

İSTANBUL HALK EKMEK A.Ş. (İHE)'YE AİT ÇOK DEPOLU ARAÇ ROTALAMA PROBLEMİNİN META-SEZGİSEL YÖNTEMLER İLE OPTİMİZASYONU

Emrah ÖNDER

***İstanbul Üniversitesi, İşletme Fakültesi
Sayısal Yöntemler Anabilim Dalı***

ÖZET

Çok Depolu Araç Rotalama Problemi (ÇDARP) son yıllarda oldukça ilgi gören karmaşık bir kombinatoriyal problemdir. ÇDARP birden fazla sayıda depodan birçok müşteriye ürün dağıtımına ait optimum rotaların tasarım problemidir. Toplam kat edilen mesafenin en küçüklenmesi problemin amaç denklemini oluşturmaktadır. Günümüz iş hayatında ürün dağıtım maliyetleri toplam lojistik maliyetlerinin önemli bir bölümünü oluşturmaktadır. Bu çalışmada Genetik Algoritma (GA) ve Parçacık Sürü Optimizasyonu (PSO) sezgisel teknikleri birlikte kullanılarak ÇDARP'nin kümeleme ve araç rotalama alt problemlerini etkin şekilde çözebilen bir sezgisel yöntem önerilmektedir. İstanbul Halk Ekmek (İHE) A.Ş. ucuz, sağlıklı ve yüksek kalitede ekmek ve ekmek ürünleri üretimi ve dağıtımını yapmak üzere İstanbul Büyükşehir Belediyesi bünyesinde kurulan bir ticari organizasyondur. İHE fabrikalarının (Cebeci, Edirnekapı ve Kartal) toplam günlük üretim kapasitesi 1 milyon ekmektir ve İstanbul'un günlük ekmek ihtiyacının %13'unu karşılamaktadır. İHE'de dağıtımını gerçekleştiren 75 araç mevcuttur ve günde 3 kez 1012 müşteriye dağıtım yapılmaktadır. GA ve PSO birlikte kullanılarak İHE A.Ş.'nin çok depolu araç rotalama problemine optimum çözüm aranmıştır ve mevcut durum ile karşılaştırılmıştır.

Anahtar Sözcükler: Araç Rotalama Problemi (ARP), Parçacık Sürü Optimizasyonu (PSO), Genetik Algoritma (GA), Optimizasyon, Meta-Sezgisel Yöntemler

OPTIMIZATION OF MULTI-DEPOT VEHICLE ROUTING PROBLEM OF ISTANBUL HALK EKMEK A.S. (IHE) BY USING META-HEURISTIC METHODS

ABSTRACT

The Multi Depot Vehicle Routing Problem (MDVRP) is a well known and complex combinatorial problem which has received considerable attention in recent years. MDVRP can be described as the problem of designing optimal routes from several depots to a number of customers. The objective is to find a set of routes which minimizes the total distance traveled. In today's business world, transportation cost typically is an important part of the total logistics costs. An efficient heuristic method combining the genetic algorithm (GA) and particle swarm optimization (PSO) is proposed for solving clustering and VRP subproblems of the MDVRP. Istanbul Halk Ekmek (IHE) A.S. is an economical organization established by the İstanbul Great City Municipality for producing and distributing cheap, healthy and high quality bread and floury products. The total daily capacity of the IHE factories (Cebeci, Edirnekapi, Kartal) is the production of more than 1 million breads (13% market share). IHE delivers bread to 1012 customers three times a day with 75 vehicles. The proposed GA-PSO based heuristic technic is used to solve the MDVRP of IHE A.S. and results are compared with the current delivery strategies.

Key words: Vehicle Routing Problem (VRP), Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Optimization, Meta-Heuristic Methods

GİRİŞ

Türkiye hali hazırda yakıt fiyatlarının en yüksek olduğu ülkelerden biridir. Bu sorun Türkiye’de ürün dağıtımını yapan işletmeler için ARP çözümünün önemini arttırmaktadır. Dağıtım yönetiminin etkin ve verimli gerçekleştirilmesi maliyet düşüşlerini doğrudan etkilemektedir. Uygun dağıtım rotalarının elde edilmesi sonucu oluşan kazanımlar, işletmelerin karlılığını arttırmaktadır.

Araç rotalama problemini yöneticilerin, araç sürücülerinin, dağıtım yapılacak bölgeye hakim personelin tecrübesi ve direktifleri ile çözen çok sayıda büyük işletme mevcuttur. Büyük araç filolarına sahip bu işletmelerde rotaların optimumuna yakınlığı bilinmemektedir. Benzin ve motorin fiyatlarının en yüksek olduğu ülkelerden biri olan Türkiye için işletmelerin ARP’ye gereken önemi vermemesi rekabet güçlerini azaltan başlıca unsurlardan biridir.

Uygulamada İstanbul Halk Ekmek A.Ş.’nin (İHE) ürün dağıtımında en uygun rotalar tespit edilmiştir ve mevcut dağıtım stratejileri ile karşılaştırılmıştır.

1.ARAÇ ROTALAMA PROBLEMİ

Gezgin satıcı problemi (GSP) ve araç rotalama problemi (ARP) kombinatoryal optimizasyondaki en önemli problemlerden ikisidir. Lojistik yönetiminin en önemli bileşenleri olan ve 50 yılı aşkın süredir yöneylem araştırması kapsamında çözümünü araştırılan bu problemler birçok önemli algoritmanın geliştirilmesini sağlamıştır. Kesin çözüm yöntemleri, sezgisel çözüm yöntemleri ve meta-sezgisel çözüm yöntemleri ARP ve GSP çözümü için geliştirilen algoritmaların ana başlıklarıdır. Bu çözüm yöntemleri günümüzde birçok farklı kombinatoryal optimizasyon probleminin çözümünde kullanılmaktadır (Laporte, Gilbert, Les Cahiers du Gerad, 2009, s:6).

Araç rotalama problemi (ARP) polinom zamanda deterministik olmayan (NP-Zor) problem türlerinden biridir. Tüm ARP uygulamalarında kesin çözüme ulaşan tek bir çözüm yolu bulunmamaktadır. Probleme kesin bir çözüm bulmamakla beraber yaklaşık sonuç bulabilen sezgisel yaklaşımlar karmaşık bir optimizasyon problem olan araç rotalama problemlerinde sıklıkla kullanılan çözüm yöntemleridir.

ARP’de amaç toplam ulaşım maliyetini veya toplam kat edilen mesafeyi minimize etmektir. Araç kapasitesi gibi kısıtları da dikkate alarak araçların uygun rotalara atanması ile ARP çözümüne ulaşılabilir. Basit ARP yapısında müşteriler $i=2, 3, \dots, n$ olarak indekslenir ve $i=1$ depoyu göstermektedir. Araçlar ise $k=1, 2, \dots, m$ olarak indekslenir. i . müşterinin q_i miktarında talebi vardır. i . ve j . müşteriler arasındaki ulaşım maliyeti c_{ij} ’dir.

k . aracın kapasitesi Q_k ’dir. Tüm müşterilerin ve araçların q_i ve Q_k ’ya göre azalan sırada listelenir. Basit ARP (her aracın tek rotası vardır ve depo ile başlar depo ile biter) tüm müşterilere servis verilirken toplam maliyeti en küçükleyecek şekilde araçların rotalanmasıdır. Basit ARP gerçek hayat uygulamalarında karşılaşılan bazı değişkenleri ve kısıtları (zaman penceresi, rotada harcanan maksimum zaman kısıdı, dağıtım ve

