf{x}_j - \mathbf{x}_i\|^2}, & \text{eğer } f(\mathbf{x}_j) > f(\mathbf{x}_i) \end{cases} \quad \forall i. \quad (3.2)$$

Denklem 3.2'de m , popülasyon büyüklüğünü göstermektedir. Denklem 3.2'de görüldüğü gibi \mathbf{F}_i , parçacıklar arasındaki mesafe ile ters orantılı ve parçacıkların yükleri çarpımları ile doğru orantılıdır. Parçacık yükleri de, parçacıkların gösterdiği çözümler ile doğru orantılı olduğundan $f(x_i)$ ile $f(x_j)$ arasındaki fark ne kadar büyük olursa parçacık i 'ye o kadar büyük bir bileşke kuvvet (\mathbf{F}_i) uygulanır.

EMA mekanizmasında, her parçacığın pozisyon vektörünün bileşen değerleri için üst ve alt sınırlar belirlenir. Başka bir deyişle, k . koordinat değerine (karar değişkenine) bir üst sınır (U^k) ve bir alt sınır (L^k) tanımlanır. Bu sınırlama aynı zamanda, çok boyutlu çözüm uzayının genişlemesini de engelleyerek, işlem hacminin büyümesini engeller. Bir parçacığın üzerine etki edecek bileşke kuvvet hesaplandıktan sonra, parçacık bileşke kuvvetin doğrultusunda Denklem 3.3'de gösterildiği gibi hareket ettirilir. Denklem 3.3'te λ daha önce gidilmeyen noktalara da gidilmesini sağlayan 0-1 arası rassal bir değişken ($\lambda \in \mathbb{R}: 0 \leq \lambda \leq 1$), *RNG* ise bir vektördür ve bileşenleri, \mathbf{x}_i pozisyon vektörünün her k . bileşeni için belirlenen üst ve alt sınırlar içerisinde yapılabilecek hareketi belirler. Denklem 3.4 bu sınırlamayı özetlemektedir.

$$\mathbf{x}_i = \mathbf{x}_i + \lambda \frac{\mathbf{F}_i}{\|\mathbf{F}_i\|} \text{ (RNG)} \quad (3.3)$$

$$x_i^k \in \mathbb{R}: \quad U^k \leq x_i^k \leq L^k, \quad k = 1, \dots, n. \quad (3.4)$$

EMA mekanizmasının temel adımları Şekil 3.3'te özetlenmiştir.

```

1:Başlangıç (pop)
2:while (Durdurma kriteri sağlanmıyorken) do
3:   f_Hesapla (pop)
4:   q_Hesapla (pop)
5:   F_Hesapla (pop)
6:   Hareket (pop)
7:   S_Kontrol (pop)
8:end while

```

Şekil 3.3 Elektromanyetik Algoritma'nın Temel Adımları

Şekil 3.3'te, pop popülasyon bireylerini, $f_Hesapla()$ fonksiyonu parçacıkların amaç fonksiyonunun, $q_Hesapla()$ ve $F_Hesapla()$ ise sırasıyla, yük ve bileşke kuvvetlerinin hesaplamalarını gösterir. $Başlangıç()$, başlangıç çözümlerini gösteren pozisyon vektörlerini belirlenen (U^k, L^k) sınırları içinde kalmak üzere rassal olarak oluşturur. $Başlangıç()$, Şekil 3.4'te özetlenmiştir. Şekil 3.4'te m popülasyon büyüklüğünü, n ise pozisyon vektörünün uzunluğunu gösterir. Pozisyon vektörünün uzunluğu, problem tipine göre belirlenir. Örnek olarak, n -değişkenli problem için, n -boyutlu bir uzay kullanılır. Başka bir deyişle, pozisyon vektörünün uzunluğu n olarak belirlenir. $S_Kontrol$ fonksiyonu ise, belirlenen sınırların dışına çıkan bileşenlerin sınır değerlerine eşitlenmesi sağlar (bkz. Şekil 3.3, Satır 7). $S_Kontrol$ fonksiyonu detaylı olarak Şekil 3.5'te açıklanmaktadır.

```

1: for i = 1 to m do
2:   for k = 1 to n do
3:      $\lambda \leftarrow \text{Uniform}(0,1)$ 
4:      $x_i^k \leftarrow L + \lambda (U^k - L^k)$ 
5:   end for
6: end for

```

Şekil 3.4 Başlangıç Fonksiyonu Sözde Kodu

```

1: for i=1 to m do
2:   for k=1 to n do
3:     if  $x_i^k > U^k$  then
4:        $x_i^k = U^k$ 
5:     elseif  $x_i^k < L^k$  then
6:        $x_i^k = L^k$ 
7:     end if
8:   end for
9: end for

```

Şekil 3.5 S_Kontrol Fonksiyonu Sözde Kodu

3.3 Araç Rotalama Problemleri için EMA

3.3.1 Yük, Kuvvet ve Hareket Değerlerinin Hesaplanması

Bir önceki bölümde de detaylı bir şekilde açıklandığı gibi EMA mekanizması temel olarak 3 adımdan oluşmaktadır:

- (1) Yük hesaplaması,
- (2) Kuvvet hesaplaması
- (3) Parçacıkların bileşke kuvvet doğrultusunda hareket ettirilmesi.

Bu çalışma kapsamında; yük ve kuvvet hesaplamaları, parçacıkların hareket ettirilmesi mekanizmaları, Debels ve diğ. (2006) tarafından geliştirilmiş EMA mekanizmasını temel almıştır. İlk olarak, parçacık i 'nin yükü, parçacık j 'ye göre hesaplanır. Denklem 3.5'te parçacık i 'nin yük hesaplaması verilmiştir. Denklem 3.5'te görüldüğü gibi, parçacık i 'nin parçacık j 'ye göre hesaplanan yükü, pozitif veya negatif olabilir. Başka bir deyişle, temel EMA mekanizmasının tersine (Bkz. Denklem 3.1) bu çalışmada kullanılan yöntemde yükler işaretlidir. Denklem 3.5'te, $f(\mathbf{x}_{worst})$ ve $f(\mathbf{x}_{best})$, sırasıyla popülasyonun en kötü ve en iyi amaç fonksiyonu değerlerini göstermektedir.

$$q_{ij} = \frac{f(\mathbf{x}_i) - f(\mathbf{x}_j)}{f(\mathbf{x}_{worst}) - f(\mathbf{x}_{best})}, \quad \forall i, \forall j, i \neq j. \quad (3.5)$$

Parçacık j 'nin parçacık i 'ye uyguladığı kuvvet Denklem 3.5'teki gibi hesaplanmaktadır.

$$\mathbf{F}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)q_{ij}, \quad \forall i, \forall j, i \neq j. \quad (3.6)$$

Parçacık i ve parçacık j 'nin amaç fonksiyonlarının eşit olduğu durumlarda, Denklem 3.5' e göre $q_{ij} = 0$ olacağından, parçacık j tarafından parçacık i üzerine uygulanan kuvvet de, $\mathbf{F}_{ij} = 0$ olacaktır. En küçükleme problemi için, eğer $\mathbf{x}_i, \mathbf{x}_j$ 'den daha iyi bir çözüm gösteriyorsa ($f(x_i) < f(x_j)$), q_{ij} negatif olacağından, \mathbf{F}_{ij} de negatif olacaktır ($\mathbf{F}_{ij} < 0$). Başka bir deyişle, eğer parçacık j , parçacık i 'den daha iyi bir çözüm gösteriyorsa, parçacık j parçacık i 'yi kendine doğru çekecektir. $f(x_i) > f(x_j)$ olduğu durumlarda ise tam tersi geçerli olacaktır. Parçacıkların üzerlerine etki eden bileşke kuvvet doğrultusunda hareket ettirilmesi ise Denklem 3.7 ile gerçekleştirilir.

$$\mathbf{x}_i = \mathbf{x}_i + \frac{\sum_j \mathbf{F}_{ij}}{iter}, \quad \forall i, i \neq j. \quad (3.7)$$

Denklem 3.7'de $iter$, iterasyon sayısını ifade etmektedir. Bileşke kuvveti (\mathbf{F}) iterasyon sayısına bölünerek, iterasyon sayısı ilerledikçe, hareketin kısıtlanması hedeflenmiştir. Böylece, aynen tavlama benzetimi algoritmasındaki sıcaklık düşürülmesi gibi, iterasyonlar ilerledikçe, aramanın global en iyi nokta etrafında kalması öngörülmüştür.

