



**İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**DOKTORA TEZİ**

**SÜRÜ ZEKÂSI OPTİMİZASYON TEKNİĞİ VE TEDARİK  
ZİNCİRİ YÖNETİMİNDE BİR UYGULAMA**

**Tarık KÜÇÜKDENİZ  
Endüstri Mühendisliği Anabilim Dalı  
Endüstri Mühendisliği Programı**

**Danışman  
Prof. Dr. Şakir ESNAF**

**Haziran, 2009**

**İSTANBUL**



**İSTANBUL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ**

**DOKTORA TEZİ**

**SÜRÜ ZEKÂSI OPTİMİZASYON TEKNİĞİ VE TEDARİK  
ZİNCİRİ YÖNETİMİNDE BİR UYGULAMA**

**Tarık KÜÇÜKDENİZ  
Endüstri Mühendisliği Anabilim Dalı  
Endüstri Mühendisliği Programı**

**Danışman  
Prof. Dr. Şakir ESNAF**

**Haziran, 2009**

**İSTANBUL**

Bu çalışma 17 / 06 / 2009 tarihinde ařağıdaki jüri tarafından Endüstri Mühendisliğı Anabilim Dalı Endüstri Mühendisliğı programında Doktora Tezi olarak kabul edilmiştir.

Tez Jürisi

Prof. Dr. Şakir ESNAF  
İstanbul Üniversitesi  
Mühendislik Fakültesi

Prof. Dr. Ekrem MANİSALI  
İstanbul Üniversitesi  
Mühendislik Fakültesi

Prof. Dr. Güneş GENÇYILMAZ  
İstanbul Kültür Üniversitesi  
İktisadi ve İdari Bilimler Fakültesi

Prof. Dr. Semra BİRGÜN  
İstanbul Ticaret Üniversitesi  
Mühendislik ve Tasarım Fakültesi

Doç. Dr. Alp BARAY  
İstanbul Üniversitesi  
Mühendislik Fakültesi

**1.**

## **ÖNSÖZ**

Bu çalışmamı sunarken, akademik hayatım boyunca bana her türlü konuda destek olan ve çalışmalarımı yönlendirmemde yardımcı olarak yeni bilgilere ulaşma ve üretme fırsatını sağlayan danışmanım Prof. Dr. Şakir ESNAF' a, ayrıca, akademisyenliğe başlamamdan itibaren yardım ve önerilerini hiçbir zaman esirgemeyen Prof. Dr. Ekrem MANİSALI'ya, bilginin paylaşıldığı huzurlu bir çalışma ortamı sağlayan değerli çalışma arkadaşlarıma teşekkürlerimi borç bilirim.

Hayatları boyunca tüm imkanlarını çocukları için seferber eden, varlıklarından her zaman güç aldığım anne ve babama saygı ve şükranlarımı sunarım.

Son olarak, doktora çalışmam boyunca manevi desteğini sürekli hissettiğim, zor zamanlarımda yanımda olan sevgili Banu ATEŞEDURAN' a teşekkür ederim.

**Mayıs, 2009**

**Tarık KÜÇÜKDENİZ**

## İÇİNDEKİLER

<b>ÖNSÖZ</b> .....	<b>1</b>
<b>İÇİNDEKİLER</b> .....	<b>2</b>
<b>ŞEKİL LİSTESİ</b> .....	<b>4</b>
<b>TABLO LİSTESİ</b> .....	<b>5</b>
<b>SEMBOL LİSTESİ</b> .....	<b>6</b>
<b>ÖZET</b> .....	<b>7</b>
<b>SUMMARY</b> .....	<b>8</b>
1. GİRİŞ 1	
1.2. KATKILAR.....	2
1.3. TEZİN ANA HATLARI.....	3
2. GENEL KISIMLAR 4	
2.1. TEDARİK ZİNCİRİ YÖNTEİMİ.....	4
2.1.1. Tedarik Zinciri Kavramı .....	4
2.1.2. Tedarik Zinciri Yönetimi Kavramı .....	5
2.1.3. Tedarik Zinciri Yönetimi Bileşenleri.....	6
2.2. TEDARİK ZİNCİRİNİN MODELLENMESİ.....	6
2.2.1. Tedarik Zinciri Problemleri .....	7
2.3. TEDARİK ZİNCİRİ YÖNETİMİ MODELLERİ.....	9
2.3.1. Kavramsal Modeller.....	9
2.3.2. Analitik Modeller.....	10
2.3.3. Yapay Zekâ Tabanlı ve Sezgisel Modeller .....	11
2.3.4. Simülasyon Modelleri.....	12
2.4. TESİS YERİ SEÇİMİ PROBLEMİ.....	13
2.4.1. Kapasite Kısıtsız Çoklu Tesis Yeri Seçimi Problemleri.....	15
2.4.2. Çoklu Tesis Yeri Seçimi Problemlerinde Öbekleme Analizi.....	16
2.4.3. Talep Noktalarının Öbeklemesi İçin Bulanık C-Ort. Algoritması .	18
2.4.4. Talep Noktalarının Öbeklemesi İçin Gustafson – Kessel Alg. ....	20

2.4.5.	Tesis Yerlerinin Tespiti için Melez Yöntemler .....	23
2.4.6.	Bulanık Öbeleme - Ağırlık Merkezi Melez Yöntemi.....	23
2.4.7.	Bulanık Öbeleme - Konveks Programlama Melez Yöntemi .....	24
2.4.8.	Ulaştırma Maliyetinin Hesaplanması .....	25
2.5.	SÜRÜ ZEKASI .....	26
2.6.	PARÇACIK SÜRÜ OPTİMİZASYONU YÖNTEMİ.....	28
2.6.1.	Çıkış Noktası .....	28
2.6.2.	Standart Algoritma .....	30
2.6.3.	Uygulama Alanları .....	36
2.6.4.	PSO ile Evrimsel Hesaplama Yöntemleri Arasındaki İlişki.....	39
2.6.5.	Performansı Artırmaya Yönelik Düzenlemeler.....	39
2.6.6.	1' Tip Parçacık Sürü Optimizasyonu Modeli.....	42
2.6.7.	Sürünün En İyisi İçin Önerilen Yöntemler: .....	43
2.6.8.	PSO Tabanlı Öbeleme Analizi Yöntemleri .....	46
2.6.9.	PSO Tabanlı Tesis Yeri Seçimi Yöntemleri .....	48
3.	MALZEME VE YÖNTEM 50	
3.1.	ODAK BİREYLİ SÜRÜ OPTİMİZASYONU ÖBEKLEME YÖNTEMİ.....	50
3.1.1.	Modelin Yapısı .....	51
3.1.2.	Başlangıç İşlemleri.....	53
4.	BULGULAR 57	
4.1.	V İndeksi .....	58
4.2.	Dunn İndeksi .....	59
4.3.	Test Veri Kümelerinde Sonuçlar .....	60
4.3.1.	I. Test Veri Kümesi (IRIS Veri Kümesi) .....	60
4.3.2.	II. Test Veri Kümesi .....	63
4.3.3.	III. Test Veri Kümesi .....	69
4.3.4.	Tesis Yeri Seçimi Probleminde OBSO Yöntemini uygulanması.....	73
5.	TARTIŞMA VE SONUÇ 77	
	KAYNAKLAR 82	
	ÖZGEÇMİŞ 90	

## ŞEKİL LİSTESİ

Şekil 2.1	: Tedarik zinciri boyunca bilgi, malzeme ve para akışı.....	6
Şekil 2.2	: Tedarik Zinciri Problemlerinin Sınıflandırılması.....	8
Şekil 2.3	: BCO algoritmasının amaç fonksiyonu parametreleri.....	19
Şekil 2.4	: BCO algoritmasının işleyişi.....	20
Şekil 2.5	: Ağırlık merkezi yönteminin amaç fonksiyonu ve parametreleri.....	23
Şekil 2.6	: Konveks programlama yönteminin amaç fonksiyonu parametreleri.....	25
Şekil 2.7	: Toplam ulaştırma maliyeti fonksiyonunun parametreleri.....	25
Şekil 2.8	: Balık ve kuş sürüleri.....	26
Şekil 2.9	: Bilginin yayılımı.....	33
Şekil 2.10	: Sürü Zekası Optimizasyon Algoritması.....	36
Şekil 2.11	: Sürü zekası optimizasyon tekniği ile ilgili yayınlanmış çalışma sayısı.....	37
Şekil 2.12	: Atalet katsayısının çeşitli değerleri için başarısız deney sayısı.....	40
Şekil 2.13	: Komşuluk ilişkisi olarak halka yapısı.....	45
Şekil 2.14	: Komşuluk ilişkisi olarak yıldız yapısı.....	45
Şekil 2.15	: Komşuluk ilişkisi olarak odak yapısı.....	45
Şekil 3.1	: $t=13$ olan bir modelde OBSO yönteminin komşuluk yapısı.....	51
Şekil 3.2	: $t=0$ anında OBSO algoritmasında komşuluk yapısı.....	51
Şekil 3.3	: OBSO yönteminin sözde kodu.....	56
Şekil 4.1	: OBSO algoritması için geliştirilen programın örnek ekran görüntüsü.....	58
Şekil 4.2	: 51 noktalı veri kümesinin K-Ortalamlar – 2 öbek sonuçları.....	63
Şekil 4.3	: 51 noktalı veri kümesinin BCO – 2 öbek sonuçları.....	64
Şekil 4.4	: 51 noktalı veri kümesinin GK – 2 öbek sonuçları.....	64
Şekil 4.5	: 51 noktalı veri kümesinin PSO – 2 öbek sonuçları.....	65
Şekil 4.6	: 51 noktalı veri kümesinin OBSO – 2 öbek sonuçları.....	65
Şekil 4.7	: 51 noktalı veri kümesinin K-Ortalamlar – 3 öbek sonuçları.....	66
Şekil 4.8	: 51 noktalı veri kümesinin BCO – 3 öbek sonuçları.....	66
Şekil 4.9	: 51 noktalı veri kümesinin GK – 3 öbek sonuçları.....	67
Şekil 4.10	: 51 noktalı veri kümesinin PSO – 3 öbek sonuçları.....	67
Şekil 4.11	: 51 noktalı veri kümesinin OBSO – 3 öbek sonuçları.....	68
Şekil 4.12	: 400 noktalı veri kümesinin K-Ortalamlar – 4 öbek sonuçları.....	70
Şekil 4.13	: 400 noktalı veri kümesinin BCO – 4 öbek sonuçları.....	70
Şekil 4.14	: 400 noktalı veri kümesinin GK – 4 öbek sonuçları.....	71
Şekil 4.15	: 400 noktalı veri kümesinin OBSO – 4 öbek sonuçları.....	71
Şekil 5.1	: Iris veri kümesinde Dunn indeksine göre başarımlar.....	78
Şekil 5.2	: Iris veri kümesinde $V$ indeksine göre başarımlar.....	78
Şekil 5.3	: 51 noktalı veri kümesinde Dunn indeksine göre başarımlar.....	79
Şekil 5.4	: 51 noktalı veri kümesinde $V$ indeksine göre başarımlar.....	79
Şekil 5.5	: 400 noktalı veri kümesinde başarımlar.....	80

## TABLO LİSTESİ

<b>Tablo 2.1</b>	: Standart SZO algoritmasının parametreleri .....	35
<b>Tablo 4.2</b>	: Iris veri kümesinde Dunn indeksine göre başarımlar .....	62
<b>Tablo 4.3</b>	: Iris veri kümesinde hatasına göre başarımlar .....	62
<b>Tablo 4.4</b>	: 51 noktalı veri kümesinde $V$ indeksine göre başarımlar .....	68
<b>Tablo 4.5</b>	: 51 noktalı veri kümesinde Dunn indeksine göre başarımlar .....	69
<b>Tablo 4.6</b>	: 400 noktalı veri kümesinde $V$ indeksine göre başarımlar .....	72
<b>Tablo 4.7</b>	: 400 noktalı veri kümesinde Dunn indeksine göre başarımlar .....	72
<b>Tablo 4.8</b>	: Tesis yeri seçimi problemi için 51 noktanın koordinat ve talep verisi	74
<b>Tablo 4.9</b>	: Hesaplanan maliyet parametresi değerleri .....	75
<b>Tablo 4.10</b>	: Tesis yeri problemi için sonuç öbek merkezi koordinatları .....	75
<b>Tablo 4.11</b>	: Tesis yeri problemi için sonuç öbek merkezi koordinatları .....	76