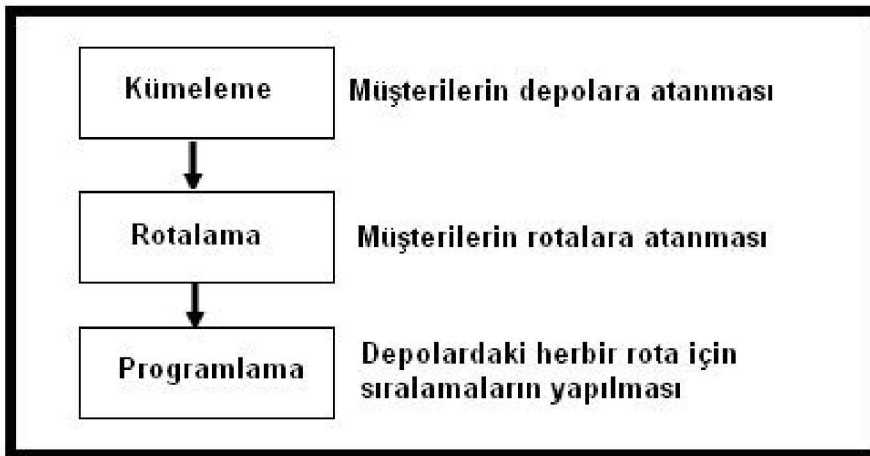
toplamanın eş zamanlı yapılabilmesi, sürücülerin çalışma zamanlarına ait kısıt vb.) dikkate almaz (Lawler, E. L., v.d., 1992, s: 433).

Standart kapasiteli araç rotalama problemlerinde homojen araç filosu tek bir depodan bir grup müşteriye servis vermektedir. Her araç sabit kapasiteye sahiptir ve bu kapasitenin üzerinde yük alınamaz. Her müşterinin ise servis verilmesi gereken bilinen talep miktarları mevcuttur. Her müşteriye bir tek araç tarafından bir kez servis verilmektedir. Her araç depodan yola çıkarak, rotasını tamamladıktan sonra depoya geri dönmelidir. Her aracın kat edebileceği rotaların uzunlukları için kısıtlar konulabilir. Amaç her müşteriye servis verecek şekilde araç filosunun toplam yol alacağı mesafeyi en küçüklerken her araç için dağıtım sırasının bulunmasıdır.

Birden çok depoya sahip işletmelerde, her deponun kendi müşteri portföyüne ve kendi müşteri bölgesine sahip olması sıklıkla rastlanan bir durumdur. Böyle durumlarda işletme çok adet tek depolu ARP problemi ile karşı karşıyadır. Her müşteriye kendi lokasyonuna uygun depodan çıkan araç tarafından servis verilmektedir. Standart ÇDARP’de her araç rotası aynı depodan başlar ve aynı depoda sonlanır (Crevier, B., Cordeau, J. F., Laporte, G., 2007, s:757). ÇDARP çözümü sırasında üç tip karar almak gereklidir. Müşterilerin depolara göre kümelenmesini içeren grüplama problemi bunlardan ilkidir. Aynı depoya atanmış müşteriler rotalara göre filodaki araçlara atanmalıdır. Bu atama sırasında araç kapasiteleri aşılmamalıdır. Son karar problemi ise aynı rotadaki müşterilerin servis alma sıralarının belirlenmesidir. ÇDARP’de kısa dağıtım zamanı müşteri tatminini artırır.

Bunun yanı sıra amaçlardan biri de filodaki araç sayısını minimize etmek olabilir. Araç sayısını azaltmak toplam dağıtım maliyetini de azaltacaktır. Öncelikli amaç işletmeler arasında farklılık gösterse de esas olan ÇDARP’de dağıtım verimliliğini arttırmaktır (Ho, W., v.d., 2007, s:550).

Şekil 1: ÇDARP çözüm süreci



2.GENETİK ALGORİTMA

John Holland 1975 yılında yeni çocuklar oluşturmak üzere ebeveynlerden gelen bilgi parçacıklarını farklı kombinasyonlarda birleştirip genetik algoritma tekniğini geliştirmiştir ve GA terimini literatüre kazandırmıştır. GA'nın başlıca yeniliği çaprazlama adında yeniden oluşum operatörünü kullanmasıdır. Çaprazlama operatörü GA'nın en önemli bileşenlerinden biridir (Aarts, E, Lenstra J.K., 1997, s:322). Çaprazlama operatörü aday çözümleri genetik olarak yeniden yapılandırır.

Sayısal optimizasyonda, permütasyon kodlama, tersine çevirme mutasyonu, n-nokta çaprazlama ve turnuva seçimi sıkça kullanılan yöntemlerdir. Bunlarla birlikte elitlik operatörü GA'nın performansına etki eden önemli unsurlardan biridir. Elitlik kavramının altında yatan mantık her jenerasyondan en iyi değere sahip bireylerin belli bir oranına dokunulmadan yeni kuşaklara aktarılmasıdır. Ebeveyn jenerasyonu her iterasyonda bir çocuk bireye mutasyon uygulanarak çocuk jenerasyonunu oluşturur. Bir somaki iterasyondaki popülasyon değişime uğramayan ebeveynler ve yeni oluşturulmuş çocukların bileşiminden oluşturulur (Paterlini, S., Krink T., 2006, s:1227-1228).

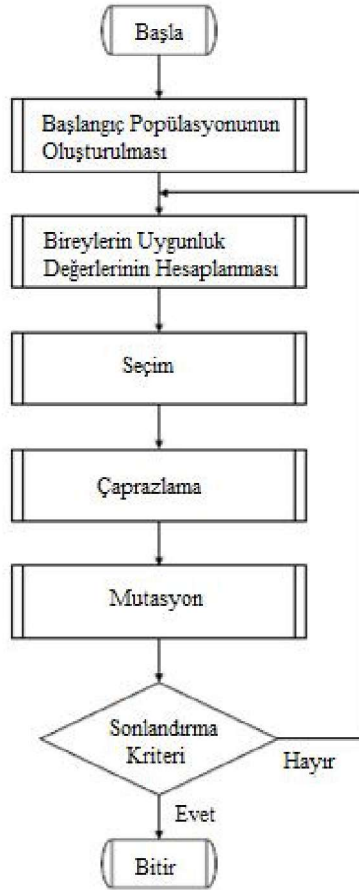
Probleme ait potansiyel çözümler parametreler kümesi olarak sembolize edilir. Gen olarak da bilinen bu parametreler değerler dizgisi biçimini oluşturacak şekilde birleştirilir. Yöneylem araştırması problemlerine göre amaç fonksiyonunda yer alan değişkenlerin her birinin alacağı değerler kromozomlarda temsil edilir. Bir çocuğun kromozomu genlerden oluşur ve bu genler rastlantı olarak ebeveynlerinden aktarılır. Ebeveyn seçiminde bireyin uygunluk değeri önem arz eder. Uygunluk değeri yüksek olan ebeveyn kullanılan seçim yönteminin çalışma prensibi dikkate alınarak seçilme şansını artırır. GA popülasyondaki her çözüm alternatifine ait kromozom dizgisini değerler ve uygunluk değeri elde eder (Khoo, K. G., Suganthan, P. N., 2002, s:1592). Çözülecek her problemde düşünülmesi gereken önemli bileşenlerden biri uygunluk fonksiyonudur. Kromozomlara karşı uygunluk fonksiyonu belli bir uygunluk değerini geri döndürür. Bu değer kromozomlar tarafından temsil edilen bireyin faydası ve yeteneği ile orantılıdır. GA'nın üreme aşamasında bireyler popülasyondan seçilir ve yeni jenerasyonu oluşturacak çocuk bireyleri ortaya çıkarmak üzere tekrar yapılandırılır. İki ebeveynin seçimi soması ebeveynlerin kromozomları çaprazlama ve mutasyon mekanizmaları kullanılarak tekrar yapılandırılır (Beasley,D., Bull D.R., Martin R.R., 1993, s:60). İki birey alınarak kromozom dizgileri rastsal seçilen noktadan kesilir. Böylece iki bireyden iki baş kısmı ve iki kuyruk kısmı olmak üzere dört kromozom parçası elde edilir. Daha soma kuyruk kısımları yer değiştirilerek yeni iki çocuk birey elde edilir. Her çocuk birey genlerinin belli kısımlarını her iki ebeveyninden alır. Bu süreç tek nokta çaprazlama olarak bilinir. Çaprazlama genellikle çiftleştirilmek üzere seçilmiş tüm birey ikililerine uygulanmaz. Çaprazlama GA'larm arama uzayının benzer fakat araştırılmamış bölgelerine ulaşmayı sağlayan bir arama operatörüdür. Gen takası genetik algoritmanın motoru kabul edilir.