Burada, not edilmesi gereken önemli bir nokta ise, Denklem 3.5 ve 3.6'nın sadece i ve j parçacık çifti için verilmiş olmasıdır. Başka bir deyişle, sadece parçacık j 'nin, parçacık i üzerindeki etkisini göstermektedir. Birey sayısı ikiden fazla olan popülasyonlarda, bu işlemler, Denklem 3.7'de olduğu gibi, tüm i, j çiftleri için gerçekleştirilerek parçacık i 'nin hareketi sağlanır. Şekil 3.6'da verilen sözde kod,

parçacık i 'nin hareket etmesini sağlayacak \mathbf{F} bileşke kuvvetinin parçacıklar arası yük değerleri kullanılarak nasıl hesaplandığını göstermektedir.

```

1: for i = 1 to m do
2:    $\mathbf{F}_i = 0$ 
3:   for j = 1 to m do
4:     if (i  $\neq$  j) then
5:        $q_{ij} = \frac{f(x_i) - f(x_j)}{f(x_{\text{worst}}) - f(x_{\text{best}})}$ 
6:        $\mathbf{F}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)q_{ij}$ 
7:        $\mathbf{F}_i = \mathbf{F}_i + \mathbf{F}_{ij}$ 
8:     end if
9:   end for
10:   $\mathbf{x}_i = \mathbf{x}_i + \frac{\mathbf{F}_i}{\text{iter}}$ 
11: end for

```

Şekil 3.6 Kuvvet, Yük ve Hareket Değerlerinin Hesaplanması Sözde Kodu

Geleneksel EMA mekanizmasında, arama uzayını, n -boyutlu uzay içerisinde sınırlı tutmak için, pozisyon vektörünün her k . bileşeni için bir üst sınır (U^k) ve bir alt sınır (L^k) belirlenir. Geleneksel EMA mekanizmasının tersine, önerilen EMA'da tüm bileşenler için ortak bir üst sınır (U) ve alt sınır (L) belirlenmiştir. Başka bir deyişle, tüm boyutlar için $U^k = U$ ve $L^k = L$ olarak alınır. Bu sebeple, $S_{kontrol}$ mekanizması Şekil 3.7'ta görüldüğü gibi güncellenmiştir.

```

1: for i = 1 to m do
2:   for k = 1 to n do
3:     if  $x_i^k > U$  then
4:        $x_i^k = U$ 
5:     elseif  $x_i^k < L$  then
6:        $x_i^k = L$ 
7:     end if
8:   end for
9: end for

```

Şekil 3.7 Güncellenmiş S_Kontrol Mekanizması Sözde Kodu

Parçacıkların pozisyon vektörleri (\mathbf{x}_i), önceden belirlenen alt sınır (L) ve üst sınır (U) arasında rassal şekilde oluşturulur (bkz. Şekil 3.8).

```

1: for i = 1 to m do
2:   for k = 1 to n do
3:      $\lambda = \text{Uniform}(0,1)$ 
4:      $x_i^k = L + \lambda (U - L)$ 
5:   end for
6: end for

```

Şekil 3.8 Güncellenmiş Başlangıç Fonksiyonu Sözde Kodu

3.3.2 Amaç Fonksiyonu Hesaplama Prosedürü

EMA ilk olarak sürekli fonksiyonların eniyilenmesi için oluşturulduğundan, pozisyon vektörleri ile ifade edilen koordinat değerleri, doğrudan amaç fonksiyonu hesaplamasında değişken değerleri olarak kullanılmıştır. Fakat EMA'nın, ARP gibi süreksizlik içeren kombinatorik problemlerin çözümünde kullanılması için farklı bir amaç fonksiyonu hesaplama prosedürüne ihtiyaç vardır. Başka bir deyişle, koordinat değerleri, müşteriler (düğümler) gibi kesikli değişkenleri ifade edemeyeceğinden, bir şekilde bu pozisyon vektörlerinin çözümlenerek, uygun

birer çözüm göstermeleri sağlanmalıdır. Bu sebeple, bu çalışmada güncellenmiş bir amaç fonksiyonu hesaplama prosedürü (AFH) önerilmiştir.

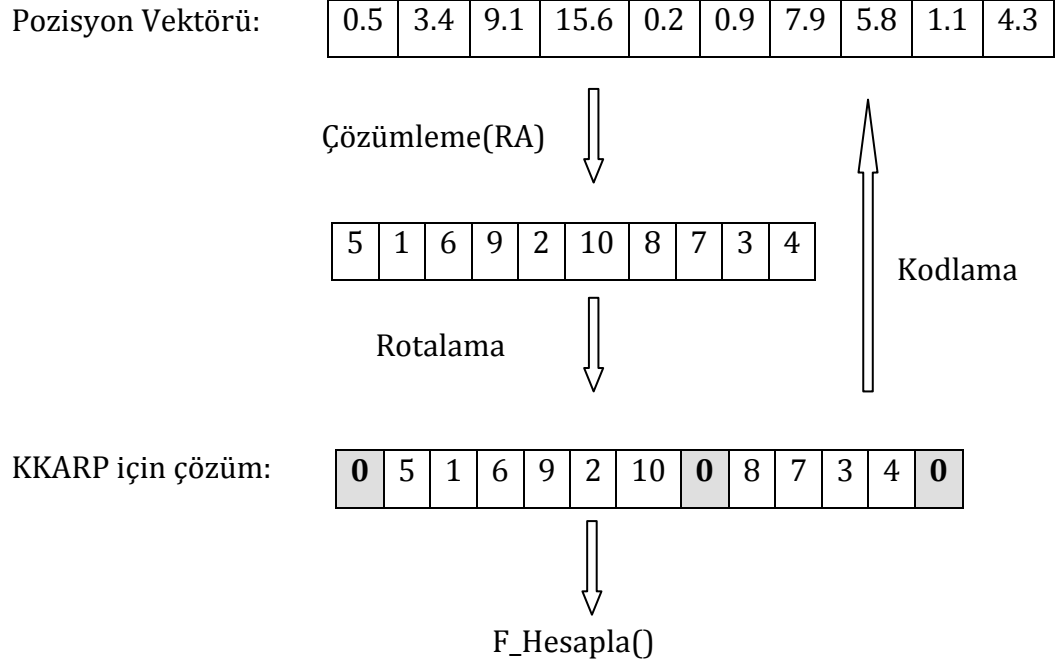
AFH, temel olarak, (1) Kod çözme, (2) Rotalama, (3) Kodlama adımlarından oluşmaktadır. Kod çözme adımında, RA yöntemi parçacıklara uygulanır. Başka bir deyişle, pozisyon vektörünün bileşenleri RA ile çözümlenir (Chang ve diğ. 2008, Wu ve diğ. 2007). RA yönteminde, herhangi n -boyutlu bir düzlem için (n -müşterili problem), pozisyon vektörlerinin bileşenleri (koordinat değerleri) azalmayan bir şekilde sıralanır. Sıralanan bu koordinat değerlerinin orijinal indisleri ise bir çözüm gösterir. Şekil 3.9'da 10 boyutlu bir problem (10 müşterili) için RA yöntemi gösterilmiştir. Orijinal pozisyon vektöründeki en küçük bileşen değeri $\min x_k = 0.2$ olduğundan, RA uygulandıktan sonra 5 numaralı bileşen ilk sırada olacak ve diğer indisler de aynı şekilde azalmayan bileşen değerlerine göre sıralanacaktır. Sonuç olarak, çözümlenmiş pozisyon vektöründeki sıralama da, uygun bir çözümü göstermektedir. Şekil 3.9'da verilen örnek için bu sıralama: 5-1-6-9-2-10-8-7-3-4 sırasındadır.

Pozisyon vektörü (Önce):										
k indisi:	1	2	3	4	5	6	7	8	9	10
k . bileşen değeri:	0.5	3.4	9.1	15.6	0.2	0.9	7.9	5.8	1.1	4.3
Pozisyon vektörü (Çözümlenmiş):										
Sıralama:	5	1	6	9	2	10	8	7	3	4
Sıralanmış Bileşen Değeri:	0.2	0.5	0.9	1.1	3.4	4.3	5.8	7.9	9.1	15.6

Şekil 3.9 Rassal Anahtar Yöntemi

Rotalama adımında, bir önceki aşamada çözümlenerek elde edilmiş olan müşteri sıralarının, rotalanarak KKARP için uygun birer çözüm göstermeleri sağlanır. Bu aşamada, depo düğümü (düğüm 0) bu sıralama içerisinde, araç kapasitesinin dolduğu noktalara eklenir. Başka bir deyişle, kapasite kısıtının sağlanamadığı (araç kapasitesinin dolduğu) durumlarda, depodan yeni bir araç çıkartılarak rotanın uygun bir çözüm olarak kalması sağlanır. Kodlama bölümünde ise, pozisyon vektörünün koordinat değerleri geri kodlanarak orijinal haline getirilmesi sağlanır. Böylece, amaç fonksiyonu değeri hesaplanmış parçacık, bir

sonraki iterasyonda kullanılabilir hale geri getirilir. Şekil 3.10, güncellenmiş AFH prosedürünü özetlemektedir.



Şekil 3.10 Güncellenmiş Amaç Fonksiyonu Hesaplama Prosedürü

3.3.3 Yerel İyileştirme

Bu çalışma kapsamında, önerilen EMA mekanizmasına İterasyonlu Yer Değiştirme (İYD) yerel arama metodu entegre edilerek EMA'nın daha etkin ve verimli bir şekilde çalışması sağlanmıştır. İYD, diğer yerel arama metotlarına (2-opt, 3-opt, NNH vb.) göre daha hızlı çalışan bir metottur (William ve diğ. 2008). İYD ilk olarak Ho ve Jing (2003) tarafında bir sıralama probleminde genetik algoritma tarafından bulunan çözümlerin iyileştirilmesinde kullanılmıştır. William ve diğ. (2008), İYD'yi çok depolu araç rotalama problemlerinde genetik algoritma ile birlikte kullanmışlardır. İYD algoritmasının temel adımları şöyle sıralanabilir:

Adım 1: Rassal olarak seçilen bir parçacığın (Ebeveyn), rassal olarak iki bileşenini seç.