## SEMBOL LİSTESİ

<b>SZO</b>	: Sürü Zekası Optimizasyonu
<b>BCO</b>	: Bulanık C-Öbekleme
<b>GK</b>	: Gustafson-Kessel
<b>OBSO</b>	: Odak Bireyli Sürü Optimizasyonu
<b>KP</b>	: Konveks Programlama
<i>d</i>	: Boyut sayısı
<i>z<sub>i</sub></i>	: <i>i.</i> veri elemanının konum vektörü
<i>S<sub>Z</sub></i>	: Veri elemanları kümesi
<i>l</i>	: Veri elemanı sayısı
<i>c</i>	: Öbek sayısı, Odak birey sayısı
<i>m</i>	: Birey sayısı
<i>b<sub>i</sub></i>	: <i>i.</i> birey
<i>S<sub>S</sub></i>	: <i>i.</i> bireyin konum vektörü
<i>v<sub>i</sub></i>	: <i>i.</i> bireyin hız vektörü
<i>p<sub>i</sub></i>	: <i>i.</i> bireyin o ana kadar gösterdiği en iyi performansın konum vektörü
<i>g<sub>i</sub></i>	: <i>i.</i> bireyin komşuları içerisinde en iyi performansa sahip olanın en iyi performansının konum vektörü
<i>J</i>	: Amaç fonksiyonu
<i>S<sub>i</sub></i>	: <i>i.</i> öbeği oluşturan veri elemanları kümesi
<i>f<sub>i</sub></i>	: <i>i.</i> odak birey
<i>S<sub>F</sub></i>	: Odak bireyler kümesi

## ÖZET

### **SÜRÜ ZEKÂSI OPTİMİZASYON TEKNIĐİ VE TEDARİK ZİNCİRİ YÖNETİMİNDE BİR UYGULAMA**

Bu çalışmada, tedarik zinciri yönetiminde oldukça önemli bir paya sahip olan tesis yeri seçimi problemleri için sürü zekası optimizasyon algoritmasına dayalı yeni bir öbikleme analizi yöntemi sunulmuştur. Sürü zekâsı, zor problemlerin çözümünde başarı sağlayan sezgisel bir yöntemdir. Bu yöntemin temel yapı taşları balık ve kuş sürüleri, karınca kolonileri, termitler ve arılar gibi sosyal canlıların davranışlarından esinlenilerek geliştirilmiştir. Sürü zekası optimizasyonu böcek yada balık sürülerinin davranışlarını taklit eden bir algoritmadır. Sürüdeki davranışlar çok boyutlu bir uzayda hız ve konum olmak üzere iki karakteristiğı olan bireyler ile modellenir. Bu bireyler çok boyutlu uzayda hareket ederken keşfettikleri en iyi konumu hafızalarında tutarlar. Eriştikleri iyi konumları birbirleriyle iletişime geçerek paylaşır ve kendi konum ve hızlarını bu iyi konumlara göre ayarlarlar.

Sürü zekâsı yaklaşımının birey ve komşuluk unsurlarını ele alarak yeni bir komşuluk yapısı ve birden fazla odak bireyin tanımlanması ile geliştirilen yeni bir öbikleme algoritması ile tedarik zinciri yönetiminde tesis yeri seçimi problemine yeni bir çözüm yöntemi geliştirilmiştir. Odak birey tabanlı bu yeni öbikleme algoritmasının mevcut öbikleme algoritmaları ile çeşitli test problemleri üzerinde performansı karşılaştırılmıştır.

Ortaya konan sonuçlar geliştirilen yeni yöntemin başarılı bir öbikleme analizi yöntemi olduğunu ve tedarik zinciri yönetiminde tesis yeri seçimi problemlerinde de başarıyla uygulanabileceğini göstermiştir.

## **SUMMARY**

### **PARTICLE SWARM OPTIMIZATION TECHNIQUE AND AN APPLICATION IN SUPPLY CHAIN MANAGEMENT**

A new swarm intelligence based clustering analysis method for facility location – allocation problems in supply chain management is presented in this study. Swarm intelligence is a heuristic method which is successfully applied to hard optimization problems. Algorithm is inspired by behaviors of social beings like fish schools, bird flocking, ant colonies, termites and bees. Behaviors in the swarm are modeled by individuals who have two main characteristics, namely position and velocity, in the hyperspace. These individuals remember their best previous position and at the same time share knowledge of their positions among each other. At each iteration, these individuals adjust their position and velocity vector by considering their current position and velocity, their memory and mentioned shared knowledge.

This new solution approach alters behaviors of particles in the swarm and neighborhood structure so that multiple focal particles are defined which reflect cluster centers. Clustering based facility location-allocation problems are solved using this new clustering approach. Comparisons are made on test problems between proposed focal particle based algorithm and other well known clustering analysis algorithms.

Results indicate that the new clustering algorithm is performed well against other well-known clustering algorithms and successfully applied in facility location-allocation problems in supply chain management context.

## 2. GİRİŞ

Bu çalışmada parçacık sürü optimizasyonunun temel metaforlarına dayanan yeni bir öbekleme analizi yöntemi geliştirilmiştir ve geliştirilen yöntem tedarik zinciri yönetiminde çoklu tesis yeri seçimi problemlerinin çözümüne uygulanmıştır. Sürü zekası, çeşitli problemlerin çözümünde bir grup olarak hareket eden kuş, balık, termit gibi canlıların sosyal düzenlerini modelleyen yaklaşımlardan esinlenerek geliştirilmiş bir optimizasyon metodolojisidir. Sürü zekası kavramının temelinde yatan lidersiz hareket eden, basit ve etkileşim içerisindeki bireyler olgusu, bu metodoloji kullanılarak çeşitli optimizasyon yöntemlerinin doğmasını sağlamıştır. Parçacık sürü optimizasyonu yöntemi de bu temel prensipler üzerine kurulmuş bir yöntemdir. Parçacık sürü optimizasyonu ilk olarak bu tip basit canlıların karmaşık problemleri çözmekteki etkinliğini sosyal ve psikolojik açıdan inceleyen ve temel etkileşimleri matematiksel olarak ifade etmeye çalışan bir yöntem olarak ele alınmıştır. Ancak ortaya konan ilerleme neticesinde bu temel metaforların pek çok mühendislik ve temel doğa bilimleri alanlarında da başarıyla uygulanabileceği görülmüştür.

Tesis yeri seçimi problemi 1960'lardan beri incelenen bir optimizasyon problemi olmakla birlikte tesis yeri seçimi problemini öbekleme analizi yaklaşımı ile ele alarak çözümleyen sınırlı sayıda çalışma bulunmaktadır. Farklı öbekleme analizi tekniklerinin tesis yeri seçimi problemlerinde başarımının karşılaştırıldığı çalışma sayısı ise daha azdır. Tesis yeri seçimi problemini, tesislerin ve talep noktalarının coğrafi konumları ve maliyetlere yada çevresel faktörlere etki eden çeşitli özelliklerin de modele dahil edilebilmesi açısından öbekleme analizi yaklaşımı ile çözmek etkili bir yöntem olarak görülse de literatürdeki öbekleme analizi yöntemlerinin amaç fonksiyonlarının, tesis yeri seçimi probleminin amaç fonksiyonu ile farklı hedeflere sahip olabilmesinden dolayı öbekleme ile yer seçimi optimizasyonunun bağdaştırılması amacıyla çoğunlukla melez yöntemler geliştirilmiş ve bu yöntemler başarılı olmuştur.

## 2.1. AMAÇLAR

Bu çalışmada yapay zeka ve sezgisel algoritmaların karmaşık optimizasyon problemlerini çözme konusundaki üstünlüklerinden faydalanırken, öbikleme analizi yaklaşımının tesis yeri seçimi problemlerini sürekli uzayda tanımlayabilme ve coğrafi konumlara uygun çözümlenebilme özelliklerinden faydalanılarak, takip eden amaçlara ulaşmaya çalışılmıştır.

- Öbikleme analizinin, tedarik zinciri yönetiminde tesis yeri seçimi problemlerine uygulanması için bir yöntem geliştirmek.
- Parçacık sürü optimizasyonu yaklaşımını kullanarak yeni bir öbikleme algoritması geliştirmek.
- Geliştirilen öbikleme analizi algoritmasının başarımını literatürdeki önemli algoritmalar ile karşılaştırarak test etmek.
- Geliştirilen öbikleme analizi algoritmasının tesis yeri seçimi problemlerine uygulamak ve literatürdeki benzer yöntemlerle performansını karşılaştırmak.

## 2.2. KATKILAR

Bu tezin literatüre temel katkıları takip eden maddelerde açıklanmıştır.

- Parçacık sürü optimizasyonu yönteminde birden fazla odak bireyin tanımlandığı yeni bir komşuluk yapısı sunulmuş;
- Bu yeni komşuluk yapısının yeni bir öbikleme analizi yöntemi olarak uygulaması geliştirilmiş;
- Yeni komşuluk yapısına sahip algoritmanın uygulanan probleme göre amaç fonksiyonu esnekliğine sahip olması sağlanmış;
- Tesis yeri seçimi problemlerinde öbikleme analizi kullanılarak bir optimizasyon modeli tanımlanmış;
- Tanımlanan problem geliştirilen öbikleme analizi yöntemi ile çözümlenerek sonuçlar irdelenmiştir.

### 2.3. TEZİN ANA HATLARI

Geliştirilen öbikleme analizi yöntemi, parçacık sürü optimizasyonu yönteminin temel işleyişini, bir veri kümesinin istenen sayıda öbeğe ayrılması için kullanılacak olan birden fazla odak bireyli yeni bir komşuluk yapısı ve odak bireyler etrafında öbek oluşumunu sağlayacak bir algoritma ile geliştirerek yeni bir yöntem olarak ortaya çıkmıştır. Bu yöntemin öncelikle bir öbikleme analizi olarak başarısı, çeşitli test problemleri üzerinde, literatürdeki K-Ortalamlar, Bulanık C-Ortalamlar ve Gustafson-Kessel algoritmaları ile karşılaştırılarak incelenmiş ardından öbikleme analizinin tedarik zincirinde yer seçimi problemlerinde güçlü bir çözüm aracı olarak kullanılabilmesi gösterilmiştir. Bu amaçla, geliştirilen yöntem kapasite kısıtsız, gerçek bir çoklu tesis yeri seçimi problemi üzerinde uygulanarak, literatürdeki diğer benzer yöntemlerle karşılaştırmalı analizi yapılmıştır.

Çalışmanın birinci bölümünde tedarik zinciri yönetimi kavramı anlatılmış ve tedarik zincirinde tesis yeri seçimi problemi tanımlanmıştır. Bu bölümde özel olarak tesis yeri seçimi problemlerinde ele alınmış olan öbikleme analizi yöntemleri incelenmiştir.

İkinci bölümde sürü zekası metodolojisi açıklanmış ve sürü zekası metodolojisinin temel prensiplerinden yola çıkarak geliştirilen parçacık sürü optimizasyonu yöntemi ele alınmıştır. Bu aşamada parçacık sürü optimizasyonu yönteminin algoritması incelenmiş, kullanım alanları ve literatürde bu algoritmanın başarımını arttırmak için yapılan çalışmalar değerlendirilmiştir. Ayrıca bu bölümde parçacık sürü optimizasyonu yönteminin tesis yeri seçimi problemlerine uygulandığı çalışmalar incelenmiştir.

Üçüncü bölümde ise bu çalışmada önerilen odak bireyli sürü optimizasyonu yöntemi yeni bir öbikleme analizi yöntemi olarak tanıtılarak, yöntemin temel parametreleri, algoritması ve işleyişi açıklanmıştır. Bu yeni yöntemin başarımını test problemlerinde karşılaştırmalı olarak ele alınmış ve bir tesis yeri seçimi problemine uygulanarak sonuçlar incelenmiştir.

### **3. GENEL KISIMLAR**

#### **3.1. TEDARİK ZİNCİRİ YÖNTEİMİ**

##### **3.1.1. Tedarik Zinciri Kavramı**

Tedarik zinciri, tüketicilerin elinde ürün ya da hizmet formunda değer üreten, farklı süreç ve aktiviteleri kapsayan organizasyonlar ağıdır (Christopher, 1998). APICS sözlüğünde tedarik zinciri şöyle tanımlanmaktadır: “1. Tedarikçi ile firma arasındaki, ilk hammaddeler ile tamamlanmış ürünlerin son tüketimi arasında yer alan süreçler; 2. Ürün ve hizmetlerin müşterilere sunulması için değer zincirini harekete geçiren firma içi ve dışı fonksiyonlar”.

Tedarik zinciri konseyi şu tanımlı kullanmaktadır: “Lojistik uzmanları tarafından sık olarak kullanılan tedarik zinciri terimi, tedarikçinin-tedarikçisi aşamasından, müşterinin-müşterisi aşamasına kadar, nihai ürünün üretimi ve teslimi için gereken tüm çabaları kapsamaktadır. Dört temel süreç -planlama, kaynak, üretim, teslim- bu çabaları geniş ölçüde tanımlamaktadır; bunlar arz ve talep yönetimi, hammadde ve parça tedarik kaynakları, üretim ve montaj, depolama ve envanter dağıtım, sipariş girişi ve sipariş yönetimi, tüm kanalda dağıtım ve müşteriye teslim aşamalarını içermektedir.”.