Nesiller arasında genetik değişkenliği sağlayan kromozomlardaki bozulma yapısıdır. Çocuk bireylerin bazı genleri rastlantısal olarak değişim gösterir. Genetik algoritmada kullanılan mutasyon operatörü iterasyonlar gerçekleşirken yerel optimum noktalarda çözümün sıkışmasının önüne geçer. Mutasyon genetik algoritmada ikincil rolü

oyunmaktadır (Goldberg, D. E., 1989, s:14). Yerel optimum noktaya ulaşıldığında sonlanan basit komşuluk arama yönteminden farklı olarak GA teoride sonsuza kadar çalışabilen stokastik bir arama yöntemidir. Pratikte ise sonlandırma kistasına ihtiyaç duyulur. En yaygın yaklaşım uygunluk hesaplama işlemlerinin sayısını sınırlandırmak, bilgisayar hesaplama zamanını belirlemek veya popülasyondaki çeşitliliği takip ederek belli bir eşiğin altına düştüğünde algoritmayı sonlandırmaktır (Colin Reeves, 2003, s:64). Popülasyon büyüklüğü yüksek bir değer seçilirse popülasyondaki çeşitlilik fazla olmakta ancak daha fazla uygunluk değeri hesaplanacağından algoritmanın çalışma süresi artmaktadır (Siriwardene, N. R., Perera, B. J. C., 2006, s:417).

Genetik algoritmaya ait akış diyagramı Şekil 2’de gösterilmektedir (Taniguchi, E., ve Shimamoto, H., 2004, s:241).

Şekil 2: Genetik algoritmaya ait akış diyagramı



3.PARÇACIK SÜRÜ OPTİMİZASYONU

1927 yılında Karl Von Frisch arıların yuvalarına dönerken yanlarında nektar ve polen yanı sıra bilgi de getirdiklerini keşfetmiş ve iletişim dillerini çözmeye çalışmıştır Sürüdeki diğer bireylerin çağrularına cevap veren birçok işçi arı başlangıç konumu etrafında rastsal bir noktaya varırlar. Bu yüzden eğer etrafta daha iyi bir noktaya denk gelirlerse o noktaya yaklaşma yönünde hareketler sergilerler (Clerc, M., 2006, s:29). Parçacık sürü optimizasyonu (PSO) 1995 yılında Kennedy ve Eberhart tarafından geliştirilen evrimsel

hesaplama tekniğidir (Şevkli, M., Yenisey, M. M., 2006, s:59). Diğer evrimsel hesaplama tekniklerinde olduğu gibi PSO popülasyon temelli arama algoritmasıdır ve parçacık olarak isimlendirilen rastsal çözüm değerlerinin oluşturduğu popülasyon ile başlatılır. Diğer evrimsel hesaplama tekniklerinden farklı olarak PSO'daki her parçacığa hız bileşeni eşlik etmektedir. Parçacıklar geçmiş davranışlara bağlı dinamik olarak ayarlanan hızları ile arama uzayında uçarlar. Bu yüzden parçacıkların arama sürecinde daha iyi arama bölgelerine doğru uçuş eğilimleri vardır (Yuhui Shi, 1999, s:9)

PSO algoritmasının çıkış noktası kuş ve balık gibi hayvanların sosyal davranışları ve sürü teorisi. PSO algoritmasının ilgi çekici teknikler arasında yer almasının başlıca nedenlerinden birisi ayarlanacak az sayıda parametre içermesidir. PSO algoritmasının performansını birçok araştırmacı değişik veri setlerinde ve uygulamalarda test etmiş ve başarılı sonuçlara ulaşmıştır. Genetik algoritmaya benzer olarak PSO'da da sistem rastgele çözümlerden oluşan popülasyonla başlar ve potansiyel çözüm noktalarını jenerasyon sayısı kadar güncelleyerek optimum sonuca yaklaştırmaya çalışır. Ancak çaprazlama ve mutasyon gibi genetik algortmada bulunan evrimsel operatörler PSO'da bulunmamaktadır. PSO'da parçacık olarak isimlendirilen potansiyel çözüm noktaları o anki daha iyi çözüm noktalarını izleyerek problemin çözüm uzayında “uçarlar”.

Genetik algoritma ile parçacık sürü optimizasyonu arasındaki farklardan en belirgin olanı GA yapısal olarak kesikli özellik gösterir ve değişkenler 0 ve 1'lerden oluşan ikilik sistemde kodlanır. PSO ise yapısal olarak sürekli özellik gösterir ve kesikli değişkenlerde çalışılırken algoritmanın temel yapısının modifiye edilmesi gerekmektedir (Hassan, R., Cohanım, B., Weck, O., 2005, s:2). Parçacık Sürü Optimizasyonunun temelini iki unsur oluşturmaktadır. Bunlar yapay hayat (balık, kuş vb.) ve sürü teorisi (Kennedy, J. ve Eberhart, R. C., 1995, s:1944). PSO'nun orijinal versiyonu çözüme başlangıçta çok hızlı yakınsarken, hassas ayar yakınsamasında yavaştır (Yuhui Shi, a.g.e., s:9).

PSO'da parçacık olarak adlandırılan potansiyel çözümler bireyleri oluşturmaktadır. GA'da olan seçim operatörü PSO'da bulunmamaktadır. Tüm parçacıklar algoritmanın iterasyonları süresince sürüden elenmezler ve sonuç bulunana kadar varlıklarına devam ederler (Shi, Y., Eberhart, R. C., 1999, s:1945). PSO'daki her parçacık D-düzlemlili problem uzayında kendi ve sürüdeki diğer bireylerin uçuş tecrübeleriyle dinamik olarak ayarlanan hızlarda uçmaktadırlar. PSO algoritmasında parçacıklar hız ve konumlarını her iterasyonda değiştirirler. Hız güncelleme denkleminin üç bileşeni bulunmaktadır. Hız güncelleme denklemindeki ilk kısım bir önceki hızın “eylemsizliğini” göstermektedir. Bazı kaynaklarda bu kısım “momentum” kısmı olarak adlandırılır. Hız birden bire değişmez, maddelerin bir eylemsizliği vardır. Mevcut hızdan değişim olmaktadır. İkinci kısım parçacığın kendi kendine düşünmesini ve kendi geçmiş tecrübesini gösteren “idrak” kısmı olarak adlandırılabilir. Üçüncü kısım ise parçacıklar arasındaki işbirliğini temsil eden “sosyalleşme” kısmıdır. Bu kısım ile parçacıklar sürünün uçuş tecrübesinden yararlanırlar. Eğer hızlanmaların toplamı maksimum hızı geçmesine sebep olursa hız maksimum hız ile sınırlanır. Mevcut pozisyon ve hedef pozisyon arasındaki alan dikkate alınarak maksimum hız parametresi kullanıcı tarafından saptanır (Xie, X. F., Zhang, W. J., Yang, Z. L., 2002, s: 1457-1458).