Adım 2: Seçilen iki bileşeni yer değiştir.

Adım 3: Yer değiştirilmiş bileşenleri komşuları ile yer değiştirerek, 4 adet daha çocuk üret.

Adım 4: Çocukların amaç fonksiyonlarını hesapla.

Adım 5: En iyi çözümü gösteren çocuk, ebeveynden daha iyi ise, ebeveyn ile çocuğu yer değiştir. Daha iyi bir çocuk bulunamıyor ise dur.

İYD algoritmasının uygulamasına bir örnek Şekil 3.11 'da verilmiştir.

Ebeveyn	0.5	3.4	9.1	15.6	0.2	0.9	7.9	5.8	1.1	4.3
Çocuklar										
C1	0.5	3.4	0.9	15.6	0.2	9.1	7.9	5.8	1.1	4.3
C2	0.5	3.4	0.9	15.6	9.1	0.2	7.9	5.8	1.1	4.3
C3	0.5	3.4	0.9	15.6	0.2	7.9	9.1	5.8	1.1	4.3
C4	0.5	0.9	3.4	15.6	0.2	9.1	7.9	5.8	1.1	4.3
C5	0.5	3.4	15.6	0.9	0.2	9.1	7.9	5.8	1.1	4.3

Şekil 3.11 İterasyonlu Yer Değiştirme Algoritması

Her iterasyonda bileşen indislerinin, bileşen değerlerine göre sıralaması gerekeceğinden EMA'nın işlem hacmi diğer temel meta-sezgisel yöntemlere göre daha fazladır. Bu sebeple, geleneksel yerel iyileştirme yöntemleri yerine İYD tercih edilmiştir. İYD'nin diğer metotlar ile karşılaştırıldığında iki temel üstünlüğü vardır:

(1) İYD, çözümlenmiş rotalar yerine, doğrudan pozisyon vektörleri üzerinde çalışmaktadır.

(2) Diğer temel iyileştirme metotlarına göre çok daha hızlı çalışmaktadır.

Bu iki temel avantajı yüzünden İYD, EMA ile kullanıldığında, zaten yüksek olan işlem sayısının daha fazla artmasını engellemektedir.

3.3.4 Melez Elektromanyetik Algoritma

Bu çalışmada önerilen melez EMA (HEMA), Şekil 3.12'de özetlenmiştir. Bu melez mekanizmaya göre ilk önce, İYD ile geliştirilecek parçacık sayısı ($[\lambda m]$) rassal olarak belirlenir (bkz. Şekil 3.12, Satır 7-8). İYD ile geliştirilecek parçacıklar ise, GA'da olduğu gibi rulet çemberi (Goldberg 1975) yöntemine göre seçilir (bkz. Şekil 3.12, Satır 8-10). Rulet Çemberi metodunda, daha iyi çözümler gösteren parçacıklara, daha yüksek seçilme olasılığı atanarak, bu parçacıkların İYD için seçilmeleri sağlanmaya çalışılmıştır. Rulet Çemberi metodunun sözde kodu Şekil 3.13'te verilmiştir. Şekil 3.12'da izin verilen maksimum iterasyon sayısı, *max_iter* ile ifade edilmektedir.

```

1: Başlangıç (pop)
2: iter = 1
3: while iter ≤ max_iter do
4:   for i = 1 to m do
5:     f(xi) = f_Hesapla(xi)
6:   end for
7:   λ ← Uniform(0,1)
8:   for j = 1 to ⌈λm⌉ do
9:     i = RuletÇemberi(pop)
10:    IYD(xi)
11:  end for
12:  F_Hesapla(pop)
13:  Hareket(pop)
14:  S_Kontrol(pop)
15:  iter = iter + 1
16: end while

```

Şekil 3.12 Melez Elektromanyetik Algoritma Sözde Kodu

```

1: toplam = 0
2: for i=1 to m do
3:   toplam = toplam + f(xi)
4: end for
5: p = Uniform(0,1)
6: olasılık = 0
7: for i = 1 to m do
8:   olasılık = olasılık + f(xi)/toplam
9:   if p ≤ olasılık then
10:    seçim = i
11:    break
12:  end if
11: end for

```

Şekil 3.13 Rulet Çemberi Sözde Kodu

4. Araştırma Sonuçları ve Yorumlar

Bu bölüm, bir önceki bölümde açıklanan melez elektromanyetik algoritmanın (HEMA) literatürden alınan test problemleri üstünde uygulanması ve sonuçların sezgisel ve meta-sezgisel yöntemler ile hem çözüm kalitesi hem de çözüm süresi açısından karşılaştırılmasını içermektedir. HEMA ilk olarak 14 adet küçük-orta ölçekli problemden oluşan Christofides (1979) problemleri üzerinde test edilmiştir. Buna ek olarak 20 adet büyük ölçekli problemlerden oluşan Golden (1998) test problemleri üzerinde HEMA'nın kalitesi incelenmiştir.

Bu çalışma kapsamında önerilen HEMA, C++ programlama dilinde kodlanmış ve 2.00 GB RAM ve Intel Core2 Duo, 2.00 Ghz işlemciye sahip bir bilgisayarda çalıştırılmıştır. Her meta-sezgisel algortmada olduğu gibi, algoritmanın uygulama parametrelerinin çözüm hızı ve kalitesine ciddi etkileri vardır. Bu sebeple, yapılan öncül deneylerden de faydalanarak, HEMA'nın temel parametreleri önceden belirlenmiştir. Algoritmada kullanılan parametreler her iki problem seti için de Tablo 4.1'de özetlenmiştir. Tablo 4.1'te, Christofides test problemlerinin küçük ve orta büyüklükteki problemlerden oluşmasından dolayı, izin verilen maksimum iterasyon sayısı (max_iter), problem büyüklüğünün 10 katı olarak alınmıştır ($10n$). Çok daha büyük ölçekli problemlerden oluşan Golden test problemleri için ise, maksimum iterasyon sayısı problem boyutunun 5 katı olarak alınmıştır ($5n$). Maksimum iterasyon sayısı dışında, koordinat değerlerinin alt (L) ve üst (U) sınırları ise $[-100,100]$ olarak, popülasyon büyüklüğü (m) ise Christofides problemleri için 25, Golden problemleri için ise 50 olarak alınmıştır. Tüm problem setleri için amaç fonksiyonu, toplam yolu en küçükleme olarak alınmıştır.

Tablo 4.1 HEMA Parametreleri

	max_iter	$[L, U]$	m
Christofides	$10n$	$[-100, 100]$	25
Golden	$5n$	$[-100, 100]$	50

4.1 Christofides Test Problemleri

Uygulamanın ilk kısmında, geliştirilen HEMA literatürde sık kullanılan 14 temel ARP üzerinde test edilmiştir. Bu test problemleri, Christofides (1979) problemleri olarak bilinmektedir ve URL: <http://mscmga.ms.ic.ac.uk/jeb/orlib/vrpinfo.html> (Erişim:2009) adresinden indirilebilir. Tablo 4.2’de bu problemlerin özellikleri verilmiştir. Bu tabloda, n müşteri sayısını, Q araçların kapasitelerini, α müşterilerde harcanacak servis zamanını ve D ise oluşturulabilecek maksimum tur uzunluğunu göstermektedir. İlk 10 problem (C1-C10) rassal olarak oluşturulmuş olup, son 4 problem ise kümelenmiş müşterileri içermektedir. Bunlara ek olarak, C6, C10, C13 ve C14 problemlerinde, kapasite kısıtına ek olarak maksimum tur uzunluğu ve servis zamanı kısıtları vardır.

Tablo 4.2 Christofides Test Problemleri

Problem	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14
n	50	75	100	150	199	50	75	100	150	199	120	100	120	100
Q	160	140	200	200	200	160	140	200	200	200	200	200	200	200
D	∞	∞	∞	∞	∞	200	160	230	200	200	∞	∞	720	1040
α	0	0	0	0	0	10	10	10	10	10	0	0	50	90

Christofides test problemleri kullanılarak, İYD ile melezleştirilmiş EMA’nın (HEMA), üstünlüğünün görülebilmesi için melezleştirilmemiş EMA (EMA) ile karşılaştırılması yapılmıştır. Bu karşılaştırma sonuçları Tablo 4.3’te verilmiştir. Her problem 25 kez çözülerek, en kötü, en iyi ve ortalama çözüm değerleri belirlenmiştir. Koyu yazılmış sonuçlar, iki algoritma karşılaştırıldığında daha iyi olan çözüm değerlerini göstermektedir. Tablo 4.3 incelendiğinde, C1 problemi dışında geri kalan 13 problem için, HEMA’nın EMA’ya karşı çözüm kalitesi açısından üstünlüğü görülmektedir. Bu sonuçlar da, İYD’nin çözüm kalitesini güçlü bir şekilde etkilediğini kanıtlamaktadır. Tablo 4.3’teki son sütün ise, HEMA’nın

bulduğu sonuçların, EMA'dan farkı (% *Fark*) verilmiştir. % *Fark*, Denklem 4.1'de görüldüğü şekilde hesaplanır. $f(x^{EMA})$ ve $f(x^{HEMA})$ sırasıyla, EMA'nın ve HEMA'nın bulunduğu çözüm değerlerini göstermektedir.

$$\% \text{ Fark} = \frac{f(x^{HEMA}) - f(x^{EMA})}{f(x^{EMA})} \quad (4.1)$$

Tablo 4.3 incelendiğinde, HEMA'nın, EMA'ya göre yaklaşık olarak %3.5 daha iyi çözüm değerleri bulunduğu görülmektedir. Ek olarak, EMA'nın içerisinde yerel iyileştirme metodu bulunmadığından, daha kısa işlem süreleri ile çalıştığı söylenebilir. Fakat süre sütunları karşılaştırıldığında farkların çok ciddi boyutlarda olmadığı görülmektedir. Bu sonuç da, İYD'nin hızlı ve etkin bir yerel iyileştirme yöntemi olduğunu göstermektedir.