Quinn(2000) ise tedarik zincirini şöyle tanımlamaktadır: “Hammadde evresinden başlayarak son kullanıcıya kadar malların hareketi ile ilgili olan aktivitelerin tümü. Bunlar, kaynak ve tedarik, üretim planlama, sipariş süreci, envanter yönetimi, nakliyat, depolama ve müşteri hizmetlerini içermektedir. Daha da önemlisi, bu aynı zamanda tüm aktivitelerin gözlenmesi için gereken bilgi sistemlerini de kapsamaktadır.”.

Genel anlamda bir işletmenin amacı; ürettiği bir ürün ya da hizmeti tüketicilere sunmak ve hizmet verdiği alanda gelir sağlamaktır. Bu alanda rakiplerin artan sayısı ve rekabet koşulları karşısında ayakta kalmak amacı ile rekabet üstünlüğünün sağlanabilmesi için işletmelerin yüksek verimlilik, yüksek kalite ve düşük maliyet ile çalışmalarını gerekmektedir. Bir işletmenin kurulumundan, müşterisine sunduğu ürün ya da hizmetin

müşterinin kullanımına verdiği haline kadar geçen tüm süreçler bu esas gözetilerek yönetilmelidir.

### **3.1.2. Tedarik Zinciri Yönetimi Kavramı**

Tedarik zinciri yönetimi ürün, bilgi ve hizmet akışının, başlangıç noktasından son noktaya ulaşıncaya değin tedarik zinciri içerisindeki hareketliliğinin etkin ve verimli bir şekilde planlanması, depolanması ve taşınması hizmetidir.

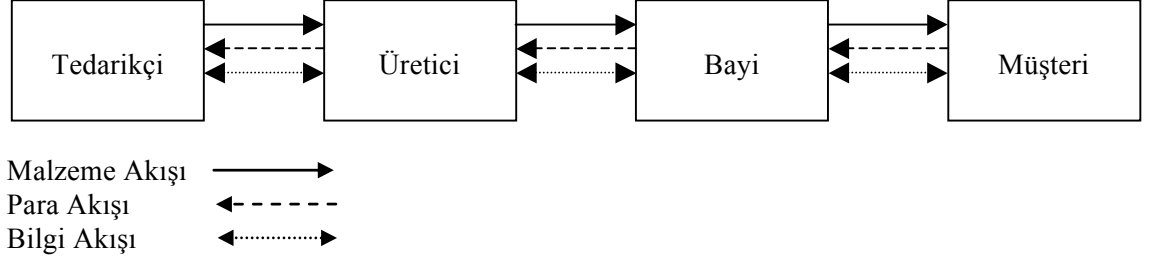
Tedarik zinciri yönetimi tüketiciye ürünleri doğru miktarda, doğru yerde ve doğru zamanda temin edebilmeyi sağlayan, ürün ya da hizmetin üretilmesi için gerekli olan birimlerin etkin olarak entegre edilmesi ile tüm sistem boyunca verimlilik ve etkinlik sağlayarak müşteri hizmet gereklerini yerine getirebilmektir. Tedarik zinciri yönetimi için farklı kaynaklarda hemen aşağıda verilmiş tanımlar da mevcuttur.

Tedarik zinciri yönetimi hammaddelerin kaynaktan elde edilmesinden son dağıtımla nihai kullanıcıya ulaştırılmasına kadar ürünün hareketinin her basamağının planlanması ve yönetilmesidir (Ross, 1998).

Tedarik zinciri yönetimi, işletme içinde ve tedarik zincirinde yer alan ticari ortakların işletme fonksiyonlarında, üretken yeteneklerini ve kaynaklarını birleştirmeye çalışan, değer yaratmak için yaratıcı çözümler geliştirmeye odaklanmış ve ürünler, hizmetler ve bilginin eş zamanlı akışını başarmış rekabetçi tedarik zincirine dönüşmelerini sağlamak amacıyla sürekli gelişen bir yönetim felsefesidir (Ross, 1998).

Entegre tedarik zincirinde bilgi ve malzemenin kesintisiz akışı esastır. Şekil 2.1 de tedarik zinciri boyunca malzemenin sağa doğru, paranın ise sola doğru aktığı gösterilmiştir. Bilgi ise tedarik zinciri boyunca her iki yönde de akar.





Şekil 2.1: Tedarik zinciri boyunca bilgi, malzeme ve para akışı

### 3.1.3. Tedarik Zinciri Yönetimi Bileşenleri

Tedarik zinciri yönetimi firmanın faaliyetlerinde stratejik seviyeden başlayıp, taktiksel seviyeye ve operasyonel seviyeye kadar geniş bir kısmı kapsar. Stratejik düzeyde uzun süreli etkileri olan kararlar verilir. Bu, fabrika ve depoların sayısı, yeri ve kapasiteleri ve materyallerin lojistik ağı boyunca akışının tasarımını kapsar. Tüm taktiksel ve operasyonel faaliyetlerin üst çatısını oluşturan bu aşamada verilecek kararlar tedarik zincirinde ortaya çıkan maliyetlerin, hizmet düzeyinin ve genel olarak firma karlılığının limitlerini belirleyen en önemli unsurlardan biridir. Taktiksel düzey, sadece bir veya birkaç yılda bir güncellenen kararları kapsar. Bunlar taşıma stratejileri, satın alma ve üretim kararları, stok politikaları ve müşterilerin hangi sıklıkla ziyaret edileceği kararlarını da içerir. Operasyonel düzey, kamyon yükleme, rota tayini, çizelgeleme gibi günlük kararları kapsar.

### 3.2. TEDARİK ZİNCİRİNİN MODELLENMESİ

Tedarik zinciri, son kullanıcıya yönelik ürün ya da hizmet biçiminde değer üreten, farklı süreç ve aktivitelerin bir araya geldiği bir organizasyonlar ağıdır (Christopher, 1998). Tedarik zinciri yönetimi, müşteri hizmet düzeyini artırırken maliyetleri düşürerek müşteri için değer yaratmak amacıyla tedarik zincirinin bütünleşmiş ve süreç odaklı tasarımını, yönetimi ve kontrolü olarak tanımlanabilir.

Tedarik zinciri stratejik, taktik ve operasyonel perspektiflerden ele alınabilecek yer seçimi, ağ yapısının kurulması, talep tahmini, satın alma, üretim, dağıtım, stok, taşıma ve müşteri hizmeti gibi birbirinden ayrı alanları kapsar. Tedarik zinciri yönetimi kavramı pek çok şekilde tanım bulmuştur. İlk olarak 1980'li yılların başlarında satın alma, üretim, dağıtım ve satışlar gibi farklı işletme fonksiyonları bağlamında yeni bir

bütünleşmiş lojistik yönetimi yaklaşımı olarak tanımlanmıştır (Houlihan, 1984). Daha sonra, bu bütünleşme yaklaşımı firma sınırlarının ötesine de taşınmış ve müşteri ile tedarikçiyi de kapsar hale gelmiştir. Son zamanlarda tedarik zinciri yönetimi kavramı çok daha geniş bir içeriğe ulaşarak, lojistikle ilgili olmayan bazı tedarik zinciri süreçlerini de bünyesine dâhil edilmiştir. Global Supply Chain Forum' a göre (Lambert ve diğ., 1998) tedarik zinciri, son kullanıcıdan asıl tedarikçilere kadar ürün, hizmet ve bilgi sağlayarak müşterilere ve hissedarlara değer kazandıran anahtar işletme süreçlerinin entegrasyonudur.

Çok disiplinli yönlerinin mevcudiyeti ve farklı tanımlamalarla kullanılması tedarik zinciri yönetimini karmaşıklarıştırmakta; öte yandan sınırlarının belli olmaması tedarik zinciri yönetimi üzerindeki akademik çalışmalar ile uygulamalar arasında bir boşluk yaratmaktadır. Tedarik zinciri yönetimi problemlerini ele alan çok sayıda model geliştirilmiş olmasına rağmen bunların çoğu henüz uygulama olanağı bulamamıştır.

Tedarik zinciri için literatürde yer alan modeller iki farklı boyutta ele alınabilir. Bunlardan birincisi problemin türü ve diğeri de modelleme yaklaşımıdır. Tedarik zinciri problemleri iki ana sınıfta gruplanır; tedarik zincirinin yapısı (biçimi) ve tedarik zincirinin koordinasyonu. Modelleme yaklaşımları açısından ise kavramsal, analitik, yapay zekâ ve simülasyon modellerinden bahsedilebilir (Giannoccaro, 2001).

### **3.2.1. Tedarik Zinciri Problemleri**

Yapısal yada koordinasyon problemleri olarak iki farklı tipe ayrılarak incelenebilecek olan tedarik zinciri problemlerinin karşılaşılan uygulama alanlarını gruplamak gerektiğinde üç ana uygulama alanı belirlenmiştir; Müşteri yönetimi, üretim yönetimi ve ürün geliştirme. Müşteri yönetimi hedef pazarın tanımlanması ve hedef müşteri kitlesine yönelik programların uygulanması ile ilgili aktiviteleri ele alır. Üretim yönetimi satın alma, talep tahmin, siparişlerin karşılanması ve lojistik gibi farklı süreçleri içerir. Ürün geliştirme, tedarik zinciri yönetimi bağlamında, tedarik zinciri için tasarım ve yerleştirme için tasarım gibi stratejileri kapsar (Giannoccaro, 2001). Bu geniş tedarik zinciri yönetimi problemleri tedarik zincirinin yapısı (biçimi) ve tedarik zincirinin koordinasyonu başlıklarıyla ayrıştırılıp incelenebilir (Şekil 2.2).



Şekil 2.2: Tedarik Zinciri Problemlerinin Sınıflandırılması

Yapı, temel olarak, tedarik zinciri ağının tasarımı bağlamında stratejik düzeydeki problemleri kapsar. Tedarik zinciri ağı, kendi içinde tedarik, üretim ve dağıtım ağlarından oluşur. Bu tür problemlerle ilgili kararlar al ya da üret problemi, satın alma stratejisi, tedarik sözleşmeleri ve tedarikçi seçimi süreci olarak tanımlanabilir. Tedarik zinciri yönetimi perspektifi doğrultusunda firmalar yalnızca kendi tedarikçilerini değil onların tedarikçilerini de ele almalıdırlar. Bunun yanı sıra firmalar, altyapı tedarikçilerinin de önemini farkına varmalıdırlar. Altyapı tedarik zinciri de dikkatlice tasarlanmalı ve yönetilmelidir.

Üretim-Dağıtım ağının tasarımı genel olarak yer seçimi ve eşleştirme problemlerinden oluşur. Bunlar, çeşitli tesislerin (fabrika ya da depo gibi) yeri kararlaştırılırken, aynı zamanda tesisler arasındaki akışların da talep odaklarına hizmet edecek şekilde paylaşılması problemleridir. Ele alınması gereken diğer konular dışarıdan edinim kararları, dağıtım kanalının tipi ve örneğin geri dönüşüm ağı ihtiyacı gibi çevresel

faktörlerdir. Sonuç olarak, bir tedarik zinciri yapı probleminin çözümü tedarik zinciri üzerindeki tesisleri ve bunlarla ilişkili bağlantıları belirlemektir.

Koordinasyon problemi ise taktik ve operasyonel bazda tedarik zinciri ağının yönetimini ele alır. Literatürde farklı koordinasyon türleri tanımlanmıştır.

Tedarik zinciri koordinasyon problemi, birkaç farklı karar verici tarafından verilen operasyonel kararların entegrasyonunu içerdiğinden oldukça karmaşık bir olgudur. Bu kararlar tek bir fonksiyon dâhilinde olabileceği gibi farklı fonksiyonları ya da birden fazla organizasyonu içerebilir. Bunun da ötesinde, mevcut bir probleme uygun tedarik zinciri modelinin seçiminde, koordinasyonun fonksiyonlar arası, organizasyonlar arası ya da fonksiyon içi olup olmayacağı kararından kaynaklanan belirgin önemin dışında, karar verme sürecinin merkezi ya da yerel olması da anahtar bir konudur. Merkezi karar verme süreci, tedarik zincirinde tüm tedarik zincirinden karar vermeye yetecek bilgileri alıp, işleyebilen ve bu kararları uygulatma gücüne sahip tek bir karar vericinin bulunduğu bir yapıdır.

Karar verme süreci merkezi değil ise tedarik zinciri üzerinde genellikle zincirin bir kısmındaki bilgi doğrultusunda karar veren çeşitli sayıda karar verici yer alır. Bu karar vericilerin birbirleriyle çakışma ihtimali olan farklı hedefleri (yerel hedefler) olması doğal olacaktır. Ardından yerel karar vericilerin kanal boyunca koordinasyonu için koordinasyon mekanizmaları gerekecektir. Aslında bu tip mekanizmalar riskin ve kazancın tüm tedarik zinciri birimleri arasında paylaşımını sağlayarak, tedarik zinciri birimlerinin kararlarının birbirleriyle uyuşması yönünde olumlu etki yaratırlar.