PSO algoritmasının en temel yapısı üç adımdan oluşmaktadır. Birinci adım parçacıkların konumlarının ve hızlarının oluşturulması, ikinci adım hızların güncellenmesi ve son adım konumların güncellenmesidir. Parçacıkları çözüm uzayındaki noktalar olarak düşünülebilir. Parçacıklar konumlarını her iterasyonda (harekette) güncellenen hızları dikkate alarak değiştirirler. Aşağıdaki denklemler parçacıklara başlangıç konumu ve hızı atanırken kullanılabilir (Hassan, R., v.d., a.g.e., s:3). Denklemler popülasyondaki parçacıkların uçuş yansımalarını tanımlamaktadır. Konum denklemi uçan parçacıkların konum güncellemesini göstermektedir. Hız denklemi ise hızın dinamik olarak güncellemesini gösterir.

$$x_0^i = x_{\min} + rand(x_{\max} - x_{\min})$$

(1)

$$v_0^i = \frac{x_{\min} + rand(x_{\max} - x_{\min})}{\Delta_t} = \frac{Konum}{Zaman}$$

(2)

PSO'da hız güncellemek için kullanılan ve üç bileşenden oluşan formül aşağıda gösterilmiştir (Hassan, R., v.d., a.g.e., s:3).

$$v_{k+1}^i = w * v_k^i + c_1 * rand \frac{(p^i - x_k^i)}{\Delta_t} + c_2 * rand \frac{(p_k^g - x_k^i)}{\Delta_t}$$

↙

Eylemsizlik Bileşeni
(Momentum)

↓

(3)

İdrak Bileşeni
(Parçacığın Hareketine Kendi Belleğinin Etkisi)

↘

Sosyalleşme Bileşeni
(Parçacığın Hareketine Sürünün Belleğinin Etkisi)

Hız güncelleme formülündeki bileşenler aşağıda gösterilmektedir:

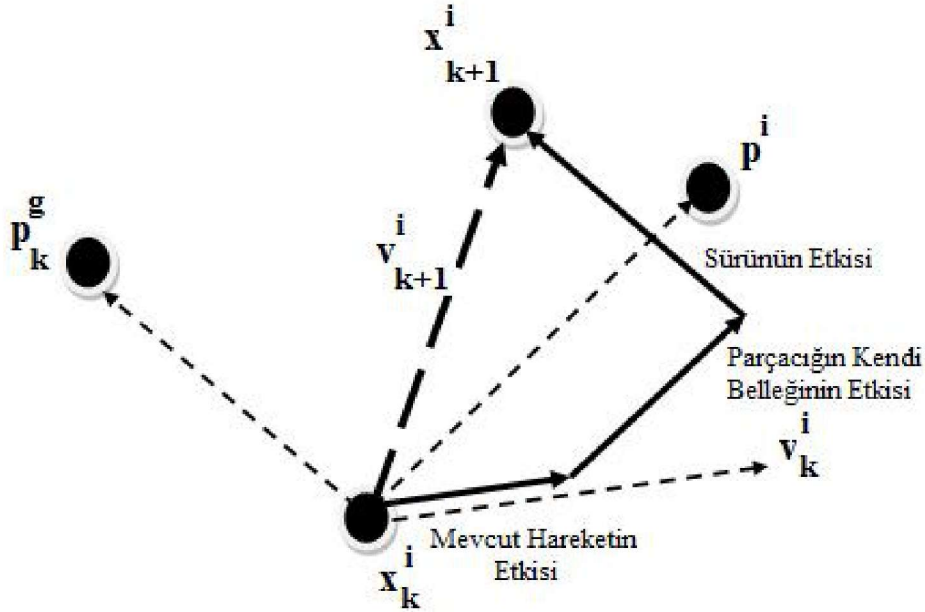
- $v_{k+1}^i = i$. inci parçacığın k+1 zamanındaki hızı
- w = atalet ağırlık katsayısı (0.4-1.4)
- $w * v_k^i$ = Mevcut Hareket (Eylemsizlik)
- c_1 = parçacığın özgüven katsayısı (1.5-2)
- $c_1 * rand \frac{(p^i - x_k^i)}{\Delta_t}$ = Parçacığın Kendi Belleğinin Etkisi (İdrak)
- c_2 =Parçacığın sürüye güvenme katsayısı (2-2.5)
- $c_2 * rand \frac{(p_k^g - x_k^i)}{\Delta_t}$ = Parçacığın Hareketine Sürünün Belleğinin Etkisi

(Sosyalleşme)

Öncelikle başlangıç sürü parçacıklarının konumları x_k^i ve hızları v_k^i değişken değerlerinin alt (x_{\min}) ve üst sınırları (x_{\max}) dikkate alınarak rastsal olarak oluşturulur. Parçacıkların konum ve hızları k alt indisi ve i üst indisi kullanılarak gösterilebilir. i indisi parçacığın numarasını, k indisi ise zamanı göstermektedir. 1. ve 2. eşitliklerde rand() 0-1 arasında

değerler alan tekdüze dağılıma uygun rastsal değişkeni göstermektedir. Başlangıç değerlerinin verildiği bu süreç sürüdeki parçacıkların tanımlanmış çözüm uzayında rastsal dağılımını gerçeklemektedir (Hassan, R., v.d., a.g.e., s:4). Reel sayılardan oluşan çözüm uzayında her mümkün çözüm parçacık olarak modellenilebilir. Her parçacığın pozisyonunu vektör olarak saptanır (Valle, Y.,v.d., 2008, s:173).

Şekil 3: PSO konum güncelleme sürecine ait örnek



İkinci adım parçacıkların k anında çözüm uzayında bulunan parçacıkların mevcut konumlarına ait fonksiyondan elde edilen amaç veya uygunluk değerlerini kullanarak $k+1$ anı için parçacıkların hızlarının güncellenmesidir. Parçacıkların uygunluk fonksiyonu değeri hangi parçacığın en iyi değere p_k^g sahip olduğunu belirlemektedir. Ayrıca her bir parçacığın o ana kadarki kendi en iyi değeri p^i belirlenir. Hız güncelleme formülü bu iki bilgi bileşenini her parçacık için kullanarak ve sürüde mevcut hareketten etkilenecek v_k^i arama yönünden v_{k+1}^i arama yönüne bir sonraki iterasyonda geçer. Hız güncelleme formülü tüm çözüm uzayının kapsanmasını sağlamak ve yerel optimumda sıkışmayı engellemek için tekdüze dağılım parametrelerini temsil eden $\text{rand}()$ ile gösterilen bazı rastsal parametreleri içerir. Hız güncelleme formülünde yeni arama yönünü etkileyen üç bileşene (mevcut hareket, parçacığın kendi belleği ve sürü belleğinin etkisi) üç ağırlık faktörü katsayısı (atalet katsayısı w , özgüven katsayısı c_1 , ve sürü belleğine güven katsayısı c_2) eşlik eder.

PSO algoritmasının basit bir yapısı vardır, kurulumu kolaydır ve hesaplama gücü yüksektir. PSO kurulumunun orijinal yapısına ait farklı gösterim şekilleri vardır. Bu gösterimlerden bir tanesi aşağıda gösterilmiştir:

PSO parametre katsayılarının ve sürü büyüklüğünün değerlerini belirle
 Parçacıkların konumlarının, hızlarının başlangıç değerlerini belirle
 Parçacıkların kendi geçmiş en iyi değerlerinin başlangıç değerini belirle
 Sürüdeki en iyi parçacığı sapta
 Döngü
 Hızları güncelle
 Konumları güncelle
 Parçacıkların kendi en iyi değerlerini sapta
 Sürüdeki en iyi değere sahip parçacığı sapta
 Yerel Arama (isteğe bağlı)
 Durdurma kriteri (Tchomte, S. T., Gourgand, M., 2009, s:58)

Orijinal PSO algoritması sırasıyla w , c_1 ve c_2 değerleri için 1, 2 ve 2 değerlerini kullanmaktadır. Ayrıca bu katsayıların alt ve üst sınırları daha önce belirtilmiştir. Konum güncellemesi her iterasyonun son adımındır. Formül (4)'de görüldüğü gibi her parçacığın konumları hız vektörleri kullanılarak güncellenir (Hassan, R., v.d., a.g.e., s:5).