Önerilen HEMA yönteminin daha detaylı bir şekilde incelenebilmesi için diğer temel meta-sezgisel yöntemler ile Christofides test problemleri üzerinden karşılaştırılması yapılmıştır. Christofides test problemleri üzerinden yapılan karşılaştırmada kullanılan meta-sezgiseller, Osman (1993) tarafından geliştirilen TA algoritması (OSM_TA), TB algoritması (OSM_TB) (Osman 2005), Baker ve Ayechev (2003) tarafından geliştirilen GA (GA_BA), Prins (2004) tarafından önerilen başka bir GA (GA_P), Yu ve diğ. tarafından (2008) önerilen KKA (KKA_Y) ve son olarak Ai ve Kachitvichyanukul (2007) tarafından geliştirilen PSO (PSO_AK) olarak sıralanabilir. Bu karşılaştırmanın sonuçları Tablo 4.4'te verilmiştir. Son sütunda verilen fark değerleri, literatürde bugüne kadar yayınlanan en iyi sonuçlar ile HEMA'nın sonuçları arasındaki yüzdellik farkı göstermektedir. Farkın 0.00 olması, HEMA'nın bilinen en iyi sonucu bulunduğunu göstermektedir. Literatürde, bugüne kadar bulunmuş en iyi sonuçlar (En İyi) Tablo 4.4'ün ilk sütununda verilmiştir. Koyu renkli sonuç değerleri HEMA'nın en iyi değerleri bulunduğu problemleri göstermektedir. Tüm meta-sezgisel yöntemlerin ortalama olarak en iyiye göre farkları (*Ort*) en alt satırda görülmektedir.

Tablo 4.4'te de görüldüğü gibi, HEMA 14 problem içerisinde 4'ü için şu ana kadar bilinen en iyi sonuçları bulmuştur. Bu sonuç geleneksel ve daha yeni olan meta-sezgisel yöntemler ile karşılaştırıldığında, HEMA'nın güçlü bir meta-sezgisel olduğu görülmektedir. C3, C4 ve C12 problemleri için sapmanın ortalama olarak %2.5 civarında olduğu görülmektedir. Diğer taraftan, kalan problemler için en iyi sonuçlar ile HEMA'nın sonuçları arasındaki farkın %0.6 civarına kadar düştüğü ve toplamda ortalama olarak %1.05 olduğu görülmektedir.

Burada dikkat edilmesi gereken önemli bir nokta ise, bu meta-sezgiseller dışında literatürde karşılaştırma yapılabilecek birçok algoritmanın mevcut olmasıdır. Fakat bu çalışma kapsamında, bu test problemleri için temel meta-sezgiseller olarak kabul edilen TA, TB ve GA dikkate alınmış, ayrıca mekanizma bakımından EMA'ya yakın olan PSO ve KKA seçilmiştir. Bu çalışmanın temel amacı, HEMA mekanizmasının KKARP çözümünde en az diğer meta-sezgiseller kadar etkili olduğunun ispatlanmasıdır.

Geliştirilen bir meta-sezgisel algoritmanın tam olarak değerlendirilebilmesi için, çözüm kalitesi dışında, işlem sürelerinin de incelenmesi gerekmektedir (Yu ve diğ. 2008). Günümüzde hem algoritmaların karmaşıklıklarının yüksek olması, hem de bu tip algoritmaların uygulandıkları bilgisayarların birbirlerinden farklı olmalarından dolayı, farklı meta-sezgiseller arasında objektif bir karşılaştırma yapmak çok zordur. Bunu rağmen, farklı işlemcilerin saniyede yaptıkları işlem sayıları (Mflops) literatürde bulunmaktadır.

Tablo 4.4 Christofides Test Problemleri Sonuçları

Problem	Eniyi	TA-OSM	TB-OSM	GA-B&A	GA-P	PSO-A&K	KKA-Y	HEMA	Sapma (%)
C1	524.61	524	528	524.81	524.61	524.61	524.61	524.61	0.00
C2	835.26	844	838	849.77	835.26	844.42	835.26	849.77	1.74
C3	826.14	835	829	840.72	826.14	829.40	830.02	844.72	2.25
C4	1028.42	1052	1058	1055.85	1030.46	1048.89	1028.42	1059.03	2.98
C5	1291.45	1354	1376	1378.73	1296.39	1323.89	1305.40	1302.33	0.84
C6	555.43	555	555	560.29	555.43	555.43	555.43	555.43	0.00
C7	909.68	913	909	914.13	909.68	917.68	909.68	909.68	0.00
C8	865.94	866	866	872.82	865.94	867.01	865.94	866.32	0.04
C9	1162.55	1188	1164	1193.05	1162.55	1181.14	1162.55	1181.6	1.64
C10	1395.85	1422	1418	1483.06	1402.75	1428.46	1395.85	1417.88	1.58
C11	1042.11	1042	1176	1060.24	1042.11	1051.87	1042.11	1042.11	0.00
C12	819.56	819	826	877.8	819.56	819.56	819.56	847.56	3.42
C13	1541.14	1547	1545	1562.25	1542.86	1546.20	1545.92	1542.86	0.11
C14	866.37	866	890	872.34	866.37	866.37	866.37	866.37	0.00
Ort.(%)	+0.00	+1.00	+2.07	+2.49	+0.85	+0.87	+0.13	+1.04	

Dongarra (2001) lineer denklem çözücüler kullanarak (LINPACK), birçok farklı işlemci için Mflops değerleri çıkartmıştır. Bu Mflops değerleri kullanılarak çok kesin olmasa da algoritma hızları arasında karşılaştırma yapılabilmektedir. Bu çalışma kapsamında, mekanizma olarak HEMA ile büyük benzerlik taşıyan KKA (Yu ve diğ. 2008) ve PSO (Ai ve Kachitvichyanukul 2007) çalışma süresi bakımından HEMA ile karşılaştırılmıştır. Bu çalışmalarda kullanılan işlemcilerin Mflops değerleri URL: http://www.netlib.org/benchmark/linpackjava/timings_list.html (Erişim: 2009) adresinden alınmıştır. Yu ve diğ., Intel Pentium 1.00 Ghz (75 MFlops) ve Ai ve Kachitvichyanukul ise Intel P4, 3.20 Ghz. (100 MFlops) işlemciye sahip bilgisayarlarda algoritmalarını uygulamışlardır. Bu çalışmada ise, Intel C2Duo, 2.00 Ghz. işlemciye sahip bilgisayar kullanılmıştır.

Tablo 4.5'te çalışmalarda kaydedilmiş süreler (K_Süre) ile Mflops değerlerinden faydalanılarak C2Duo, 2 Ghz'e göre normalize edilmiş işlem süreleri, (N_süre) saniye cinsinden verilmiştir. Tablo 4.5 incelendiğinde, KKA ve PSO'nun HEMA'ya göre daha hızlı çalıştıkları görülmektedir. HEMA'nın bu algoritmalara kıyasla daha uzun işlem sürelerine sahip olmasının en temel sebebi ise, m adet bireye sahip bir popülasyonda maksimum iterasyon sayısına kadar her bir birey için her iterasyonda, RA yöntemi içerisinde n -boyutlu bir vektöre sıralama işlemi uygulanmasıdır. Bu sıralama işlemi, HEMA'nın karmaşıklığını arttırmaktadır. Fakat daha hızlı çalışacak şekilde algoritmanın eniyilenmesi ve hızlı bir sıralama algoritması kullanılması ile bu problemin önüne geçilebileceği öngörülmüştür.