### **3.3. TEDARİK ZİNCİRİ YÖNETİMİ MODELLERİ**

#### **3.3.1. Kavramsal Modeller**

##### *Yapı Problemleri*

Kavramsal tedarik zinciri modellerinin büyük çoğunluğu tedarik zincirinin biçimi problemini ele alır. Tedarik ağı tasarımı açısından pek çok model al ya da üret kararı, dışarıdan edinim durumunda tedarikçi seçimi ve değerlendirme ile ulaştırma stratejilerinin belirlenmesi süreçlerini inceler.

### *Koordinasyon*

Kavramsal modellerde tedarik zinciri koordinasyonu yaklaşımlarının asıl hedefi tedarik zincirinin yönetiminde süreç odaklı bir yaklaşımın sağlanmasıdır. Bu tip modeller genellikle teorik çerçevede kalıp, tedarik zinciri kontrolünün merkezi olduğunu varsaymaktadır. Entegrasyonu tamamıyla sağlamış bir tedarik zinciri için fonksiyonel entegrasyon, iç entegrasyon ve dış entegrasyon aşamalarının başarılması gerekmektedir. Fonksiyonel entegrasyon satın alma, malzeme kontrol, üretim, satışlar ve dağıtımın üç bağımsız başlık altında, malzeme yönetimi, üretim yönetimi ve dağıtım, entegrasyonunu ele alır. İç entegrasyon bu üç alanda taktik entegrasyonu sağlamayı amaçlarken, dış entegrasyon bu entegre yöntemin kapsamının tedarikçi ve müşteriyi de içine alacak şekilde firma dışına genişlemesini sağlar (Giannoccaro, 2001).

### **3.3.2. Analitik Modeller**

Analitik modeller lineer programlama, dinamik programlama, tamsayı programlama, çok amaçlı programlama, markov karar süreci ve analitik hiyerarşik süreç gibi farklı yönelem araştırması tekniklerini temel alır.

### *Yapı*

Tedarik zinciri yapısı bağlamında analitik modeller temel olarak tedarikçi seçimi süreci ve üretim-dağıtım ağının tasarımını ele alırlar. Tedarikçi seçimi ve değerlendirmesi modelleri lineer ağırlıklandırma, lineer programlama, çok amaçlı programlama, AHP, Bulanık AHP, Elektre, Topsis ve ANP teknikleri ile çözülmeye çalışılır. Üretim dağıtım ağının tasarımı kapsamındaki modeller çoğunlukla tamsayı programlamanın adaptasyonunu kullanırlar.

### *Koordinasyon*

Daha önce de bahsedildiği gibi koordinasyon problemleri uygulanan karar verme sürecine bağlı olarak merkezi ya da merkezi olmayan şekilde ikiye ayrılır.

### *Merkezi Karar Verme*

Pazarlama-Üretim koordinasyonu, stok ya da üretim planlama ile fiyatlandırma kararlarının birlikte ele alınmasının yanında müşteriye sevkiyatın üretim planı ile de eş zamanlılığını içerir.

Üretim-Dağıtım koordinasyonu üretim ve dağıtım problemlerinin birlikte optimizasyonunu ele alır. Optimal üretim ve sevkiyat parti büyüklükleri için çeşitli

modeller geliştirilmiştir. Stok-Dağıtım koordinasyonu üretim-dağıtım sistemlerinde stok yönetimi problemini ele alır. Hedef, tüm sistem için en uygun olan stok politikasını belirlemektir.

#### *Yerel Karar Verme*

Yerel karar verme tarzıyla işleyen tedarik zincirinde miktar iskontoları, miktar esnekliği, yedekleme anlaşmaları iade politikaları gibi konular üzerinde durulmaktadır.

### **3.3.3. Yapay Zekâ Tabanlı ve Sezgisel Modeller**

Çeşitli tedarik zinciri yönetimi problemlerini çözmek üzere yapay zekâ teknikleri geliştirilmiştir.

#### *Yapı*

Tedarik zincirinin yapısıyla ilgili problemlerde, yapay zekâ, esas olarak tedarik stratejisi, tedarikçi seçimi ve üretim-dağıtım ağının tasarımı başlıkları altında kullanılmaktadır. Kararların daha hızlı, daha isabetli ve daha tutarlı olması için uzman sistemlere dayalı karar destek sistemlerinin geliştirilmesi gerektiği 1990'lı yılların başında dile getirilmiştir. Albino ve diğ.(1998) bulanık mantık temeline dayalı sözlü ifadeleri de işleyebilen bir tedarikçi değerlendirme modeli geliştirmiştir.

Tesis yeri seçimi problemi bu başlık altında en sık ele alınan konulardan biridir. Tesis yeri seçimi için bulanık mantık, yapay sinir ağları, genetik algoritmalar, tavlama benzetimi, diferansiyel hesaplama, genetik programlama, tabu arama, sürü zekası gibi teknikler kullanılarak çeşitli modeller geliştirilmiştir.

#### *Koordinasyon*

Merkezi karar verme hipotezi altında yapay zekâ tekniklerinin uyarlandığı koordinasyon problemleri üzerinde çalışmalar yapılmıştır. Belirsizliğin modellenmesi için bulanık küme yaklaşımının kullanıldığı tedarik zinciri stok yönetimi çalışmaları yapılmıştır. Bulanık küme teorisi, yönetimin deneyim ve kararlarını da süreçlere dâhil ederek belirsizliğin modellenmesinde son derece kullanışlı ve turtalı sonuçlar vermektedir. Bu çalışmalarda bulanık değişkenler genellikle maliyetler ve talep için kullanılmıştır (Küçükdeniz, 2004).

Tedarik zincirlerinin taktik ve operasyonel seviyede yönetimi için yerel yapıda çalışabilecek yazılımlar geliştirilmiştir. Fox (2000) tedarik zincirini her biri

sorumlulukları uyarınca birbirleriyle etkileşime girebilen, bir akıllı yazılımlar zinciri olarak ele almıştır. Akıllı yazılımlar otonom, hedefe odaklı, gerektiğinde diğer yazılımlarla iletişim ve işbirliği kurabilen süreçler olarak değerlendirilmektedirler. Bunlar problemleri yapay zekâ ve yöneylem araştırması metotları ile çözmeye çalışmaktadırlar.

Bu tip bir sistemde tedarik zinciri farklı alt sistemler halinde ele alınmaktadır. Bunlar satın alma, lojistik, taşıma, çizelgeleme, kaynak kullanımı alt sistemleridir.

### **3.3.4. Simülasyon Modelleri**

Diğer uygulamalarında da olduğu gibi simülasyon tedarik zinciri yönetimi problemlerinin daha detaylı ve daha gerçekçi bir şekilde ele alınması avantajına sahiptir. Bu sebeple tedarik zinciri yönetiminde simülasyon modelleri belli başlı problem çözümlerini test etmek amacıyla kullanılmaktadır. Alternatif olarak, basitleştirilmiş genel bir sistemin davranışının simülasyonu yolu ile meselenin ana noktalarının anlaşılması sağlanabilir. Bu ikinci durumda, sonuçların hem yapı hem de koordinasyon parametreleri üzerinde etkileri gözlemlenebilir.

Tedarik zinciri performansını arttırmak için malzeme akışını düzgünleştirmek üzerine odaklanmış olan endüstriyel dinamikler teorisi (Towill, 1996) bu başlık altında incelenmesi gereken bir yaklaşımdır. 60'lı yılların başlarında tedarik zincirinde yukarı doğru çıkıldıkça belirsiz talebin yarattığı kamçı etkisi tanımlanmıştır (Christopher, 2005). Bu etkinin tedarik zincirinin yapısına ve işlerin bitirilmesi için gerekli zaman ve karar vermedeki zayıflıklara bağlı olduğu görülmüştür. Bu etki tedarik zinciri koordinasyonuna müdahale etmek yolu ile azaltılabilir. Bu müdahaleler mevcut sipariş politikalarının ince ayarı, beklemlerin azaltılması, yerel karar verme kurallarının değiştirilmesi ve tedarik zinciri boyunca bilginin akışının daha iyi sağlanması şeklinde gerçekleştirilebilir.

### 3.4. TESİS YERİ SEÇİMİ PROBLEMİ

Bu çalışmada, bölüm 2.2.1’ de bahsedilen tedarik zinciri yönetimi problemlerinden tedarik zinciri yönetiminde tesis yeri seçimi problemi ele alınmıştır. Bu problem, bölüm 2.3’ de sınıflandırılan modelleme yaklaşımlarından, yapay zeka ve sezgisel optimizasyon tabanlı bir çözüm yöntemi geliştirilerek incelenmiştir. Tesis yeri seçimi problemi NP-zor optimizasyon problemlerinin en çok incelenmiş olanlarından (Güner, 2008). Tanım olarak müşteriler yada talep merkezlerinin konumları ve bunlara bağlı olan talep büyüklüklerinin bilindiği bir durumda tesis yerlerinin sayılarının, konumlarının, kapasitelerinin belirlenmesi ve talebin bu tesislere atanması problemidir. Bu tip problemlerde sıklıkla rastlanan amaç işletme maliyetlerinin en küçüğe indirgenmesidir. Dolayısıyla amaç fonksiyonunun, kurulum ve ulaştırma maliyetlerini kapsadığı bir optimizasyon problemi söz konusudur.

Lojistik ağı üzerinde sabit tesis yerlerinin seçimi problemi, lojistik sisteminin tamamının performansını etkileyen bir karar verme problemidir. Yerleşim problemleri tekli ve çoklu tesis yeri seçimi olmak üzere iki ana başlık altında sınıflandırılır. Tekli tesis yeri seçimi (ttys) ve çoklu tesis yeri seçimi (çtys) problemleri kapasite kısıtlı veya sonsuz kapasiteli kaynak durumuna göre incelenir. Çoklu tesis yer seçim problemi, tekli yer seçimi probleminden farklı olarak yalnızca arz/depo tesis yerlerinin belirlenmesi ile sınırlandırılmayarak hem arz/depo tesis yerlerinin belirlenmesi hem de talebin (müşteri) bu arz/depo tesislerine eşleşmelerinin incelendiği problemler olarak da ele alınmıştır.

Literatürde çtys problemleri için alternatif tesis yerleri arasında seçim yapmaya dayalı atama yöntemleri ya da alternatiflerin söz konusu olmadığı problemler için koordinat sisteminde yer belirlemeye dayalı gerçek sayı optimizasyonu yöntemleri bulunmaktadır. Coğrafi düzlemde tesis yeri seçimi problemlerini çözmek amacıyla geliştirilen yöntemlere, ağırlık merkezi, p-medyan yöntemi gibi klasik yöntemler ve öbekleme analizi yöntemi yada öbekleme analizinin diğer klasik yöntemlerle birlikte kullanıldığı melez yöntemler olarak karşılaşılmaktadır. Çoklu tesis yer seçimi probleminin geleneksel optimizasyon yöntemleri ile çözümü Sule (2001), ReVelle ve Eiselt (2005) tarafından incelenmiştir.



Bazı temel varsayımlar üzerinde ele alındığında, Sule (2001), tesis yeri seçimi problemlerini beş temel kategoride sınıflandırmıştır;

1.  $p$ -medyan problemleri
2.  $p$ -merkez problemleri
3. Kapasite kısıtsız tesis yeri seçimi problemleri
4. Kapasite kısıtlı tesis yeri seçimi problemleri
5. Karesel atama problemleri

$p$ -medyan problemi,  $p$  adet tesisin toplam maliyeti en küçük kılacak şekilde  $p$  adet yere yerleştirilmesi problemidir. Maliyet zaman, para, sevkiyat sayısı, toplam uzaklık veya diğer bir ölçüt olarak belirlenebilir. bu tip problemlerde amaç en küçük toplam maliyete ulaşmak olduğundan Minisum problemi yada Weber problemi olarak da adlandırılır (Sule, 2001).

$p$ -merkez problemleri, amaç fonksiyonu olarak, herhangi bir tesis ile bu tesisin eşleştiği talep noktaları arasındaki en büyük uzaklığın en küçük değerini bulacak şekilde formülize edildiği modellerdir. İtfaiye istasyonları, cankurtaran hizmetleri, polis merkezleri gibi acil durum ve felaket modellerinde sıklıkla kullanılan bir yöntemdir.