$$x_{k+1}^i = x_k^i + v_{k+1}^i \Delta t$$

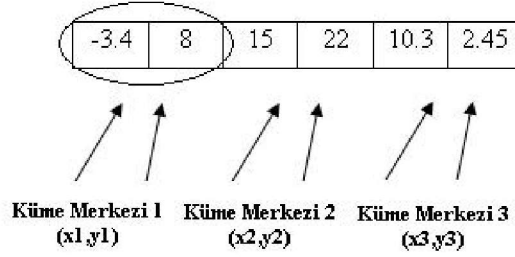
(4)

PSO ile problem çözümünün birçok avantajı mevcuttur. Bu avantajlardan aşağıda sıralanmıştır:

- PSO'nun yapısı basittir ve anlaşılması kolaydır.
- Doğrusal olmayan, türevi alınamayan, çoklu model problemleri için çözüm gücü ve hızı yüksektir.
- Sürekli veya ayrık matematiksel problemlerini optimize edebilme gücü yüksektir.
- Optimum çözüm etrafında tutarlı sonuçlar verir
- Hızlı yakınsar ve popülasyon büyüklüğüne olan hassasiyeti düşüktür (Fealko, D. R., 2005, s:11).

Kümeleme analizinin (KA) çözümündeki yeni tekniklerden biriside parçacık sürü optimizasyonudur. PSO-Kümeleme yöntemi çok boyutlu uzayda kümeleme probleminin çözümünde güçlü bir yöntemdir. n boyutlu Öklid uzayında R^n , N veri noktası K gruba ayrılır ve kümelerin merkezlerine atanır. Her parçacığın kodlanmış değeri reel sayılardan oluşan dizgi ile gösterilir. Bunlar K küme merkezini temsil eder. n boyutlu uzay için her parçacığın uzunluğu $K*n$ 'lik dizgidir. Başlangıç popülasyonu rastsal olarak oluşturulur ve farklı küme merkezleri vektörlerini gösterir. Şekil 4'de bir parçacığın kodlama örneği gösterilmektedir. Bu kodlamada boyut sayısı 2 ($n=2$) ve grup sayısı 3'dür ($K=3$). Bu parçacığın dizgesi 3 küme merkezinin koordinatlarını temsil etmektedir. [(-3,4, 8), (15, 22) ve (10,3, 2,45)]

Şekil 4: PSO'da ki tek bir parçacığın kodlanmış dizgesine ait başlangıç değeri



Parçacıkların kodlanmasından sonra PSO kümeleme algoritması şu adımlardan oluşmaktadır (Chen, C.Y., Ye, F., 2004., s:791):

Adım 1: Popülasyondaki her parçacığa ait X pozisyon vektörünün ve V hız vektörünün başlangıç değerlerinin rastsal olarak atanması. Burada parçacığa ait X değerleri küme merkezlerinin koordinatlarını temsil etmektedir.

Adım 2: Her parçacık için uygunluk fonksiyon değerlerinin elde edilmesi. i. noktanın ($x_i, i = 1, 2, \dots, N$) j. kümeye ($C_j, j \in \{1, 2, \dots, K\}$) eğer $\|x_i - z_j\| < \|x_i - z_p\|, p = 1, 2, \dots, K$ ve $j \neq p$) atanması metodu ile algoritmaya devam edilmesi.

PSO kümeleme algoritması için uygunluk fonksiyonu:

$$J = \sum_{j=1}^K \sum_{i=1}^N \|x_i - z_j\|^2$$

(5)

$$\text{Uygunluk} = k / (J + J_0)$$

(6)

Formüldeki k pozitif sabit bir sayı ve J_0 küçük değere sahip sabit bir sayıdır.

Adım 3: Parçacıkların uygunluk değerleri ile en iyi değerlerinin (P_i) kıyaslanması. Eğer mevcut değer geçmiş en iyi değerden (P_i) daha iyi ise mevcut değer en iyi değer (P_i) olarak atanması. n boyutlu uzayda (P_i) değerinin mevcut değere eşit olması.

Adım 4: Popülasyondaki en iyi uygunluk değeri ile geçmiş en iyi değer kıyaslanması, eğer son popülasyonun en iyi değeri geçmiş global en iyi değerden (P_g) daha iyi ise son popülasyona ait en iyi değer global en iyi değer (P_g) olarak atanması.

Adım 5: Hız ve pozisyon vektörlerinin PSO'nun temel hız ve pozisyon güncelleme formülleri kullanılarak yenilenmesi.

Adım 6: Adım2-Adım5 sürecinin durma kriteri sağlanana ya da önceden belirlenmiş iterasyon sayısına ulaşana kadar sürdürülmesi.

PSO ve K-ortalamlar algoritmalarının ayrı ayrı kümeleme analizinde uygulanıp performansının karşılaştırıldığı akademik çalışmaların yanında karar destek sistemlerinde kullanıcıya her iki opsiyonu da sunan çalışmalarda mevcuttur. Örneğin Chiu v.d. pazar segmentlerini belirleme karar destek sisteminde PSO+K-Ortalamlar ve K-Ortalamlar'm hangisinin kümeleme analizinde kullanılacağını kullanıcıya bırakmışlardır (Chiu, C.Y., v.d., 2009, s:4560).

4.UYGULAMA

Bu çalışmada parçacık sürü optimizasyonu ve genetik algoritma meta-sezgisel yöntemleri kullanılarak ARP çözümü geliştirilmiştir. Uygulamada İstanbul ekmek ihtiyacının %13'ünü karşılayan İstanbul Halk Ekmek A.Ş.'nin (İHE) ürün dağıtımında en uygun rotalar tespit edilmiştir. İHE'nin Edirnekapı, Cebeci ve Kartal olmak üzere üç fabrikası, bu fabrikalarda eş zamanlı dağıtım yapan toplam 75 aracı ve bu araçların dağıtım yaptığı 1012 satış noktası mevcuttur. Günde üç kez dağıtım yapan bu büyük şirkette yapılacak ARP çalışması sonucu önemli kazanımların elde edilebileceği öngörülmüştür.

Cebeci merkez fabrikasına bağlı 323 dağıtım noktası bulunmaktadır ve bu müşterilere dağıtım yapan 21 araç mevcuttur. Edirnekapı fabrikasına bağlı 452 dağıtım noktası bulunmaktadır ve 40 araç ile bu müşterilere servis verilmektedir. Kartal fabrikasında 14 araç 237 müşteriye dağıtım yapmaktadır. Edirnekapı fabrikasında dağıtım yapan 40 araç mevcuttur. Kullanılan 75 araçta kurallar gereği standarttır. Araçlar 100 km'de 60 TL yakıt tüketmektedir. Edirnekapı fabrikasına bağlı 40 rotada toplam kat edilen mesafe 1261.6 km'dir. Kartal fabrikasına bağlı 14 rotada toplam kat edilen mesafe 822.7 km'dir. Cebeci (Gaziosmanpaşa) fabrikasına bağlı 21 rotada toplam kat edilen mesafe 1911.2 km'dir. 1911,2 km ile en uzun seyahat mesafesinin Cebeci fabrikasına ait dağıtım rotalarında olmasının sebebi fabrikanın çok geniş bir alana dağıtım yapmasıdır. Kilyos, Durusu, Büyükçekmece v.b. birbirinden oldukça uzak müşteri lokasyonları Cebeci fabrikasının satış sahasına girmektedir. Mevcut rotalara ait özet bilgi aşağıdaki tabloda gösterilmiştir. Mevcut durumda İHE A.Ş.'ye ait 75 araç bir seferlik dağıtımda toplamda 3995,5 km yol kat etmektedir.