Tablo 4.5 Christofides Test Problemleri için İşlem Süreleri

Problem	PSO-A&K		KKA-Y		HEMA
	K_Süre(sn.)	N_Süre	K_Süre(sn.)	N_Süre	Süre(sn.)
C1	24	12	2	0.4	13
C2	57	28	11	2	19
C3	101	49	30	5	41
C4	223	108	211	38	132
C5	413	200	677	123	201
C6	30	15	24	4	22
C7	69	33	20	4	39
C8	115	56	57	10	103
C9	295	143	307	56	238
C10	517	251	840	153	419
C11	93	45	61	11	319
C12	88	43	31	6	190
C13	160	78	127	23	63
C14	99	48	43	8	49
Ort.		79		32	132

4.2 Golden Test Problemleri

Uygulamanın 2. bölümünde ise, geliştirilen algoritma, Christofides test problemlerine göre daha büyük ölçekli problemlerden oluşan Golden test problemleri (1998) üzerinde uygulanmıştır. Golden test problemleri 225 müşteriden 480 müşteriye kadar olan farklı test problemlerinden oluşmaktadır. Tablo 4.6'da Golden test problemlerinin özellikleri verilmiştir. Literatürde şu ana kadar bulunan en iyi sonuçlar URL: <http://www.diku.dk/~sropke> (Erişim: 2009) adresinden alınmıştır. Tablo 4.6'da, her problem için müşteri sayısı (n), araç kapasitesi (Q) ve maksimum tur zamanı (D) verilmiştir. Christofides test problemleri üzerinde karşılaştırılarak EMA'ya göre üstünlüğü kanıtlanmış HEMA'nın, Golden test problemleri için tekrar melezleştirilmemiş EMA (EMA) ile karşılaştırılmasına gerek duyulmamıştır.

Tablo 4.6 Golden Test Problemleri

Problem	n	Q	D	En İyi
P1	240	550	650	5627.54
P2	320	700	900	8447.92
P3	400	900	1200	11036.22
P4	480	1000	1600	13624.52
P5	200	900	1800	6460.98
P6	280	900	1500	8412.80
P7	360	900	1300	10181.75
P8	440	900	1200	11663.55
P9	255	1000	-	580.60
P10	323	1000	-	738.92
P11	399	1000	-	917.17
P12	483	1000	-	1107.19
P13	252	1000	-	857.19
P14	320	1000	-	1080.55
P15	396	1000	-	1340.24
P16	480	1000	-	1622.69
P17	240	200	-	707.76
P18	300	200	-	995.39
P19	360	200	-	1366.14
P20	420	200	-	1820.09

Tablo 4.7’de, Golden test problemleri üzerinde uygulanan HEMA ile elde edilmiş sonuçlar özetlenmiştir. Bunlara ek olarak HEMA, literatürden alınan sezgisel ve meta-sezgisel yöntemler ile karşılaştırılmıştır. Bu meta-sezgiseller, tabu arama (TA_C) (Cordeau ve diğ. 2001), granüler tabu arama (GTA_TV) (Toth ve Vigo 2003), genetik algoritma (GA_P) (Prins 2004), karınca kolonisi algoritması (KKA_R) (Reimann ve diğ. 2004), komşuluk arama (KA_E) (Ergun ve diğ. 2006), ark çaprazlama (AÇ_N) (Nagata 2007) ve yerel arama (YA_PR) (Psinger ve Ropke 2007) olarak sıralanabilir. Tablo 4.7’de HEMA sonuçları 20 tekrarın ortalaması olarak verilmiştir. Tablo 4.7’deki değerler bugüne kadar literatürde bulunan en iyi

sonuçlardan yüzde sapma olarak verilmiştir. Başka bir deyişle, 0.00 değeri bilinen en iyi çözümün bulunduğunu gösterir.

Tablo 4.7’de, *sayı*, bahsi geçen algoritma ile bilinen en iyi çözümlere eşit olarak bulunan problem sayısını göstermektedir. Bilinen en iyi çözümlerden ortalama yüzde olarak fark (*Ort.*) ve yapılan tekrar sonucunda bulunan kümülatif ortalama işlem süresi (*Süre*) dakika cinsinden Tablo 4.7’deki son iki satırda verilmiştir. Diğer çalışmalarda her problem için bulunan işlem süreleri kaydedilmediğinden, kümülatif ortalama işlem süreleri baz alınmıştır.

Tablo 4.7 incelendiğinde, önerilen HEMA’nın, 20 test probleminden biri için şu ana kadar bulunan en iyi sonucu bulduğu görülmektedir. Bu *sayı*, düşük gibi görünse de, diğer meta sezgiseller ile karşılaştırıldığında iyi bir sonuç olduğu görülmektedir. TA_C, GA_P ve KA_E yöntemleri ile hiçbir problem için en iyi sonuç bulunamamıştır. Ortalama %farklar (Ort %) incelendiğinde, HEMA ile, Ergun ve diğ. (2006) tarafından önerilen komşuluk arama stratejisine (KA_E) göre daha iyi sonuçlar bulunduğu görülmektedir. HEMA ile GA_P ve AÇ_N yöntemleriyle bulunan sonuçlara çok yakın sonuçlar elde edilmiştir. Toth ve Vigo tarafından (2003) önerilen TA mekanizması (GTA_TV), Reimann ve diğ. tarafından geliştirilen (2004) KKA (KKA_R) ve çok amaçlı yerel arama metodu (YA_PR) (Psinger ve Ropke 2007), HEMA ile karşılaştırıldığında bu çalışmada geliştirilen yöntemden daha üstün yöntemler olduğu görülmüştür. Özet olarak, HEMA’nın ortalama %3’lük fark değeri ile diğer sezgisel ve meta-sezgisel yöntemler kadar etkin bir yöntem olduğu söylenebilir.

Golden test problemleri üzerinde yapılan bu incelemenin daha objektif bir hale getirilmesi için, Bölüm 4.1’de de uygulandığı gibi, bu testlerin yapıldığı bilgisayar işlemcilerinin, Mflops değerleri belirlenerek, uygulamanın gerçekleştirildiği işlemciye göre normalleştirilmiş işlem süreleri cinsinden karşılaştırılmıştır (http://www.netlib.org/benchmark/linpackjava/timings_list.html, Erişim: 2009). Cordeau ve diğ. (2001) Pentium IV, 2. Ghz., Toth ve Vigo (2003) Pentium II 400 Mhz., Prins (2004) Pentium 200 Mhz., Reimann ve diğ. (2004) Pentium III 1.00 Ghz., Ergun ve diğ. (2006) Pentium III 733 Mhz., Nagata

(2007) Xeon 3.2Ghz., Pisinger ve Ropke (2007) ise Pentium IV 3.00 Ghz. işlemcilerle sahip bilgisayarlar kullanmışlardır. Bu işlemciler için Mflops değerleri ve C2Duo, 2 Ghz'e göre normalize edilmiş işlem süreleri (N_süre) ve kaydedilmiş süreleri (K_Süre) Tablo 4.8'de verilmiştir. Tüm araştırmacılar ortak olarak toplam ortalama kümülatif süreleri kaydettikleri için, karşılaştırmalar bu değerler üzerinden yapılmıştır. Golden test problemleri, müşteri sayısı ve topografik özellikleri bakımından çok benzedikleri için bu tip bir karşılaştırma yapılmasında bir sakınca görülmemiştir.

Tablo 4.8 incelendiğinde, beklenildiği gibi HEMA'nın işlem süresi açısından diğer meta-sezgiseller ile karşılaştırıldığında daha yavaş olduğu görülmektedir. HEMA, AÇ_N'ye çok yakın işlem süresi içinde problemleri çözümlerse bile diğer yöntemlerin gerisinde kalmıştır. Daha önce de belirtildiği gibi, RA metodunun içerisindeki sıralama işlemi algoritmanın yavaşlamasına sebep olmaktadır. Çözüm kalitesi açısından diğer meta-sezgiseller ile rekabet edebilen HEMA, işlem süreleri açısından geliştirilmeye açıktır. HEMA'da, her parçacığın pozisyon vektör uzunluğu doğrudan problemdeki müşteri sayısı ile bağlantılı olduğundan, Golden test problemleri gibi büyük ölçekli problemlerde işlem süreleri açısından geride kaldığı söylenebilir.