Kapasite kısıtsız tesis yeri seçimi problemi, amaç fonksiyonunun bir minisum fonksiyon olduğu ancak maliyet denklemine tesislerin konumlarına göre değişen, sabit bir kurulum maliyetinin de eklendiği problemlerdir. Tesis sayısı önceden belirli olmamakla birlikte, toplam maliyeti en küçük kılacak şekilde belirlenmektedir. Kapasite kısıtı olmadığına bu tür modellerde bir talep noktasındaki talebi birden fazla tesisle eşleştirmek hiç bir zaman uygun çözüm olmamaktadır.

Kapasite kısıtlı tesis yeri seçimi problemi, kapasite kısıtsız problemle temel olarak aynı olmakla birlikte her bir tesisin önceden belirlenmiş sabit bir kapasite kısıtı altında ele alındığı modellerdir. Bu modellerde bir talep noktasına birden fazla tesisin farklı oranlarda hizmet sağladığı durumlar ortaya çıkabilmektedir.

Karesel atama problemleri  $n$  adet tesisin,  $n$  adet konuma, toplam maliyeti en küçük kılacak şekilde eş zamanlı olarak yerleştirilmesi problemidir. Bu tip problemlerde

yerleştirilmesi gereken  $n$  tesis için  $n!$  kadar olası çözüm kombinasyonu ortaya çıkmaktadır.

#### 3.4.1. Kapasite Kısıtsız Çoklu Tesis Yeri Seçimi Problemleri

Kapasite kısıtı olmayan tesis yeri seçimi problemlerinde, modeldeki her tesisin sınırsız miktarda ürün yada malzeme üretebileceği ya da dağıtımını yapabileceği kabul edilmektedir. 1960'lardan itibaren bu tip problemlerin çözümü için doğrusal programlama, dinamik programlama,  $p$ -medyan,  $p$ -merkez, minisum ve lagranj gevşetmesine dayalı yöntemler gibi pek çok matematiksel model geliştirilmiştir (Galvao, 2004).

Matematiksel modellemeye dayalı klasik çözüm yöntemleri çözüm uzayındaki en iyi çözüme ulaşma kapasitesine sahip olmalarına rağmen ele alınan problemin büyüklüğü ve kısıtların miktarı ile orantılı olarak hesaplama ihtiyaçları artmaktadır. Sezgisel yöntemler ise her zaman en iyi çözüme ulaşmayı garanti etmemekle birlikte problemin tüm kısıtlarının modele dahil edilebilmesi ve daha düşük hesaplama ihtiyaçlarından dolayı tercih edilmektedirler.

Çtys aynı tesis tarafından hizmet alan müşterilerin öbekler olarak dikkate alındığı durumlara özel bir öbekleme problemi olarak tanımlanır (Levin ve Ben-Israel, 2004). Levin ve Ben-Israel (2004) yaptıkları çalışmada çtys probleminin çözümü için geleneksel öbekleme analizini kullanmışlardır. Bu yöntem atık toplama, satış ekiplerinin yerleşimi, dağıtım sistemlerinde depo yerlerinin belirlenmesi, iletişim ağlarında aktarım merkezlerinin konumlandırılması, denizde petrol sonda yerlerinin belirlenmesi, pazar araştırmalarında farklı müşteri gruplarının öbeklenmesi, salgın hastalıklarla mücadele, belediye hizmetleri, acil durum ve felaket yardım modellemesi gibi alanlarda ele alınmıştır (Negresios ve Palhano, 2006).

Bulanık öbekleme analizi ise noktaların öbeklerle kısmi olarak başka bir deyişle üyelik dereceleri ile ilişkilendirilmelerine izin verir. Bu bir noktanın aynı zamanda birden fazla öbeğe ait olması demektir (Döring ve diğ. 2006). Sheu (2002) otoyol olaylarını bulanık öbekleme teorilerini esas alarak tanımlayan modeli geliştirmiştir. Bu melez bulanık hiyerarşik öbekleme analizi modelinin farklı versiyonları Hu ve Sheu (2003),

Sheu (2006) ve Sheu (2007) 'nun çalışmalarında sırasıyla bir mağaza zincirinin talep tepkili lojistik dağıtım işlemleri, tele-pazarlama firmasının yine talep tepkili şehir içi lojistik dağıtım işlemleri ve otomobil, bilgisayar aksesuarları üreten bir üretici firmanın lojistik dağıtım işlemlerinde kantitatif ve çoğunlukla kalitatif müşteri özelliklerini analiz etmek ve bu müşterileri gruplamak için kullanılmıştır. Yukarıdaki üç çalışmada kullanılan bu yöntem çok özellikli müşteri sipariş verilerinin sayısal olsalar bile mutlaka iki değerli ifadelerle dönüştürülmesini gerektirmektedir.

Lin ve diğerleri (1998) bir tedarik zincirinde siparişlerin karşılanması sürecinin performansını iyileştirmek için çoklu ajan tabanlı bir PSO modeli geliştirmişlerdir. Bu modelde tedarik zincirindeki çeşitli unsurlar sürü elemanları olarak tanımlanmış ve bir tedarik zincirinde sipariş karşılama sürecinin çoklu ajan bazlı bir yaklaşımla nasıl modellenebileceği gösterilmiştir.

Bu çalışmada tedarik zinciri yönetiminin önemli problemlerinden biri olan çoklu tesis yeri seçimi problemlerinin çözümü için parçacık sürü optimizasyonuna dayalı bir öbikleme yöntemi geliştirilmiştir.

### **3.4.2. Çoklu Tesis Yeri Seçimi Problemlerinde Öbikleme Analizi**

Öbikleme çok geniş uygulama alanına sahip olan temel veri analizi yöntemlerinden biridir. Öbikleme analizinin temel amacı verinin içerisinde gömülü olan belirgin grupları tanımlayarak ortaya çıkarmaktır. Öbikleme analizi, bir küme içerisindeki elemanları, öbek içerisindeki elemanlar arasındaki benzerlikler en çok, öbekler arasındaki benzerlikler en az olacak şekilde belirli sayıda öbeğe tahsis etmektir. Özellikle veri madenciliği, yapay zeka, iz tanıma, mühendislik, sosyal bilimler, iktisat, biyomedikal gibi alanlarda öbikleme analizi sıklıkla kullanılmaktadır (Kogan ve diğ., 2006). Tesis yeri seçimi problemlerinde öbikleme analizi çoklu tesis yeri seçimi probleminin tesis yerlerinin tespiti ve hangi talep noktalarının hangi tesis ile eşleşeceğinin belirlenmesi amacıyla kullanılmıştır.

Genel olarak tesis yeri seçimi ile ilgili öbikleme analizi modellerinde  $C$  talep noktaları kümesi ve  $F$  tesis yerleri kümesi modele girdi olarak verilerek, talep noktaları ile tesisler arasındaki ulaştırma maliyetlerini en aza indirgeyecek  $F'$  alt kümesinin tespiti

hedeflenir. Bu amaçla kullanılan klasik öbikleme algoritmaları k-medyan ve k-merkez algoritmalarıdır (Zalik, 2006). K-ortalamlar algoritması ise amaç fonksiyonu k-medyan algoritması ile aynı olmakla birlikte  $F$  kümesinin elemanlarının modele girdi olarak verilmesini gerektirmez (Chen, 2004).

K-Ortalamlar öbikleme algoritmasının bazı dezavantajlarından olumsuz etkilenmemek için çeşitli öbikleme algoritmaları geliştirilmiştir. Bunların en iyilerinden biri Bezdek' in (1981) bulanık C-Ortalamlar öbikleme algoritmasıdır. Bulanık C-Ortalamlar algoritması K-ortalamlar algoritması ile benzerlikler gösteren bir öbikleme yöntemidir. K-ortalamlar algoritmasından farkı ise veri noktalarının birden fazla öbeğe üyelik derecesi ile ait olmasını sağlayan bulanık sayıları kullanarak hesaplama yapması ve sonuçlar elde etmesidir.

Bulanık C-Ortalamlar yöntemi ve bu yöntemle yapılan eklentiler finans, tıp, manyetik görüntüleme, stok yönetimi, hücresel imalat, akustik, iz tanıma, haberleşme ağları gibi alanlarda uygulanmıştır (Bharitkar ve Kyriakakis, 2001; Cheng ve diğ., 1998; Eschrich, 2003; Esnaf, 1998; Lu ve diğ., 2006; Nascimento ve diğ., 2005; Özçakar ve Esnaf, 1998; Türkşen ve Esnaf, 1997; Türkşen ve Esnaf, 1998).

Esnaf ve Küçükdeniz (2006, 2009), Bezdek' in Bulanık C-Ortalamlar (BCO) (Bezdek ve Dunn, 1975) ve Gustafson ve Kessel' in GK (Gustafson ve Kessel, 1979), sırasıyla biri dairesel diğeri eliptik olan, bulanık öbikleme algoritmaları ile önce sonsuz kapasiteli çtys problemini ttys problemine dönüştürmüş daha sonra ttys problemini ağırlık merkezi yöntemi ile çözerek dairesel ve eliptik öbikleme ile ağırlık merkezini birleştiren melez yöntemlerin sonuçlarını karşılaştırmalı olarak vermiştir. Geliştirilen modelde öncelikle müşterileri ifade eden talep noktaları coğrafi konumlarına göre BCO dairesel bulanık öbikleme tekniği ile gruplanırlar ve öbek merkezleri tesislerin başlangıç yerleri olarak kabul edilir. İkinci aşamada, önceki aşamada tesislerin başlangıç yerleri olarak kabul edilen öbek merkezleri ağırlık merkezi yöntemi kullanılarak ulaştırma maliyetini en az kılacak şekilde kesinleştirilir.

Esnaf ve Küçükdeniz (2007) benzer bir yaklaşımı kapasite kısıtlı çoklu tesis yeri seçimi problemine uygulamışlardır. Bu çalışmada klasik öbikleme fonksiyonundan farklı

olarak, önceden belirlenmiş tesis yerleri için bulanık öbeklemenin üyelik derecesi belirleme fonksiyonundan yararlanılmış ve kapasite kısıtı altında tesis - müşteri eşleşmesini sağlayan bir algoritma geliştirilmiştir. Geliştirilen algoritma taleplerin sürekli olarak en uygun tesise atanmasını sağlayan bir yazılımda kullanılmıştır.

Esnaflar ve Küçükdeniz (2008), 2007 yılında geliştirdikleri algoritmanın geliştirilmiş halini kapasite kısıtı altında çoklu tesis yeri seçimi problemlerinin genel çözümü olarak ele almışlardır. Bu çalışmada, tesis – müşteri eşleşmeleri ve tesis öbeklerinin belirlenmesi için BCO yönteminin her bir talep noktasının öbekler için ayrı ayrı tespit ettiği üyelik derecelerinden faydalanılmıştır. Müşteri talepleri ilgili öbekteki tesisin kapasitesi yeterli olduğu sürece, talebin bu öbek için belirlenen üyelik derecesi kadar miktarı, öbekteki tesise atanarak kapasite kısıtı probleme dâhil edilmiştir.

### **3.4.3. Talep Noktalarının Coğrafi Öbeklemesi İçin Bulanık C-Ortalamlar Algoritması**

Bir veri kümesinin en uygun bulanık öbeklerini bulma problemi Dunn (1974) tarafından üyelik fonksiyonlarına bağlı bir amaç fonksiyonunun en küçüğe indirgenmesi olarak tanımlanmıştır. Bezdek (1981), bulanık C-ortalamlar olarak adlandırılan geliştirilmiş öbekleme algoritmaları ailesini geliştirmiştir. Literatürde yer alan çok sayıdaki öbekleme algoritmaları içerisinde bulanık C-ortalamlar algoritması en popüler olanlarından biridir (Emami ve diğ., 1996). Bulanık C-Ortalamlar öbekleme algoritması, C-Ortalamlar fonksiyonu olarak adlandırılan bir amaç fonksiyonunun en küçüklenmesi temeline dayanmaktadır (Bezdek ve Dunn, 1975). BCO algoritması standart Öklid uzaklık formunu kullanır. Bu hiperküresel öbekleri ortaya çıkarır. Dolayısıyla, sadece aynı şekil ve oryantasyona sahip öbekleri tespit edebilir. Genel olarak bu öbekler dairesel şekillidir.

BCO öbekleme algoritması için üç girdi parametresi gereklidir.

- Öbek sayısı
- Bulanıklaştırma ağırlık faktörü( $p$ )
- Maksimum tolerans ( $\epsilon$ )

Zalik (2006)' de tesis yeri seçimi problemlerinde bulanık öbekleme yaklaşımını ele almış ve algoritma parametreleri için uygun değerlerin  $p=1,7$  ve  $\varepsilon=10^{-6}$  olduğunu göstermiştir.

BCO algoritmasında amaç fonksiyonu formül 2.1' deki gibi tanımlanır.

$$J_p(U, v) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik})^p \|a^k - v_i\|^2 \quad (2.1)$$

Formül 2.1' deki parametreler şekil 2.3' de açıklanmıştır.