Tablo 1: Mevcut rotalara ait bilgiler

	Müşteri Sayısı	Araç Sayısı	Toplam Mesafe (km)	Yakıt Tutarı (TL)
Edirnekapı Fab. (E)	452	40	1261,6	756,96
Kartal Fab. (K)	237	14	822,7	493,62
Cebeci Fab. (C)	323	21	1911,2	1146,72

Matlab ile yazılan programda amaç, çok depolu ARP'nin tüm dağıtım noktalarını ele alıp en uygun küme merkezlerini, başka bir deyişle en uygun dağıtım merkezi lokasyonlarının belirlenmesidir. Böylece çok depolu ARP'de kat edilen toplam mesafeyi en küçükleyen optimum küme merkezleri PSO yöntemi kullanılarak bulunabilir. Programda yer alan PSO kavramları ve kümeleme analizi ile bağlantıları aşağıda belirtilmektedir.

Tablo 2: Parçacık Sürü optimizasyonu kontrol parametreleri

Parçacık Sürü Optimizasyonu Parametre Değerleri	
Parametre	Parametrenin Değeri
Popülasyon büyüklüğü	50
İterasyon sayısı	4000
Atalet ağırlık katsayısı	0.7
Parçacığın Özgüven Katsayısı	1.47
Parçacığın Sürüye Güvenme Katsayısı	1.47
Durdurma Kriteri	İterasyon Sayısı

Matlab programlama dilinde yazılan kodun girdi ve çıktıları aşağıda gösterilmektedir.

Programın Girdileri:

- İterasyon sayısı
- Atalet ağırlığı katsayısı
- Parçacığın özgüven katsayısı
- Parçacığın sürüye güvenme katsayısı
- Küme sayısı
- Tüm düğümlerin GPS koordinatları

Programın Çıktıları:

- Belirlenen küme merkezi GPS değerleri (merkezler)
- Düğümlerin belirlenen kümelerden hangisine atandığını gösteren atama matrisi
- Düğümlerin atandıkları küme merkezlerine olan Öklid uzaklıklarının toplamı.

PSO ile iki boyutlu düzlemde yapılan kümeleme analizine ait kodlamaya parçacıklara başlangıç değerleri verilerek başlanır. Sürüdeki 50 parçacığın başlangıç değerleri rastsal olarak verilmiştir. Ancak alt ve üst sınırlar olarak minimum ve maksimum değere sahip düğüm koordinatları alınmıştır.

Daha sonra iterasyon sayısı, atalet ağırlığı katsayısı, parçacığın özgüven katsayısı, parçacığın sürüye güvenme katsayısı, sürü büyüklüğü girdi parametreleri belirlenir. Müşterilere ait GPS koordinatları matris yapısında sisteme tanıtılır.

İterasyon boyunca parçacıkların X ve Y boyutlarının konumları güncellenir, küme merkezlerinin dağıtım noktalarına olan uzaklıkları bulunur. Parçacıkların kendine en yakın olan küme merkezlerine olan uzaklıklarının toplamı hesaplanır. Eğer bulunan değer o ana kadarki en düşük değer ise kaydedilir. Daha sonra parçacıklara ait hız vektörleri güncellenir. Parçacıklar kendine en yakın olan küme merkezlerine atanır ve merkezlere olan Öklid uzaklıkları toplamı hesaplanır. İterasyonlar tamamlandığında küme merkezi değerleri, noktaların hangi merkeze atandığı ve uygunluk değeri olan toplam Öklid uzaklığı çıktıları elde edilir.

Matlab programlama dilinde kodlanan yazılım tek depolu ARP için geliştirilmiştir. Yazılımın girdi ve çıktıları aşağıda belirtilmektedir.

Programın Girdileri:

- Araç sayısı
- Her rotada ziyaret edilecek minimum müşteri sayıları
- Popülasyon büyüklüğü
- İterasyon sayısı
- Depo ve diğer düğüm noktalarına ait uzaklıklar matrisi

Programın Çıktıları:

- En uygun çözüme ait mesafeler toplamı
- En uygun rota
- En uygun kırılma noktaları

Matlab ile yazılan programda amaç, ARP’de kat edilen toplam mesafeyi en küçükleyen sonucu GA yöntemi kullanarak bulmaktır. Programda yer alan GA kavramları ve ARP ile bağlantıları aşağıda belirtilmektedir.

Tablo 3: Genetik algoritma kontrol parametreleri

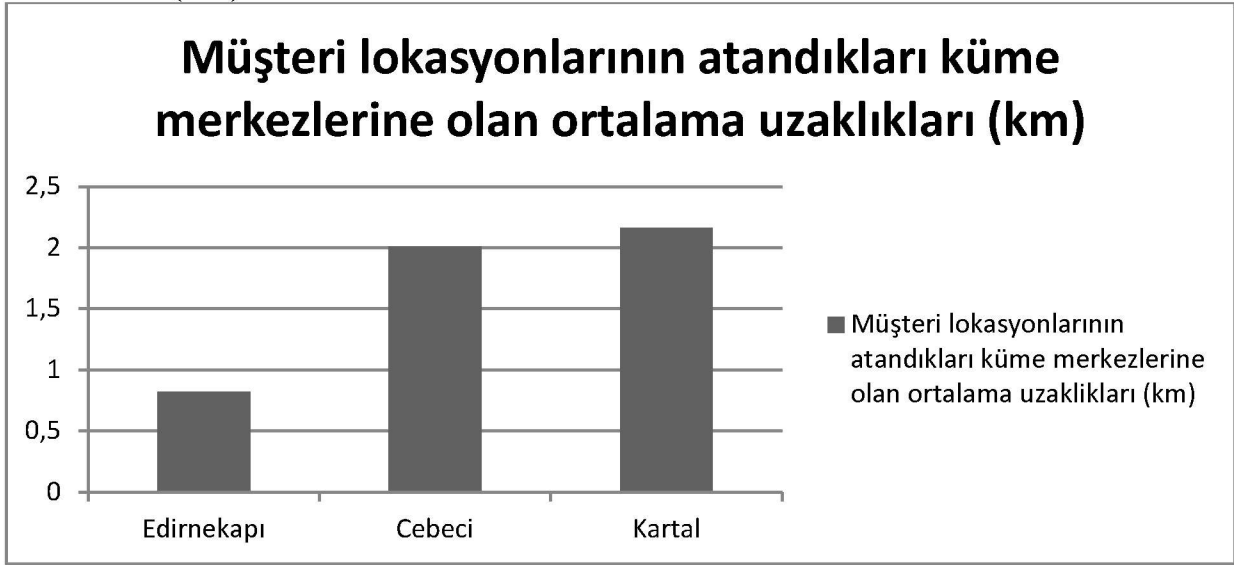
Genetik Algoritma Parametre Değerleri	
Parametre	Parametrenin Değeri
Popülasyon büyüklüğü	160
İterasyon sayısı	10000
Mutasyon Türleri	Rastsal değişim mutasyonu Kaydırma mutasyonu Tersine çevirme mutasyonu
Durdurma Kriteri	İterasyon Sayısı
Seçim stratejisi	Turnuva Seçimi

Belirlenen araç sayısından bir çıkartılarak kırılma sayısı elde edilir. Rotalara ait popülasyon ve kırılmalara ait popülasyon olmak üzere iki farklı popülasyon vardır. Bu popülasyonların başlangıç değerleri rastsal olarak belirlenir. Programda kullanılacak ara değişkenlerin boyutları belirlenir. Popülasyondaki bireylerin uygunluk değerlerinin elde edilmesini takip eden aşamada popülasyon içindeki en uygun rota bulunur. Yeni çözümün oluşturulmasında rastsal değişim mutasyonu, kaydırma mutasyonu ve tersine çevirme mutasyonu farklı kromozomlara uygulanır.

Araç sayısı 1 olarak atandığında ARP çözümü için hazırlanan kod GSP çözümünü vermektedir. Bu durumda kırılmalar matrisi çıktılarda herhangi bir değer almaz.