Tablo 4.7 Golden Test Problemleri Sonuçları

Problem	TA_C	GTA_TV	GA_P	KKA_R	KA_E	AÇ_N	YA_PR	HEMA
P1	0.97	0.88	1.93	0.34	2.03	-	0.42	2.01
P2	2.48	0.77	1.24	0.00	5.56	-	0.25	4.37
P3	0.01	1.48	3.32	0.00	9.70	-	0.10	1.92
P4	0.85	0.10	9.44	0.00	12.42	-	0.08	0.00
P5	4.57	0.00	3.66	0.00	1.69	-	0.09	6.87
P6	1.48	0.20	6.54	0.00	5.03	-	0.04	9.76
P7	0.84	0.34	3.59	0.14	9.18	-	0.00	1.13
P8	1.77	2.34	3.20	1.42	8.32	-	0.43	4.37
P9	1.17	-	2.20	1.88	1.26	0.00	0.78	5.01
P10	1.87	-	1.72	1.69	1.48	0.00	1.35	1.31
P11	1.30	-	2.06	1.73	1.70	0.00	0.60	0.98
P12	1.11	-	3.61	2.40	2.48	0.12	1.07	7.73
P13	2.18	-	1.35	2.10	1.60	0.00	0.87	2.74
P14	1.99	-	1.45	0.53	1.53	0.00	1.37	0.67
P15	1.75	-	2.18	2.02	2.01	0.00	1.47	1.89
P16	1.50	-	1.83	1.74	1.25	33.81	1.01	0.76
P17	0.45	-	0.47	0.38	1.23	0.00	0.16	1.54
P18	1.93	-	2.15	1.95	2.81	0.00	0.71	3.04
P19	1.29	-	2.55	0.76	2.83	0.00	0.59	2.99
P20	1.88	-	5.26	1.45	3.47	0.02	0.59	1.40
Sayı	0	1	0	5	0	9	1	1
Ort %	1.57	0.76	2.99	1.03	3.88	2.83	0.60	3.01
Süre(dk.)	56.11	17.55	66.90	49.33	137.05	413.68	107.61	216.89

Tablo 4.8 Golden Test Problemleri için İşlem Süresi

Yöntem	TA_C	GTA_TV	GA_P	KKA_R	KA_E	AÇ_N	YA_PR	HEMA
Mflops	532	60	22	315	88	600	840	915
K_Süre(dk.)	65	17	67	49	137	414	107	217
N_Süre(dk.)	26	<1	1	11	9	189	68	217

SONUÇ

Bu çalışma kapsamında, lojistik sistemlerin temelini oluşturan kapasite kısıtlı araç rotalama problemlerinin çözümü için literatüre yakın zamanda giren elektromanyetik algoritma kullanılarak melez bir yaklaşım geliştirilmiştir. Elektromanyetik algoritma elektrik yüklü parçacıkların birbirlerine uyguladıkları itme ve çekme kuvvetlerini esas alan popülasyon tabanlı yeni bir meta-sezgisel yöntemdir. Geliştirilen algoritmayı kesikli değişkenlere sahip eniyileme problemlerinde kullanabilmek için, bu çalışmada rassal anahtar metodunu kullanan güncellenmiş bir amaç fonksiyonu hesaplama prosedürü önerilmiştir. Önerilen algoritmanın, yerel iyileştirme metodu olan iterasyonlu yer değiştirme yöntemi ile melezlenerek daha etkin çalışması sağlanmıştır.

Çalışma kapsamında geliştirilen melez algoritma, literatürden alınan çok sayıda test probleminin üzerinde uygulanmıştır. Bu uygulamalarda, melez elektromanyetik algoritma çözüm kalitesi ve çözüm süresi bakımından diğer temel meta-sezgisel yöntemler ile karşılaştırılmıştır. Sonuçlar incelendiğinde, HEMA'nın çözüm kalitesi bakımından en az diğer meta-sezgisel yöntemler kadar başarılı olduğu görülmüştür. Çözüm süreleri karşılaştırıldığında ise, HEMA'nın az da olsa literatürdeki temel meta-sezgisellerin gerisinde kaldığı anlaşılmıştır. Bunun en önemli sebebi rassal anahtar metodu içerisinde gerçekleştirilen sıralama işlemidir. Her parçacık için, her iterasyonda yapılan bu sıralama işlemi algoritmanın karmaşıklığını arttırmaktadır. Bununla birlikte, çözüm sürelerinin yinede kabul edilebilir sınırlar içerisinde kaldığı da söylenebilir.

Diğer algoritmalar ile kıyaslandığında, EMA'nın en temel sorunu, algoritmanın karmaşıklığı olduğu söylenebilir. Rassal anahtar yöntemi içerisinde kullanılan sıralama algoritması yerine daha hızlı bir algoritmanın kullanılması az

da olsa bu problemin önüne geçebilir. Daha etkin bir güncelleme ise rassal anahtar metodunun yerine başka bir yöntem geliştirilmesiyle sağlanabilir. Önerilen algoritma üzerinde ileride bu tür çalışmalar planlanmaktadır.

Her meta-sezgiselde olduğu gibi, algoritma parametrelerinin algoritmanın hızı üzerinde ciddi etkileri vardır. Elektromanyetik algoritma için literatürde bu konuda da ciddi eksiklikler bulunmaktadır. Çalışmanın bundan sonraki aşamalarında, çok faktörlü deney setleri hazırlanarak farklı problemler üzerinde optimal parametre seviyelerine karar verilmesi planlanmaktadır. Böylece popülasyon büyüklüğü, iterasyon sayısı gibi parametrelerin eniyilenmesi sağlanabilir. Bu tip bir deney tasarımının algoritmanın hızı üzerinde de ciddi etkileri olacağı öngörülmektedir.

KAYNAKLAR

A. VAN BREEDAM. 1994. "An Analysis of the Behavior of Heuristics for the Vehicle Routing Problem for a Selection of Problems with Vehicle-Related, Customer-Related, and Time-Related Constraints". Ph.D. dissertation, University of Antwerp.

AGARWAL, Y., MATHUR, K., SALKIN, H.M. 1989. A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks* 19, 731–749.

AI, T.J. & KACHITVICHYANUKUL, V. 2007. A particle swarm optimization for the capacitated vehicle routing problem. *International Journal of Logistics and SCM Systems*, 2, 50 - 55.

AI, T.J. & KACHITVICHYANUKUL, V. 2009. Particle swarm optimization and two solution representations for solving the capacitated vehicle routing problem *Computers & Industrial Engineering* 56, 380–387.

ALTINKEMER K. ve GAVISH B. 1991. Parallel Savings Based Heuristics for the Delivery Problem. *OPERATIONS RESEARCH* Vol. 39, No. 3.

BAKER, B.M., AYECHIEW, M.A. 2003. A genetic algorithm for the vehicle routing problem. *Computer and Operational Research*. 7, 301-317.

BALL, M.O., MAGNANTI, T.L., MONMA, C.L., NEMHAUSER, G.L.: (eds.) 1995. *Network Routing*. Handbooks on Operations Research and Management Science, 8. Amsterdam: Elsevier.

BEAN, J. C. 1994. Genetic algorithms and random keys for sequencing and optimization. *ORSA Journal on Computing*, 6(2), 154–160.

BIANCHESSI, N., RIGHINI, G. 2007. “Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery,” *Computers & Operations Research*, Vol: 34, pp. 578–594.

BİRBİL, S. I., & FANG, S. C. 2003. An Electromagnetism-like Mechanism for Global Optimization. *Journal of Global Optimization*, 25, 263–282.

BİRBİL, S. I., & FEYZİOĞLU, O. 2003. A global optimization method for solving fuzzy relation equations. *Lecture Notes in Artificial Intelligence*, 2715, 718–724.

BJARNADOTTİR, A.S. 2004. Solving Vehicle Routing Problems with Genetic Algorithms. Msc. Thesis(published). Technical University of Denmark.

BODIN, L.D., GOLDEN, B.L., ASSAD, A.A., BALL, M.O. 1983. “Routing and scheduling of vehicles and crews, the state of the art,” *Computers and Operations Research*, Special Issue on the Routing and Scheduling of Vehicles and Crews, Vol: 10, No 2, pp. 63-211.

BRANDÃO, J. 2004. “A Tabu search algorithm for the open vehicle routing problem,” *European Journal of Operational Research* 157, Vol: 3, pp. 552–564.

BRANDÃO, J. 2006. “A new Tabu search algorithm for the vehicle routing problem with backhauls,” *European Journal of Operational Research*, Vol: 173, pp. 540–555.

BULLENEIMER, B., HARTL, R.F., STRAUSS, C. 1999. An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research*, 89, 319-328.

BULLNHEIMER B., HARTL R.F., STRAUSS C. 1997. A New Rank Based Version of the Ant System: A Computational Study, *Central European Journal for Operations Research and Economics*.

CHANG P.-C., SHIH-HSIN C. & CHIN-YUAN F. 2009. A hybrid electromagnetism-like algorithm for single machine scheduling problem. *Expert Systems with Applications*, 36(2-1), 1259-1267.

CHEN, A. L., YANG, G.K., Wu, Z.M. 2006. Hybrid discrete particle swarm optimization algorithm for capacitated vehicle routing problem. *Journal of Zhejiang University Science A*, 7, 607-614.

CHIANG, W.C., RUSSELL, R.A., 1996. Simulated annealing metaheuristics for the vehicle routing problem with time windows. *Annals of Operations Research* 63, 3-27.

CHIANG, W.C., RUSSELL, R.A., 1997. A reactive tabu search metaheuristic for the vehicle routing problem with time windows. *INFORMS Journal on Computing* 9, 417-430.

CHRISTOFIDES, N., MINGOZZI, A., & TOTH, P. 1979. The vehicle routing problem. In N. Christofides, A. Mingozzi, P. Toth & C. Sandi (Eds.), *Combinatorial Optimization* (pp. 325-338). Chichester: Wiley.

CLARKE, G., & WRIGHT, J. W. 1964. Scheduling of vehicles from a depot to a number of delivery points. *Operations Research*, 12, 568-581.

CLERC, M. 2006. *Particle Swarm Optimization*. London: ISTE.

CORDEAU, J. F., LAPORTE, G., & MERCIER, A. 2001. A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society*, 52, 928-936.

CORDEAU, J., GENDRAU, M., LAPORTE, G. 1997. A "Tabu search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems," Vol: 30, Issue 2, pp. 105-119.