$c$	: Algoritma ile elde edilmek istenen önceden belirlenmiş öbek sayısı
$U = \{[u_{ik}], 1 \leq i \leq c, 1 \leq k \leq n\}$	: Bölümlenme matrisi
$v = \{[v_i], 1 \leq i \leq c\} \mid v_i \in \mathfrak{R}^m$	: Öbek prototiplerinin vektörleri
$a^k (1 \leq k \leq n)$	: Öbeklenecek veri kümesi
$p \in (1, \infty)$	: Üyelik derecelerinin hesaplanmasında ağırlık faktörü

Şekil 2.3: BCO algoritmasının amaç fonksiyonu parametreleri

Optimizasyon problemi formül 2.1' deki kısıt altında  $J_p(U, v)$ 'nin en aza indirgenmesi olarak tanımlanır.

$$\sum_{i=1}^c u_{ik} = 1 \quad \forall k \quad (2.2)$$

Optimizasyon probleminin çözümü için kullanılan üyelik fonksiyonu ve öbeklerin son prototip vektörlerinin hesaplanması formül 2.3 ve 2.4' da gösterilmiştir (Bezdek, 1981).

$$v_i = \sum_{k=1}^n (u_{ik})^p a_k / \sum_{k=1}^n (u_{ik})^p, \quad 1 \leq i \leq c \quad (2.3)$$

$$u_{ik} = 1 / \sum_{j=1}^c \left( \frac{\|a^k - v_i\|}{\|a^k - v_j\|} \right)^{2/(p-1)}, \quad 1 \leq i \leq c, \quad 1 \leq k \leq n \quad (2.4)$$

Formül 2.3 ve 2.4' ü kullanarak ardışık yinelemelerle öbek merkezleri ve üyelik dereceleri tespit şekil 2.4'de algoritmanın sözde kodu ile açıklanmıştır.

Adım 1: Öbek sayısı  $c'$  yi belirle,  $p'$  yi belirle,  $v_1, v_2, \dots, v_c$  için başlangıç değerlerini belirle

Adım 2: Formül 2.4 ile  $u_{ik}$  üyelik derecelerini hesapla ( $1 \leq i \leq c$ ,  $1 \leq k \leq n$ ).

Adım 3: Formül 2.3'ü kullanarak öbek merkezlerini  $v_1^{yeni}, v_2^{yeni}, \dots, v_c^{yeni}$  güncelle

Adım 4: Eğer  $\max_i \left\{ \|v_i - v_i^{new}\|_{err} \right\} < \epsilon$  ise dur, değilse adım 2' ye git.

Şekil 2.4: BCO algoritmasının işleyişi

Şekil 2.4' deki  $\epsilon$  sonlandırma toleransını ifade etmektedir.

Bu çalışmada test problemlerinin BCO algoritması ile performanslarının hesaplanması için Balasko ve diğ. (2005) Matlab için geliştirdikleri bir program kullanılmıştır. Parametre değerleri olarak ise Zalik (2006)' in belirttiği değerler kullanılmıştır.

#### 3.4.4. Talep Noktalarının Coğrafi Öbeklemesi İçin Gustafson – Kessel Öbekleme Algoritması

Gustafson-Kessel, standart BCO algoritmasının farklı bir uzaklık hesaplama yöntemi kullanılarak geliştirilmiş versiyonudur. Bu sayede farklı geometrik şekillere sahip öbeklerin tespiti mümkün hale gelmiştir. Gustafson-Kessel algoritması (GK) Mahalanobis uzaklık normunu kullanır (Gustafson ve Kessel, 1979). GK algoritması BCO algoritması ile aynı girdilere sahiptir. Algoritma benzer şekilde ardışık yinelemeler neticesinden önceden belirlenmiş bir sonlandırma kriteri gerçekleşinceye kadar çalıştırılır (Babuska, 2002). Algoritmanın amaç fonksiyonu denklem 2.5'de verilmiştir.

$$J(X; U, V, \{A_i\}) = \sum_{i=1}^c \sum_{k=1}^N (\mu_{ik})^m D_{ikA_i}^2 \quad (2.5)$$

Burada  $U = [\mu_{ik}] \in [0,1]^{cxN}$   $X \in R^{nxN}$  verisinin bulanık bölüm matrisi,  $V = [v_1, v_2, \dots, v_c]$ ,  $v_i \in R^n$  öbek prototiplerinin (merkezlerinin) vektörü ve  $m \in [1, \infty)$  sonuç öbeklerinin bulanıklığını belirleyen parametredir. Uzaklık normu  $D_{ikA_i}$  bir veri setindeki farklı geometrik şekilleri dikkate alabilir ve hemen aşağıdaki gibi hesaplanır:

$$D_{ikA}^2 = \|x_k - v_i\|_{A_i}^2 = (x_k - v_i)^T A_i (x_k - v_i) \quad (2.6)$$

Her bir öbeğin ölçüsü 2.5 numaralı formülde optimizasyon değişkenlerinden biri olarak kullanılan yerel bir  $A_i$  norm indirgeme matrisi için tanımlanır. Bu uzaklık normunun, verilerin yerel topolojik yapısına uyum sağlamasına izin verir.

Bu öbekleme algoritmasında veri örnekleri küçük veya öbek içindeki veriler doğrusal olarak birbirleriyle ilişkili olduğunda kovaryans matrisi tekil hale gelebilmektedir. Bu problemin basit ve etkili bir biçimde çözülebilmesi için Babuska ve diğ. (2002) aşağıda detayları verilen GK algoritmasını geliştirmişlerdir:

Verilen  $X$  veri kümesi için,  $c, m, \varepsilon, p_i$  standart parametreleri koşul sayısı eşiği  $\beta$  ve ağırlık parametresi  $\gamma$  seçilir. Bölüm matrisi başlangıç değerleri belirlenir ve tüm veri kümesi için kovaryans matrisi,  $F_0$ , hesaplanır.

$l=1, 2, \dots$  için tekrarla:

1.Adım: Öbek merkezlerini hesapla:

$$V_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m x_k}{\sum_{k=1}^N (\mu_{i,k}^{(l-1)})^m}, 1 \leq i \leq c \quad (2.7)$$

2.Adım: Öbek kovaryans matrisini hesapla:

$$F_i^{(l)} = \frac{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m (x_k - v_i^{(l)})(x_k - v_i^{(l)})^T}{\sum_{k=1}^N (\mu_{ik}^{(l-1)})^m}, 1 \leq i \leq c \quad (2.8)$$



Ölçeklenmiş özdeş matrisi ekle:

$$F_i = (1 - \gamma)F_i + \gamma(F_0)^{1/n} I \quad (2.9)$$

$F_i$ 'den  $\lambda_{ij}$  ve  $\Phi_{ij}$  özdeğerlerini çıkar,

$\lambda_{i,\max} = \max_j \lambda_{ij}$  bulunur ve

$\lambda_{i,\max} = \lambda_{ij} / \beta, \forall j$  için  $\lambda_{i,\max} / \lambda_{ij} \geq \beta$  yapılır.

$F_i$  aşağıdaki formülle yeniden oluşturulur:

$$F_i = [\Phi_{i,1} \dots \Phi_{i,n}] \text{diag}(\lambda_{i,1} \dots \lambda_{i,n}) [\Phi_{i,1} \dots \Phi_{i,n}]^{-1} \quad (2.10)$$

3.Adım: Uzaklıkları hesapla:

$$D_{ikA_i}^2(x_k - v_i) = (x_k - v_i^{(l)})^T [(\rho_i \det(F_i))^{1/n} F_i^{-1}] (x_k - v_i^{(l)}) \quad (2.11)$$

4.Adım: Bölüm matrisini güncelle.

$1 \leq k \leq N$  için;

Eğer  $D_{ikA_i} > 0$   $1 \leq i \leq c$  için

$$\mu_{i,k}^{(l)} = \frac{1}{\sum_{j=1}^c (D_{ikA_i}(x_k, v_i) / D_{ik}(x_k, v_i))^{2/(m-1)}}, \quad (2.12)$$

Aksi halde,

Eğer  $D_{ikA_i} > 0$  ve  $\mu_{i,k}^{(l)} \in [0,1]$

$$\mu_{i,k}^{(l)} = 0$$

Aksi halde,  $\sum_{i=1}^K \mu_{i,k}^{(l)} = 1$

$\|U^{(l)} - U^{(l-1)}\| < \varepsilon$  gerçekleşene kadar devam et.

### 3.4.5. Tesis Yerlerinin Tespiti için Melez Yöntemler

Bu çalışmada sunulan Odak Bireyli Sürü Zekası Optimizasyonu yönteminin tesis yeri seçimi problemlerindeki başarımının değerlendirilmesi amacıyla, bulanık öbikleme, bulanık öbikleme ve ağırlık merkezi, bulanık öbikleme ve konveks programlama ile parçacık sürü optimizasyonu tabanlı çeşitli melez/melez olmayan yöntemlerle karşılaştırmalar yapılmıştır. Tesis yeri seçimi problemlerinde tek başına öbikleme analizi yaklaşımlarından daha iyi BCO tabanlı sonuçlar veren iki farklı yöntem ele alınmıştır;

1. BCO öbikleme algoritması ve ağırlık merkezi melez yöntemi
2. BCO öbikleme algoritması ve konveks programlama melez yöntemi

### 3.4.6. Bulanık Öbikleme - Ağırlık Merkezi Melez Yöntemi

Bu yöntem Esnaf ve Küçükdeniz (2006) tarafından önerilmiş bir yöntemdir. Tesislerin koordinat eksenindeki konumları ve tesislerin hizmet sağlayacağı öbekler 2.4.3 ve 2.4.4'de detayları verilen BCO ve GK bulanık öbikleme algoritmaları ile tespit edildikten sonra her öbek içerisindeki müşterilerin talep büyüklükleri göz önüne alınarak özel bir ağırlık merkezi yöntemi uygulanır. Ağırlık merkezi yönteminin amacı müşteri talep noktalarıyla tesisler arasındaki ve tesislerle tedarikçiler arasındaki ulaştırma maliyetlerini en aza indirmektir. Şekil 2.5 amaç fonksiyonu ve yöntemin temel parametrelerini göstermektedir (Ballou, 1999).

<p>En Az <math>\rightarrow TM = \sum_i V_i R_i d_i</math></p> <p><math>TM</math> : Toplam Maliyet</p> <p><math>i</math> : talep (müşteri) veya hammadde (tedarikçi) indeksi</p> <p><math>V_i</math> : <math>i</math> noktasından ulaştırılacak miktar</p> <p><math>d_i</math> : kurulacak tesisin <math>i</math>. talep noktasına uzaklığı</p> <p><math>R_i</math> : Ulaştırma maliyeti</p>
---

Şekil 2.5: Ağırlık merkezi yönteminin amaç fonksiyonu ve parametreleri

Bu yöntemin ilk aşamasında her bir öbeğin ağırlık merkezi formül 2.13, 2.14 ve 2.15 ile hesaplanır:

$$\bar{X} = \frac{\sum_i V_i X_i}{\sum_i V_i} \quad (2.13)$$

$$\bar{Y} = \frac{\sum_i V_i Y_i}{\sum_i V_i} \quad (2.14)$$

$$d_i = \sqrt{(X_i - \bar{X})^2 + (Y_i - \bar{Y})^2} \quad (2.15)$$

Ardından bu değerler kullanılarak  $d_i$  yeniden hesaplanır. Bulunan  $d_i$  değeri formül 2.16 ve 2.17' da yerine konarak yeni  $\bar{X}$  ve  $\bar{Y}$  koordinatları bulunur.

$$\bar{X} = \frac{\sum_i V_i R_i X_i / d_i}{\sum_i V_i R_i / d_i} \quad (2.16)$$

$$\bar{Y} = \frac{\sum_i V_i R_i Y_i / d_i}{\sum_i V_i R_i / d_i} \quad (2.17)$$

Bulunan son koordinatlara göre  $d_i$  yeniden hesaplanır ve  $\bar{X}$  ve  $\bar{Y}$  koordinatlarında çok az değişiklik oluncaya kadar yinelemelere devam edilir.

### 3.4.7. Bulanık Öbekleme - Konveks Programlama Melez Yöntemi

Detayları 2.4.3'de verilen BCO algoritması ile BCO-Ağırlık Merkezi yönteminde olduğu gibi öbekler belirlenir ve problem çok tesisli yer seçimi probleminden tek tesisli yer seçimi problemine dönüştürülür. Daha sonra Can ve diğerlerinin (2006) çalışmasında kullandığı, talep noktalarının koordinatları ve talep büyüklüklerini girdi olarak kabul ederek ulaştırma maliyetlerini en aza indirgeyecek koordinatları hesaplayan bir konveks programlama modeli BCO sonuçlarına uygulanmıştır. Modelin amaç fonksiyonu formül 2.18'de gösterilmiştir.

$$[\text{En Az}] Z = \sum_{i=1}^n w_i \left[ (m_x - x_i)^2 + (m_y - y_i)^2 \right]^{\frac{1}{2}} \quad (2.18)$$

$$-m_x \leq 0, \quad -m_y \leq 0$$

Formül 2.18' deki denklemde gösterilen parametrelerin açıklamaları şekil 2.6' de yer almaktadır.