PSO ile kümeleme analizi yapıldığında müşteri lokasyonlarının atandıkları küme merkezlerine olan ortalama uzaklıkları Edirnekapı Fabrikası için 0.821 km, Cebeci Fabrikası için 2.011 km ve Kartal Fabrikası için 2.161 km olarak bulunmuştur. Bu değerler aşağıdaki grafikte gösterilmiştir.

Şekil 5: Müşteri lokasyonlarının atandıkları küme merkezlerine olan ortalama uzaklıkları (km)

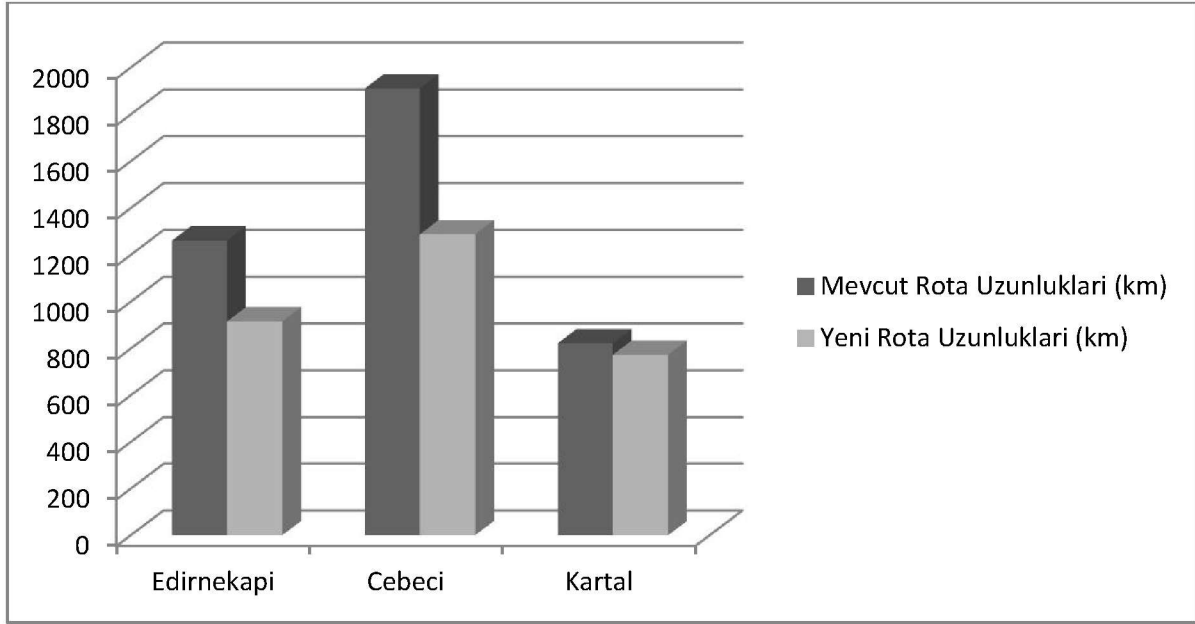


Mevcut rotalarda toplam rota uzunluğu 3995.5 kilometreyken, optimizasyon çalışması sonucu bu değerde 1017.4 km'lik iyileşme sağlanmıştır. Mevcut rota uzunlukları ve yeni rota uzunluklarına ait özet Tablo 4'de gösterilmiştir.

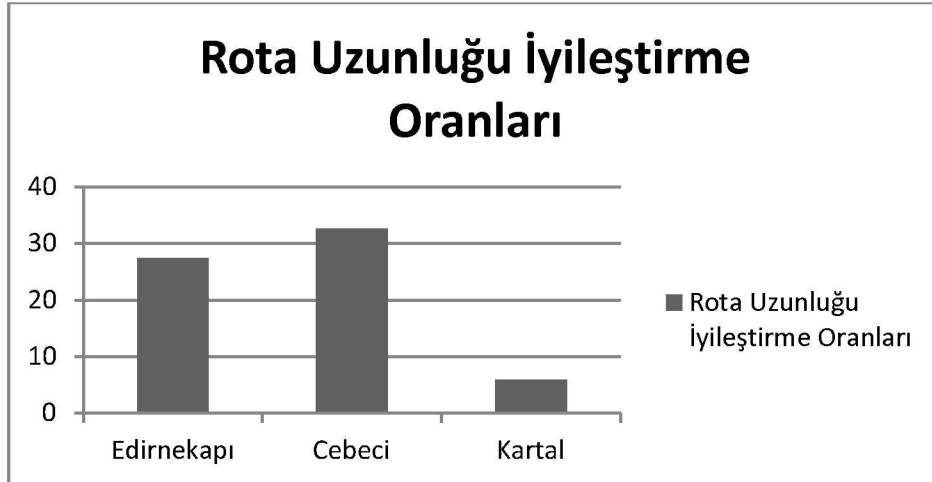
Tablo 4: Mevcut rota uzunlukları ve yeni rota uzunluklarına ait özet tablo

	Mevcut Rota Uzunluğu (km)	Yeni Rota Uzunluğu (km)	Kazanım (km)	Yakıt Kazanımı (TL)
Edirnekapı Fabrikası	1261,6	916	345,6	207,36
Cebeci Fabrikası	1911,2	1288,3	622,9	373,74
Kartal Fabrikası	822,7	773,8	48,9	29,34
TOPLAM	3995,5	2978,1	1017,4	610,44

Her üç fabrikaya ait mevcut rota uzunlukları ve yeni rota uzunlukları Şekil 6'da gösterilmiştir.

Şekil 6: Her üç fabrikaya ait mevcut rota uzunlukları ve yeni rota uzunlukları

Günde üç kez dağıtım yapan bu büyük şirkette mevcut araç sayıları ile yeni rotalama yapıldığında Kartal Fabrikası'nda %5.94'lik, Edirnekapi Fabrikası'nda %27.39'luk ve Cebeci Fabrikası'nda %32.59'luk iyileşme oranları elde edilmiştir. Bu oranlar aşağıdaki grafikte gösterilmiştir.

Şekil 7: Rota uzunluğu iyileştirme oranları

SONUÇ

Parçacık Sürü Optimizasyonu ve Genetik Algoritma bu çalışmada çok depolu araç rotalama problemine uygulanmıştır. Genetik Algoritma ve Parçacık Sürü Optimizasyonu meta-sezgisel teknikleri birlikte kullanılarak Çok Depolu Araç Rotalama Probleminin kümeleme ve araç rotalama alt problemlerini etkin şekilde çözebilen bir meta-sezgisel yöntem önerilmiştir. İstanbul'da üç fabrikası, 75 dağıtım yapan aracı ve 1012 müşterisi bulunan İstanbul Halk Ekmek A.Ş.'ye ait gerçek verilerle algoritmalar çalıştırılmış ve sonuçlar raporlanmıştır.

Araç rotalama probleminde ürün dağıtım yapılacak müşteri sayısı arttıkça problemin zorluk derecesi üstel olarak artmaktadır. Uygulama için seçilen şirketin müşteri sayısı göreceli olarak oldukça fazladır. Parçacık sürü optimizasyonu bu zorluğun aşılmasında kullanılmıştır. PSO ile tüm müşteriler araç (rota) sayısı kadar kümeye ayrılmıştır ve daha küçük boyutlu alt problemler elde edilmiştir. 1012 müşteriye ait en uygun rotaların tek hamlede belirlenmesi yerine 10-25 müşteriye ait uygun rotaların belirlenmesi çok daha kolay yapılabilir. Böylece çok depolu araç rotalama problemi rota sayısı adedinde gezgin satıcı problemine indirgenmiş olur. Bu alt problemlerin boyutu çok küçük olduğu için meta-sezgisel yöntemlerin birçoğu çözümde yakın sonuçlar verebilmektedir. Uygulamada bu aşamada genetik algoritma kullanılmıştır.