CORDEAU, J.F., LAPORTE, G. 2001. "A Tabu search algorithm for the site dependent vehicle routing problem with time windows," *INFOR*, Vol: 39, pp.292-298.

CRISPIM, J., BRANDÃO, J. 2005. "Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls," *Journal of the Operational Research Society*, Vol: 56, pp. 1296-1302.

DANTZIG, G., FULKERSON, R., & JOHNSON, S. 1954. Solution of a large-scale travelling salesman problem. *Operations Research*, 2, 393-410.

DEBELS, D., REYCK, B. D., LEUS, R., & VANHOUCKE, M. 2006. A hybrid scatter search/electromagnetism meta-heuristic for project scheduling. *European Journal of Operational Research*, 169, 638-653.

DESROCHERS, M. ve VERHOOG, T.W. 1989. A Matching Based Savings Algorithm for the Vehicle Routing Problem, *Les Cahiers du GERAD G-89-04*, Ecole des Hautes Etudes Commerciales de Montreal.

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

DOERNER, K., GRONALT, M., HARTL, R. F., REIMANN, M., STRAUSS, C., & STUMMER, M. 2002. Saving ants for the vehicle routing problem. In Cagnoni et al. (Eds.), *EvoWorkshops 2002*. LNCS (vol. 2279, pp. 11-20). Berlin: Springer-Verlag.

DONGARRA, J. 2001. Performance of various computer using standard linear equations software. Report CS-89-85, University of Tennessee.

EILON, S., WATSON-GANDY, C. D. T., & CHRISTOFIDES, N. 1971. Distribution management: Mathematical modelling and practical analysis. NY: Hafner Publication Co.

EKSİOĞLU, B., VURAL A.V., REISMAN A. 2009. The vehicle routing problem: A taxonomic review. Computers & Industrial Engineering (In Press).

ERGUN, Ö., ORLIN, J. B., & STEELE-FELDMAN, A. 2006. Creating very large scale neighborhoods out of smaller ones by compounding moves. Journal of Heuristics, 12, 115–140.

EROL, V. 2006. “Araç Rotalama Problemleri için Popülasyon ve Komsuluk Tabanlı Metasezgisel Bir Algoritmanın Tasarımı ve Uygulaması”, Yüksek Lisans Tezi (Yayımlanmış), Sistem Mühendisliği, Yıldız Teknik Üniversitesi.

FALKENAUER, E. 1996. A hybrid grouping genetic algorithm for bin packing. Journal of Heuristics, 2:5(30).

FISHER, M. L. 1994. Optimal Solution of Vehicle Routing Problems Using Minimum K-trees, Operations Research 42, 626-642.

FISHER, M. L. ve JAIKUMAR, R. 1981. A Generalized Assignment Heuristic for Vehicle Routing. Networks, 11:109-124, 1981.

GAMBARDELLA, L.M., TAILLARD, E., AGAZZI, G., 1999. MACSVRPTW: A multiple ant colony system for vehicle routing problems with time windows. Technical Report IDSIA-06- 99, Lugano, Switzerland.

GENDREAU, M., GUERTIN, F., POTVIN, J., SE’GUIN, R. 2006. “Neighborhood search heuristics for a dynamic vehicle dispatching problem with pick-ups and deliveries,” Transportation Research Part C, Vol: 14, pp. 157–174.

GENDREAU, M., LAPORTE, G., & SEGUIN, R. 1996. Stochastic vehicle routing. *European Journal of Operational Research*, 88, 3–12.

GENDREAU, M., LAPORTE, G., MUSARAGANYI, C., TAILLARD, E.D. 1999. A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research* 41, 421-451.

GILLET ve L. R. MILLER. 1974. "A Heuristic Algorithm for the Vehicle Dispatch Problem". *Operations Research*, 22:340-349.

GLOVER. 1986. Future paths for integer programming and links to artificial intelligence, *Comp. Oper. Res.* 13, 533–549.

GLOVER. 1989. Tabu search-Part I, *ORSA J. Comput.* 1, 190–206.

GLOVER. 1990. Tabu search-Part II, *ORSA J. Comput.* 2, 4–32.

GOLDBERG D.E. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, USA.

GOLDEN, B. L., MAGNANTI, T. L., & NGUYAN, H. Q. 1972. Implementing vehicle routing algorithms. *Networks*, 7(2), 113–148.

GOLDEN, B. L., WASIL, E. A., KELLY, J. P., & CHAO, I. 1998. The impact of metaheuristics on solving the vehicle routing problem: Algorithms, problem sets and computational results. In T. G. Grainic & G. Laporte (Eds.), *Fleet management and logistics* (pp. 33–56). Boston: Kluwer.

HADJICONSTANTINO, E., CHRISTOFIDES, N., MINGOZZI, A. 1996. A new exact algorithm for the vehicle routing problem based on q -paths and k -shortest paths relaxations. *Ann. Oper. Res.* 61, 21–43.

HOLLAND-I. H. 1975. *Adaptation in Natural and Artificial Systems*. University of Michigan Press. Ann Arbor.

HOMBERGER, J., GEHRING, H., 1999. Two evolutionary metaheuristics for the vehicle routing problem with time windows. In: Laporte, G., Semet, F. (Eds.), *Metaheuristics for Location and Routing Problems*. *Information Systems and Operational Research* 37 (Special issue), 297–318.

KENNEDY, J., & EBERHART, R. 1995. Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, 1942–1948.

KENNEDY, J., & EBERHART, R. C. 2001. *Swarm Intelligence*. San Francisco: Morgan Kaufman Publishers.

KINDERWATER ve M. W. P. SAVELSBERGH. 1997. "Vehicle Routing: Handling Edge Exchanges". In E. H. L. Aarts and J. K. Lenstra (eds), *Local Search in Combinatorial Optimization* Wiley, Chichester.

KIRKPATRICK, S.; C. D. GELATT, M. P. VECCHI. 1983. Optimization by Simulated Annealing. *Science. New Series* 220 (4598): 671-680

LAPORTE ve F. SEMET. 1999. Classical heuristics for the vehicle routing problem. Technical Report G-98-54, GERAD.

LAPORTE, G. 1992. "The vehicle routing problem: an overview of exact and approximate algorithms". *Eur. J. Opl. Res.* 59, 345–358, 28.

LAPORTE, G. 1997. Vehicle routing. In: Dell'Amico, Maffioli, Martello (eds.) *Annotated Bibliographies in Combinatorial Optimization*. New York, Wiley.

LAWRENCE, B. ve BRUCE, G. 1981. Classification of vehicle routing and scheduling. *Networks*, 11, 97-108.

LEE, Z.-J, SU -F., ve LEE, C.-F. 2003. Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics. IEEE Transactionson Systems, Man and Cyberkinetics, Part B, 33, 133-121.

LENSTRA, J., RINNOOY KAN, A., 1981. Complexity of vehicle routing and scheduling problems. Networks 11, 221–227.

LEVIN, A. 1971. Scheduling and fleet routing models for transportation systems. Transportation Science, 5(3), 232–256.

LIEBMAN, J. C. 1970. Mathematical models for solid waste collection and disposal 38th national meeting of the Operations Research Society of America. Bulletin of the Operations Research Society of America, 18(2).

M. GENDREAU, G. LAPORTE, J-Y. POTVIN. 1999. Metaheuristics for the vehicle routing problem. Technical Report. G-98-52, GERAD.

MARKS, D. H., & STRICKER, R. 1970. Routing for public service vehicles. ASCE Journal of the Urban Planning and Development Division, 97(UP2), 165–178.

MENDES, J. J. M., GONCALVES, J. F., & RESENDE, M. G. C. 2005. A random key based genetic algorithm for the resource constrained project scheduling problem. AT&T Labs Research Technical Report TD-6DUK2C.

MIN, H. 1989. "The multiple vehicle routing problem with simultaneous delivery and pick-up points," Transportation Research, Vol: 23A, pp. 377-386.

MINGOZZI, A., GIORGI, S., BALDACCI, R. 1999. "An Exact Method for the Vehicle Routing Problem with Backhauls," Transportation Science, Vol: 33, No 3, pp. 315-329.

MONTANÉ, F.A.T, GALVÃO, R.D., 2006. "A Tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service," *Computers & Operations Research*, Vol: 33, pp. 595-619.

NAG, B., GOLDEN, B.L., ASSAD, A. 1988. "Vehicle routing with site dependencies," In: Golden B, Assad A, editors, *Vehicle routing: methods and studies*, Amsterdam: Elsevier, pp. 149-159.

NAGATA, Y. 2007. Edge assembly crossover for the capacitated vehicle routing problem. *Evolutionary Computation in Combinatorial Optimization*, LNCS, 4446, 142–153.

NAGY, G., SALHI, S. 2005. "Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries," *European Journal of Operational Research*, Vol: 162, pp. 126-141.

NORMAN, B. A., & BEAN, J. C. 2000. Scheduling operations on parallel machine tools. *IIE Transactions*, 32(5), 449–459.

O'CONNOR, A. D., & DE WALD, C. A. 1970. A sequential deletion algorithm for the design of optimal transportation Networks. 37th national meeting of the Operations Research Society of America. *Bulletin of the Operations Research Society of America*, 18(1).