$n$	: Talep noktası sayısı
$w$	: Talep büyüklüğü
$x, y$	: Talep noktalarının koordinatları
$m_x, m_y$	: En uygun noktanın koordinatları

Şekil 2.6: Konveks programlama yönteminin amaç fonksiyonu parametreleri

### 3.4.8. Ulaştırma Maliyetinin Hesaplanması

Toplam ulaştırma maliyeti denklem 2.19' daki gibi hesaplanır.

$$TUM = \sum_{k=1}^n \sum_{i=1}^c (w_k d_{ik} c_{ik}) \quad (2.19)$$

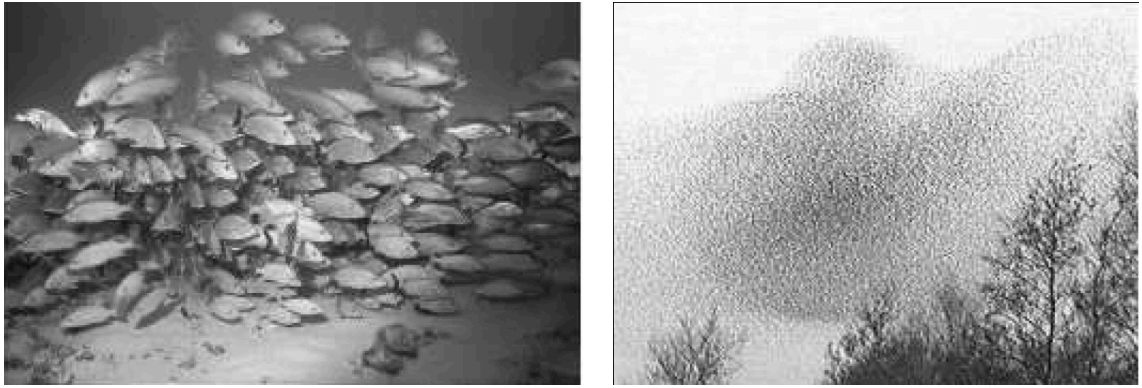
Toplam ulaştırma maliyeti formülündeki parametreleri şekil 2.7' de açıklanmıştır.

$TUM$	: Toplam ulaştırma maliyeti
$c_{ik}$	: $i$ . tesisten $k$ . talep noktasına birim ulaştırma maliyeti
$w_k$	: $k$ . talep noktasının talep büyüklüğü
$d_{ik}$	: $i$ . tesis ile $k$ . talep noktası arasındaki mesafe

Şekil 2.7: Toplam ulaştırma maliyeti fonksiyonunun parametreleri

### 3.5. SÜRÜ ZEKASI

Koloni veya sürü halinde yaşayan sosyal böcekler ve diğer canlılar herhangi bir otorite olmadan çalışırlar. Bu tip yapılarda takım çalışması bireyler arasındaki çeşitli etkileşimler sonucu kendi kendine ortaya çıkan bir unsurdur (Bonebeau, 2001). Bu etkileşimler ne kadar basit olursa olsun bir araya geldiklerinde büyük problemler için etkin çözümler sunabilirler. Karıncaların olası sonsuz sayıda rota içerisinde yiyecek kaynağına giden en kısa yolu bulabilmeleri buna bir örnektir. Bir grup halinde bulunan sosyal canlıların ortaya koydukları kolektif davranışa sürü zekası denir.



Şekil 2.8: Balık ve kuş sürüleri

Sürü Zekası algoritmaları balık, kuş, arı gibi sürü halinde hareket eden canlıların sosyal düzenlerini modelleyen yaklaşımlardan esinlenilerek geliştirilmiş algoritmalarlardır. Bu algoritmaların iki temel özelliği herhangi bir liderin olmaması ve sürü içindeki etkileşimlerin yerel olmasıdır. Bu iki özellik nedeniyle, geliştirilen modellerde komşuların hızına uyma, komşularla çarpışmaktan kaçınma ve komşulardan çok uzaklaşmaktan kaçınma gibi davranışlar ortaya çıkmaktadır.

Sürü zekâsı doğadaki ya da yapay olarak oluşturulmuş ortamlardaki kendi kendine organize olabilen süreçlerin incelendiği görece yeni bir disiplindir. Hayvan davranışları ve etimoloji ile ilgilenen araştırmacılar tarafından yıllar içerisinde ortaya konmuş olan modeller tarafından esinlenerek zor ve karmaşık hesaplamalı problemlerin çözümü hedeflenmiştir.

Sürü zekâsı çatısı altında incelenen konu başlıklarından biri karınca kolonisi optimizasyon tekniğidir. Burada odak noktası kesikli optimizasyon problemleridir. Karınca kolonisi optimizasyonu gezgin satıcı problemi, karesel atama problemi, çizelgeleme, araç rotalama gibi pek çok kesikli optimizasyon probleminde başarı ile uygulanmıştır. Bir diğer yaklaşım ise parçacık sürü optimizasyonu tekniğidir. Burada odak noktası sürekli optimizasyon problemleridir. Bir diğer araştırma alanı ise sürü robotiğidir. Burada birlikte çalışan çok sayıda otonom robotun kontrolü için sürü zekâsı prensipleri kullanılmaktadır.

Bu çalışmada parçacık sürü optimizasyonu tekniğini ile çoklu tesis yeri seçimi problemlerine çözüm aranmış ve parçacık sürü optimizasyonu tekniğine dayalı yeni bir yöntem geliştirilmiştir.

### 3.6. PARÇACIK SÜRÜ OPTİMİZASYONU YÖNTEMİ

#### 3.6.1. Çıkış Noktası

Bilim adamları kuş ve balık sürülerindeki organizmaların hareketlerini açıklayan çeşitli bilgisayar simülasyonları geliştirmişlerdir. Reynolds kuş sürülerinin uçuşlarındaki simetriyi incelerken (Reynolds, 1987), Heppner ve Grenander bir sürüdeki çok sayıda kuşun nasıl senkronize bir biçimde uçup, aniden topluca yön değiştirebildiklerini, aniden dağılıp yine aniden tekrar grup haline gelebildiklerini incelemiştir (Heppner ve Grenander, 1990). Söz konusu davranışların modellenmesi sırasında modellerin ağırlıklı olarak bireyler arasındaki mesafenin ayarlanmasına dayandığı görülmüştür. Buna göre sürü davranışındaki senkronizasyonu sağlayan etken, kuşların, birbirleri arasında optimum mesafe bırakmak için gösterdikleri çabanın bir fonksiyonudur.

Parçacık sürü optimizasyonu algoritması 1995 yılında James Kennedy ve Russell Eberhart tarafından geliştirilmiştir. Yukarıda bahsedilen sosyal metaforları temel alan bu algoritma ile arama ajanları arasında bilgi paylaşımı (sosyal etkileşim) sağlanarak optimizasyon hedeflenmiştir. Algoritmanın temel fikri sürü dâhilindeki en iyi ile her bir bireyin kendi en iyisinin sürekli olarak izlenmesidir. Kennedy ve Eberhart bu yöntemi ilk olarak sosyal ve bilişsel davranışların incelenmesi amacıyla sunmuşlardır. Ancak yöntemin geniş olarak uygulandığı alan mühendislik ve temel doğa bilimleri olmuştur. Bu yaklaşım biyoloji, matematik ve sosyal bilimler gibi farklı bilim dallarındaki çeşitli konseptlerin bir araya getirilmesi ile ortaya konulmuş olan görece yeni bir optimizasyon metodolojisidir. Sürü zekası kavramının altında yatan temel fikir bir grup halindeki basit bireylerin karmaşık ve zeki davranışlar sergileyeceğidir. Bu amaçla doğada gözlemlenen grup halinde davranış yapısından esinlenilerek bir takım stratejiler geliştirilmiş ve bunlar mühendislik problemlerinin çözümü için teorik matematiksel temeller haline getirilmiştir.

Sürü birbirleriyle doğrudan ya da dolaylı olarak iletişim içerisinde olan ve kolektif olarak bir problemi çözme amacı taşıyan belirli sayıda bireylerden oluşmuş bir yığın olarak tanımlanır. Sürüdeki bireylerden her birinin amacı çevrelerine uyum sağlayacak şekilde hareket etmektir. Bu metaforun matematiksel olarak ifade edilmesi her bireyin bir uyum (amaç) fonksiyonu doğrultusunda arama uzayında hareket etmesi olarak

tanımlanmıştır. Ortaya konan yöntem ayrıca bireylerin sosyal varlıklar olduğunu belirtmektedir. Bu da bireylerin diğer bireylerle iletişime açık oldukları anlamına gelir. Sürüdeki her bir bireye açık olan iki tür bilgi vardır. Birincisi kendi deneyimleridir. İkincisi ise belirli sayıdaki diğer bireyin (bu sayı sürünün tamamı da olabilir yalnızca diğer bir birey de olabilir) en iyi deneyimleridir. Bu iki bilgi bireyin öğrenme davranışına etki ederek sürü içerisinde bir tür bilişsel akışı temsil etmektedir.

Bireyler kendi geçmiş başarılarından ve iletişim içerisinde olduğu diğer bireylerin başarılarından etkilenme eğilimindedir. Bu iletişimin şekli Kennedy ve Eberhart tarafından sosyometrik prensipler olarak adlandırılmıştır. Bireyler kendi aralarında çeşitli yollarla iletişim kurabilirler. En basit yöntem ikili iletişimdir. Bu durumda her birey kendine en yakın diğer iki birey ile iletişim halindedir. Bunun yerine kendine en yakın  $k$  adet birey de seçilebilir. Eğer bir bireyin iletişim için seçtiği birey sayısı sürüdeki toplam birey sayısının altında ise buna *lbest* prensibi denir (Eberhart, Kennedy, 1995). Eğer birey tüm sürü ile haberleşiyor ise buna *gbest* prensibi denir.

Sürüde her birey kendi davranışını yönetecek minimum basitlikte tanımlanmıştır. Temel olarak bu bireysel davranışın özellikleri şunlardır:

- Davranışlar olumlu geri beslemeden güç alır
- Davranışlar olumsuz geri besleme ile dengelenir
- Davranışlar rastgele salınımlar gösterebilir
- Davranışlar hayata geçirilme aşamasında diğer bireylerin davranışlarından bilgi alır

Sürü zekasının başlıca avantajları şunlardır;

- Ölçeklenebilirlik: Sürünün büyüklüğü problemin arama uzayının büyüklüğüne göre ayarlanabilir.
- Merkezi kontrol mekanizmasının olmaması bireysel hataların göz ardı edilmesini kolaylaştırır. Bazı bireylerin başarısız olması tüm sürüyü doğrudan etkilemez.
- Sistemdeki değişiklikler ajanları etkiler. Bu sayede ajanlar değişebilir, yok olabilir yada yeniden sisteme girebilirler



- Sürü zekası tabanlı yöntemlerde kullanıcının modele etki edeceği parametre sayısı azdır.

### 3.6.2. Standart Algoritma

Parçacık sürü optimizasyonu algoritması, yinelemeler boyunca hareket eden bir takım bireyden oluşan bir sürünün davranışını ele almaktadır. Parçacık sürü optimizasyonu yöntemi (PSO) bir arama uzayında belirli bir fonksiyonun yerel/küresel en uygun değerlerini arar. PSO ile incelenebilen modeller;

- Matematiksel olarak formülize edilebilen problemler
- Çok adımlı algoritmalar
- Hesaplanabilir çıktıları olan gerçek bir süreç yada sürecin simülasyonu olabilir.

Bir problemin PSO yöntemi ile ele alınabilmesi için temel kriter, seçilen sistemin herhangi bir anında değerinin hesaplanabilir olmasıdır (Clerc, 2006).

Temel PSO algoritması, bir sürü olarak tanımlanan birden fazla bireyin arama uzayında verilen bir fonksiyonun en iyi noktalarını yinelemeli bir süreç ile aramasını hedefler. Her birey aday bir çözümü temsil eder. Bireyin arama uzayındaki durumu bireyin konum vektörü ile ifade edilir. Bu algortmada her yineleme bireylerin buldukları konumu değerlendirmeleri, kendilerinin daha önce bulunmuş oldukları konumlar içerisinde en iyi olanı değerlendirmeleri ve tanımlanan bir komşuluk çerçevesinde diğer bireylerle etkileşim kurarak onların en iyilerini değerlendirmeleri işlemlerini kapsar. Bu süreç neticesinde her yinelemenin sonunda bireyin bir sonraki yinelemede gideceği yön hesaplanır ve birey arama uzayında hareket eder.