Parçacık sürü optimizasyonu diğer meta-sezgisel yöntemlere göre yeni tekniklerden biridir. Yaklaşık 15 yıllık bir geçmişi olan parçacık sürü optimizasyonu bu çalışmada müşterilerin kümeleneşi aşamasında kullanılmıştır. Parçacık sürü optimizasyonu kurulumu, kodlanması ve anlaşılması kolay bir algoritmadır. PSO ile kümelemenin diğer kümeleme algoritmalarına göre en büyük avantajı başlangıç küme merkezi değerlerine bağımlılığının az olmasıdır. Bunun yanı sıra PSO'da ayarlanacak az sayıda kontrol parametresi bulunmaktadır.

Uygulamada İstanbul Halk Ekmek A.Ş.'nin (İHE) ürün dağıtımında en uygun rotalar tespit edilmiştir. İHE'nin Edirnekapı, Cebeci ve Kartal olmak üzere üç fabrikası, bu fabrikalarda eş zamanlı dağıtım yapan toplam 75 aracı ve bu araçların dağıtım yaptığı 1012 satış noktası mevcuttur. PSO ile kümeleme analizi yapıldığında müşteri lokasyonlarının atandıkları küme merkezlerine olan ortalama uzaklıkları Edirnekapı Fabrikası için 0.821 km, Cebeci Fabrikası için 2.011 km ve Kartal Fabrikası için 2.161 km olarak bulunmuştur. Günde üç kez dağıtım yapan bu büyük şirkette mevcut araç sayıları ile yeni rotalama yapıldığında Kartal Fabrikası'nda %5,94'lik, Edirnekapı Fabrikası'nda %27,39'luk ve Cebeci Fabrikası'nda %32,59'luk iyileşme oranları elde edilmiştir. Mevcut rotalarda toplam rota uzunluğu 3995,5 kilometreyken, optimizasyon çalışması sonucu bu değerde 1017,4 km'lik iyileşme sağlanmıştır. Günde üç kez ve ayda 26 gün ürün dağıtımının yapıldığı İHE A.Ş.'de bir yılda 952.286 kilometre kazanım elde edilmiştir. Bu kilometre tasarrufu sadece yakıt kazanımı olarak 571.372 TL değerindedir. Yakıt kazanımına ek olarak araçların yıpranma payı, sürücü yevmiyesi v.b. maliyet azalışları sağlanmaktadır. Dağıtım rota uzunluklarının azaltılması ile kaza riski ve zararlı gaz salınımı azalmaktadır.

Sektör incelendiğinde birçok işletmede rotalama çalışmaları sevkiyat ve pazarlama bölümünde çalışan personelin veya araç sürücülerinin dağıtım bölgesine olan hâkimiyetleri ve tecrübeleri doğrultusunda gerçekleşmektedir. Analitik ve algoritmaya bağlı çalışmalar sonucu elde edilebilecek kazanımlar göz ardı edilmektedir. Bazı şirket yöneticileri bu konuda yapılacak çalışmaların getireceği kazanımları öngörememektedir. Kimi yönetici ise böyle bir ihtiyacın farkında olduğu halde çözüm geliştirecek gruplara ya da danışmalara ulaşamamaktadır. Bu konuda profesyonel çalışma grupları kurularak işletmelere büyük kazanımlar sağlanabileceği düşünülmektedir.

KAYNAKÇA

Aarts, E, Lenstra J.K., **“Local Search In Combinatorial Optimization”**, John Wiley & Sons Ltd, 1997

Beasley,D., Bull D.R., Martin R.R., **“An overview of genetic algorithms”**, University of Computing, 1993, 15(2), s:58-65

Chen, C.Y., Ye, F., **“Particle Swarm Optimization Algorithm and Its Application to Clustering Analysis”**, Proceedings of the 2004 IEEE, International Conference of Networking, Sensing and Control, Taipei, Taiwan, March, 2004, s:789-795

Chiu, C.Y., Chen, Y.F., Kuo, I.T., Ku, H.C., **“An Intelligent market segmentation system using k-means and particle swarm optimization”**, Expert Systems with Applications, 36, 2009, s: 4558-4565

Clerc, M., **“Particle Swarm Optimization”**, 2006, UK, Iste Publication

Colin Reeves, **“Handbook of Metaheuristics, Chapter 3: Genetic Algorithm”**, Derleyiciler: Glover, F., Kochenberger, G.A., Kluwer Academic Publishers, 2003, Dordrecht

Crevier, B., Cordeau, J. F., Laporte, G., **“The multi-depot vehicle routing problem with inter-depot routes”**, European Journal of Operational Research, 176, 2007, s:756-773

Fealko, D. R., **“Evaluating particle swarm intelligence techniques for solving university examination timetabling problems”**, Graduate School of Computer and Information Sciences, Nova Southeastern University, Doktora tezi, 2005, s:11

Goldberg, D. E., **“Genetic Algorithms in Search, Optimization and Machine Learning,”** Addison-Wesley, 1989

Hassan, R., Cohanım, B., Weck, O., **“A Comparison of Particle Swarm Optimization and The Genetic Algorithm”**, Structural Dynamics & Materials Conference 18-21 April 2005, Austin, Texas. s:1-13

- Ho, W., v.d., **“A Hybrid genetic algorithm for the multi-depot vehicle routing problem”**, Engineering Application of Artificial Intelligence, 2007, s:550
- Kennedy, J. and Eberhart, R. C., **“Particle swarm optimization”**, Proc. of the IEEE International Conference on Neural Networks, vol.IV, 1995, s:1942-1948
- Khoo, K. G., Suganthan, P. N., **“Evaluation of genetic operators and solution representations for shape recognition by genetic algorithms”**, Pattern Recognition Letters, 23, 2002, s:1589-1597
- Laporte, Gilbert, Les Cahiers du GERAD, **“The TSP, the VRP and their impact on combinatorial optimization”**, G-2009-57, s:6
- Lawler, E. L., Lenstra, J.K., Rinnooy Kan, A.H.G, Shmoys, D. B., **“The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization ”**, 1992
- Paterlini, S., Krink T., **“Differantial evolution and particle swarm optimization in partitional clustering”**, Computational Statistics & Data Analysis, 50, 2006, s:1220-1247
- Shi, Y., Eberhart, R. C., **“Emprical Study Of Particle Swarm Optimization”**, Proceedings of the Congress on evolutionary Computation (CEC99) (July), 1999, s: 1945-1950
- Siriwardene, N. R., Perera, B. J. C., **“Selection of genetic algorithm operators for urban drainage model parameter optimization”**, Mathematical and Computer Modelling, 44, 2006, s:415-429
- Şevkli, M., Yenisey, M. M., **“Atölye tipi çizelgeleme problemleri için parçacık sürü optimizasyonu yöntemi”**, İTÜ Dergisi, Mühendislik, Cilt:5, Sayı:2, Kısım:1, Nisan 2006, s:58-68
- Taniguchi, E., and Shimamoto, H., **“Intelligent transportation system based dynamic vehicle routing and scheduling with variable travel times”**, Transportation Research Part C: Emerging Technologies, Volume 12, Issues 3-4, June-August 2004, s: 235-250
- Tchomte, S. T., Gourgand, M., **“Particle Swarm Optimization: A study of particle displacement for solving continuous and combinatorial optimization problems”**, Int. J. Production Economics, 121, 2009, s:57-67
- Valle, Y.,v.d., **“Particle Swarm Optimization:Basic Concepts, Variants and Applications in Power Systems”**, IEEE Transactions on Evolutionary Computation, Vol:12, No:2, April, 2008, s:173
- Xie, X. F., Zhang, W. J., Yang, Z. L., **“A Dissipative Particle Swarm Optimization”**, Congress on Evolutionary Computation (CEC), Hawaii, USA, 2002, s:1456-1461