OSMAN, I.H. 1993. Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Operations Research*, 41, 421-451.

OSMAN, I.H., ABO-SINNA, M.A., & MOUSE, A.A. (2005). An effective genetic algorithm approach to multiobjective routing problems. *Applied Mathematics and Computation* 162, 769- 781.

PARK, Y.-B., & HONG, S.-C. 2003. A performance evaluation of vehicle routing heuristics in a stochastic environment. *International Journal of Industrial Engineering: Theory Applications and Practice*, 10(4), 435–441.

PISINGER, D., & ROPKE, S. 2007. A general heuristic for vehicle routing problems. *Computers & Operations Research*, 34, 2403–243.

POTVIN, J.Y., BENGIO, S., 1996. The vehicle routing problem with time windows--Part II: Genetic search. *INFORMS Journal on Computing* 8, 165–172.

POTVIN, J.Y., KERVAHUT, T., GARCIA, B.L., ROUSSEAU, J.M. 1996. The vehicle routing problem with time windows--Part I: Tabu search. *INFORMS Journal on Computing* 8, 158– 164.

PRINS, C. 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computer & Operational Research*, 31, 1985-2002.

RALPHS. 2003. Parallel Branch and Cut for Capacitated Vehicle Routing, *Parallel Computing* **29**, 607.

REIMANN,M., DOERNER, K., & HARTL, R. F. 2004. D-ants: savings based ants divide and conquer the vehicle routing problem. *Computers & Operations Research*, 31. 563–591.

ROBERTS, D., & HADJICONSTANTINO, E. 1998. A computational approach to the vehicle routing problem with stochastic demands. In P. Borne, M. Ksouri, A. Elkamel (Eds.), *Computational engineering in systems applications: 16th European Conference on operational research (EURO XVI)*, Brussels, Belgium, IEEE (pp. 139–144).

ROPKE, S., PISINGER, D. 2006. "A unified heuristic for vehicle routing problems with backhauls," *European Journal of Operational Research*, Vol: 171, pp. 750-775.

RYAN, D.M., HJORRING, C. ve GLOVER, F. 1993. Extensions of the Petal Method for Vehicle Routing. *Journal of the Operational Research Society*,44:289-296.

SALHI, S., NAGY, G. 1999. "A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling," *Journal of the Operational Research Society*, Vol: 50, pp. 1034-1042.

SARIKLIS, D., POWELL, S. 2000. "A heuristic method for the open vehicle routing problem," *Journal of the Operational Research Society*, Vol: 51, pp. 564–573.

SCHULZE, J., FAHLE, T., 1999. Parallel algorithm for the vehicle routing problem with time window constraints. In: Beasley, J.E., Sharaiha, Y.M. (Eds.), *Combinatorial Optimization: Recent Advances in Theory and Praxis*. Special issue of *Annals of Operations Research* 86, 585–607.

SNYDER, L. V., & DASKIN, M. S. 2006. A random-key genetic algorithm for the generalized traveling salesman problem. *European Journal of Operational Research*, 174(1), 38–53.

SOLOMON, M. 1983. Vehicle routing and scheduling with time window constraints: Models and algorithms. Technical report, College of Business Admin., Northeastern University, No. 83-42.

SOLOMON, M.M., DESROSIERS, J. 1988. Time window constrained routing and scheduling problems. *Transportation Science* 22, 1–13.

ŞEKER, Ş. 2007. Araç Rotalama Problemleri ve Zaman Pencereleli stokastik Araç Rotalama Probleminde Genetik Algoritma Yaklaşımı. Yüksek Lisans Tezi(Yayınlanmış). Yıldız Teknik Üniversitesi.

TAILLARD, E., BADEAU, P., GENDREAU, M., GUERTIN, F., POTVIN, J.Y. 1997. A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science* 31, 170–186.

TAILLARD. 1993. "Parallel Iterative Search Methods for Vehicle Routing Problems". *Networks* 23, 661-673.

TASGETİREN, M. F., SEVKLİ, M., LIANG, Y. C., & GENÇYILMAZ, G. 2007. Particle swarm optimization algorithm for makespan and total flow time minimization in permutation flowshop sequencing problem. *European Journal of Operational Research*, 177(3), 1930–1947

TAVAKKOLI-MOGHADDAM, R., SAFAEI, N., GHOLIPOUR, Y. 2006. A hybrid simulated annealing for the capacitated vehicle routing problems with the independent tour length. *Applied Mathematics and Computation*, 176, 445-454.

THANGIAH, S.R., NYGARD, K.E., JUELL, P.L., 1991. GIDEON: a genetic algorithm system for vehicle routing with time windows. In: *Proceedings of the 7th Conference on Artificial Intelligence for Applications*, Miami, FL. IEEE Press, New York, pp. 322–328.

THOMPSON and H. N. PSARAFTIS. 1993. "Cyclic Transfer Algorithms for the Multivehicle Routing and Scheduling Problems", *Operations Research* 41:935-946.

TILLMAN, F.A., HERING, R.W. 1971. "A study of look-ahead procedure for solving the multiterminal delivery problem," *Trans. Res.*, Vol: 5, pp. 225–229.

TOTH, P., & VIGO, D. 2003. The granular tabu search and its application to the vehicle routing problem. *INFORMS Journal of Computing*, 15, 333–348.

TOTH, P., VIGO, D. 1997. "An exact algorithm for the vehicle routing problem with backhauls," *Transportation Science*, Vol: 31, pp. 372–385.

TOTH, P., VIGO, D.(eds.). 2002. The Vehicle Routing Problem. SIAM Monographs on Discr. Math. Appl. 9.

TSOU, C.S. & KAO, C.H. 2007. Multi-objective inventory control using electromagnetism-like meta-heuristic. International Journal of Production Research, 2007, 1-16.

V. CERNY. 1985. A thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm. Journal of Optimization Theory and Applications, 45:41-51.

VALENTE M.S., GONCALVES J.F. 2009. A genetic algorithm approach for the single machine scheduling problem with linear earliness and quadratic tardiness penalties. Computers & Operations Research 36. 2707 – 2715.

WOLSEY, L.A. editor. Integer Programming. John Wiley and Sons, 1998.

WU P., KUNG-JIUAN Y., BAU-Y. H. 2007. A Revised EM-like Mechanism for Solving the Vehicle Routing Problems. Second International Conference on Innovative Computing, Information and Control, 2007. ICICIC '07. 181-181.

WU, P., YANG, W. H., & WEI, N. C. 2004. An electromagnetism algorithm of neural network analysis – An application to textile retail operation. Journal of the Chinese Institute of Industrial Engineers, 21(1), 59–67.

YAOHUANG, G., BINGLEI, X., & QIANG, G. 2002. Overview of stochastic vehicle routing problems. Journal of Southwest Jiaotong University (English Edition), 10(2), 113–121.

YU, B., YANG ZHONG-ZHEN, YAO BAOZHEN. 2008. An improved ant colony optimization for vehicle routing problem. European Journal of Operational Research, 196(1), 171-176.

ÖZGEÇMİŞ

İletişim:

Adres: Uludağ Üniversitesi Endüstri Mühendisliği Bölümü Görükle Kampusu,
16059, Bursa.

Tel: 00 90 224 294 20 91

GSM: 00 90 532 411 05 90

Eğitim:

2006-2009 Uludağ Üniversitesi, Bursa, Türkiye.

Endüstri Mühendisliği Bölümü Yüksek Lisans

GPA: 3.50/4.00

Tez konusu: Araç Rotalama Problemleri için Yeni Bir Meta-Sezgisel

Yaklaşım: Elektromanyetik Algoritma

2002-2006 Orta Doğu Teknik Üniversitesi, Ankara, Türkiye.

Çevre Mühendisliği Bölümü Lisans

GPA: 2.88/4.00

İlgi Alanları:

Araç Rotalama Problemleri, Matematiksel programlama, Optimizasyon, Yapay Zeka, Uzman Sistemler, Bulanık Mantık, Sezgisel ve Meta-Sezgisel Algoritmalar, Olasılık ve İstatistik, Bilgisayar Programlama, Simülasyon, Sağlık Sistemleri Yönetimi, Tedarik Zinciri Yönetimi, Temiz Üretim.

Bilgisayar Becerileri:

İşletim Sistemleri: Linux, Windows 98/2000/XP/Vista

Programlama Dilleri: Fortran, C, C++, Java, HTML, Visual Basic, Prolog

Uygulamalar: Visual Studio 6.0, MATLAB, MPL, GAMS, SPSS, Minitab, R, Statistica, Promodel, Net Beans, LaTeX Editor, MS Office, Joomla

TEŐEKKÜR

Bu konuda bana alıŐma firsatı veren ve her tŸrlŸ desteęi saęlayan danıŐman hocam, Prof. Dr. Erdal EMEL'e ve her zaman yanımda olan ve beni destekleyen baŐta babam Prof. Dr. Mustafa YURTKURAN ve annem Prof. Dr. Merih YURTKURAN olmak Ÿzere tŸm aileme, Merve'ye ve arkadaşlarıma teŐekkŸrŸ bir bor bilirim.