#### *Modelin Kurulum Aşamaları*

Birinci adımda arama uzayında bir sürü tanımlanmalıdır. Bu aşamada belirlenecek olan parametre sürü büyüklüğüdür. En basit yaklaşım sürünün sürekli sabit bir büyüklükte olmasıdır. Ayrıca elde edilen sonuçlara göre optimizasyon sürecinde sürü büyüklüğünün anlık olarak değiştirilebilmesini öngören daha karmaşık modeller de geliştirilmiştir (Tillett ve diğ., 2005).

Doğada gözlemlenen kolonilerden örneğin arı kolonilerinde birey sayısı 20000'e kadar çıksa da PSO yönteminde temel modeller için 20 ile 40 arası bir büyüklük yeterlidir (Clerc, 2006).

Başlangıç anında tüm parçacıklar aynı noktada bulunur. İlk zaman artırımında (birinci yineleme) bu parçacıklar rastgele olarak arama uzayına dağılır. Bu anda sistemde üretilmiş bir bilgi bulunmadığından rastgele dağılım mantıklı bir yöntem olarak kabul edilmiştir (Kennedy ve Eberhart, 1995). Sürecin basitleştirilmesi için bu rastgele dağılmış durum PSO algoritmasının başlangıç durumu olarak kabul edilir.

Başlangıç anında rastgele olarak belirlenen bir diğer parametre ise bireylerin hızlarıdır. Hız vektörü bir pozisyona uygulanan ve sonuç olarak yeni bir pozisyon veren bir operatör görevini üstlenir. Algoritmanın her yinelemesi algoritma zamanında bir birim artırım anlamına geldiğinden ve sürüdeki bireyler her artırımda hız vektörlerinin belirttiği kadar mesafe kat ettiklerinden, bu vektöre birim zamanda alınan mesafenin karşılığı olarak hız vektörü adı verilmiştir (Kennedy ve Eberhart, 1995).  $x_{\min}$  ve  $x_{\max}$  sırasıyla arama uzayında  $x$  parametresinin alabileceği en küçük ve en büyük değerler olarak tanımlandığında bu hız vektörünün başlangıç büyüklüğü arama uzayının büyüklüğüne bağlı bir fonksiyon ile hesaplanır. Konum ve hız vektörlerinin başlangıç durumunda alacakları değerleri hesaplamak için formül 2.20 kullanılır.

$$\begin{aligned} x_d &= UNIF[x_{\min}, x_{\max}] \\ v_d &= UNIF\left[\frac{(x_{\min} - x_{\max})}{2}, \frac{(x_{\max} - x_{\min})}{2}\right] \end{aligned} \quad (2.20)$$

Bu formülde  $v$  hız vektörü,  $d$  ise vektörün boyutlarını ifade etmektedir. Bunun yanı sıra her birey için, o bireyin iletişim içerisinde olacağı bilgilendiricilerinin hangi bireyler olacağı tespit edilmelidir.

Bireylerin hangi sayıda diğer bireyi bilgilendireceğinin belirlenmesi için farklı yaklaşımlar ortaya konmuştur. Eğer bireylerin tüm diğer bireyleri bilgilendirmesi benimsenirse bilginin kullanımının arttığı ve bunun iyi bir yaklaşım olduğu düşünülebilir. Ancak aynı bilgiyi alan tüm bireylerin benzer doğrultularda hareket edeceği düşünüldüğünde, bu yaklaşım optimizasyon sürecinde küresel arama

performansını düşürecektir. Aksi şekilde her bir bireyin çok az sayıda diğer birey ile bilgi alış verişi içerisinde bulunduğu durumda ise bireylerin davranışlarında farklılık sağlanmış olacak ancak bilginin değerlendirilmesinin etkinliği düşecektir. Önemli olan bir birey iyi bir konuma ulaştığında tüm diğer bireylerin doğrudan ya da dolaylı olarak bundan haberdar olmasının sağlanmasıdır.

#### *Bilgilendirilecek Birey Sayısının Belirlenmesi*

Sürüdeki bilgi alış verişi yapısının oluşturulması için “her bir yinelemede önceden belirlenmiş  $K$  adet bilgi bağlantısının oluşturulduğu kabul edilirse herhangi bir  $C$  bireyinin, diğer bir  $A$  bireyinin belirli bir anda ürettiği bilgiyi kesin olarak alabilmesi için gerçekleştirilmesi gereken yineleme sayısı  $T$  ne olmalıdır” sorusuna cevap aranmalıdır (Clerc, 2006). Bu soru,  $A$  bireyi tarafından üretilen bilgiyi  $T$  yineleme sonunda tüm diğer bireylerin alması için gereken bağlantı sayısı  $K$  ne olmalıdır şeklinde de ifade edilebilir.

Sürüdeki birey sayısı  $N$  ise, ilk yinelemede  $A$  bireyi, kendi bilgisini  $N$  adet birey içerisinde  $K$  adedine yaymaktadır. Bu durumda herhangi bir  $C$  bireyinin  $A$  bireyi tarafından seçilmeme ihtimali formül 2.21’ de gösterilmiştir.

$$p = 1 - \frac{1}{N} \quad (2.21)$$

$K$  adet tercihten sonra hala  $C$ ’ nin seçilmemiş olma ihtimali ise formül 2.22’ deki gibidir.

$$p_k = \left(1 - \frac{1}{N}\right)^K \quad (2.22)$$

İkinci yineleme ile birlikte toplam  $K^2$  adet birey rastgele olarak seçilmiş olacaktır. Dolayısıyla  $t$ . yineleme neticesinde bir bireyin seçilmeme olasılığı formül 2.23’ deki gibidir.

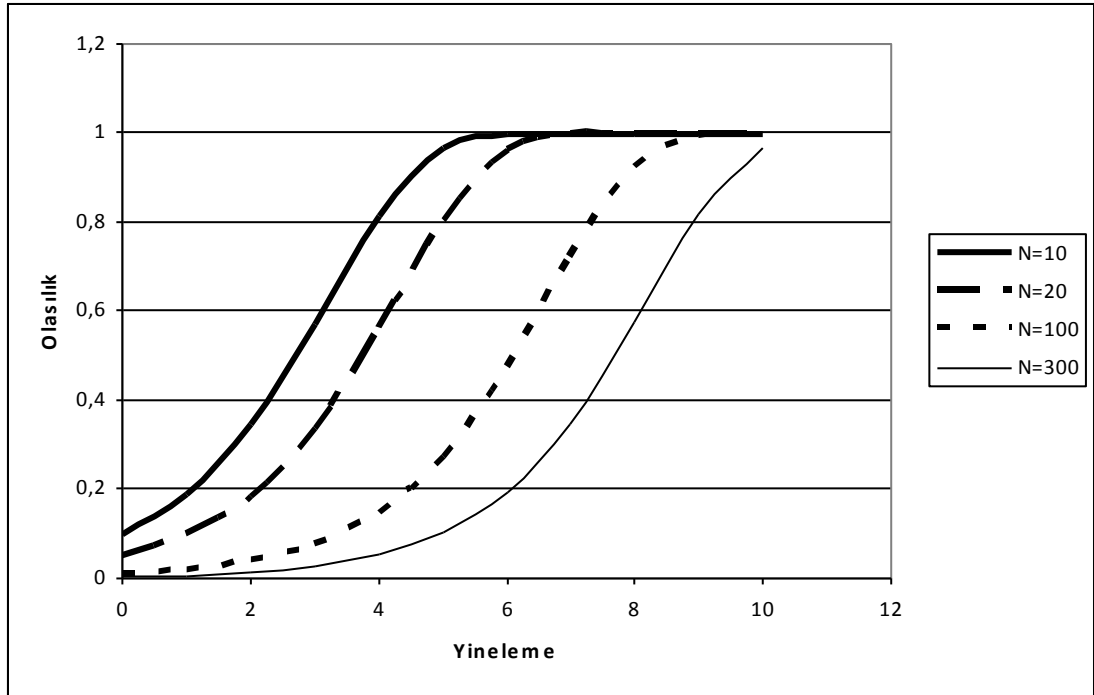
$$p_{k(t)} = \left(1 - \frac{1}{N}\right)^{K^t} \quad (2.23)$$

Sonuç olarak  $C$  bireyinin  $t$  adet yineleme neticesinde en azından bir kez bilgilendirilmiş olması olasılığı formül 2.24’ da gösterilen şekilde hesaplanır.

$$p_{(t)} = 1 - \left(1 - \frac{1}{N}\right)^{Kt} \quad (2.24)$$

Görüldüğü gibi bu olasılık  $t$ 'nin artması ile birlikte çok hızlı olarak artacaktır.  $K$ 'nin bilginin tüm bireylere yayılımı için çok büyük bir değer almasına gerek yoktur.

Şekil 2.9' da  $K=2$  için çeşitli sürü büyüklüklerinde  $t$ 'ye bağlı olarak herhangi bir bireyin bilgilendirilme olasılığı gösterilmiştir.



Şekil 2.9: Bilginin yayılımı.

### Temel Hareket Denklemi

$d$  boyutlu arama uzayında herhangi bir  $i$  bireyinin konumu  $x_i$  vektörü ile ifade edilir.

Bireyin o anki hızı ise  $v_i$  vektörü ile gösterilir. Bu iki vektör arama uzayının boyut sayısı  $d$  kadar elemandan oluşur. Her eleman, bireyin, ilgili boyuttaki konum ve hızını göstermektedir. örneğin tesis yeri seçimi probleminde tesislerin  $X$  ve  $Y$  koordinatları bulunmak isteniyorsa, hız ve konum vektörleri iki boyutlu vektörlerdir. Ya da klasik bir üretim probleminde 5 farklı ürünün çeşitli kısıtlar altında en uygun üretim miktarları hesaplanmak isteniyorsa, hız ve konum vektörleri 5'er boyutlu olacaktır. Dolayısıyla her

bir bireyin konum vektörü bu 5 ürünün üretim miktarlarını gösteren aday bir çözümü temsil edecektir.

Her yinelemede, bireylerin o anki konumları kullanılarak bir  $f$  fonksiyonunun değeri hesaplanır ve bireylerin konum ve hızları ayarlanır. Eğer bir birey,  $f$  fonksiyonunu değerlendirerek, daha önce bulduğu tüm değerlerden daha iyi değer veren bir konumda olduğunu görürse bu konumu  $p_i$  vektöründe saklar.  $p_i$  vektörü  $i$ . bireyin o ana kadar sahip olduğu değerler içerisinde en iyisidir. Bir en küçüğü arama probleminde, bu, en düşük  $f$  değeri anlamına gelir.  $p_i$  vektörünün güncellenmesi formül 2.25’ de verilen çerçevede gerçekleşir.

$$p_{id} = \begin{cases} p_{id} & \text{eğer } f(x_i) \geq f(p_i) \\ x_{id} & \text{eğer } f(x_i) < f(p_i) \end{cases} \quad (2.25)$$

$i$ . bireyin o anki konumu ( $x_i$ ) ile  $p_i$  (bireyin o ana kadar bulduğu en iyi konum) arasındaki fark bireyin o anki hızına stokastik olarak eklenerek bireyin yörüngesinin bu nokta civarında salınması sağlanır. Buna ek olarak, her bir birey için kendisi ve sürüdeki diğer bazı bireyleri içeren topolojik bir komşuluk ilişkisi tanımlanır.  $i$ . bireyin kendi tanımlanmış komşuluk yapısındaki en iyi konum  $g_i$  olarak ifade edilir ve bu değer ile bireyin o anki konumu arasındaki fark da bireyin hızına stokastik olarak eklenerek, hız vektörünün bir sonraki yineleme için değeri hesaplanır (Formül 2.26).

$$\begin{aligned} v_{id} &\leftarrow v_{id} + c_1(p_{id} - x_{id}) + c_2(g_{id} - x_{id}) \\ x_{id} &\leftarrow x_{id} + v_{id} \end{aligned} \quad (2.26)$$

Bireyin arama uzayındaki hareketine etki eden bu faktörler neticesinde bireyler yinelemeler süresince kendi en iyileri ve komşuluk kümesinin en iyisini gösteren iki değer etkisi altında kalarak arama yaparlar. Bireylerin arama uzayındaki yörüngeleri ve bu yörüngelere etki eden faktörler arasındaki ilişki Bergh ve Engelbrecht (2006)’ in çalışmasında ele alınmıştır. Bu çalışma neticesinde sürüdeki tüm bireylerin bir süre sonra durağan bir noktaya yakınsadığı ortaya çıkmıştır.

$c_1, c_2$  algoritmanın rastgele parametreleridir. Algoritma parametrelerinde kullanılan rastgele ağırlıklandırma arama uzayı içerisinde meydana gelen hareketin karakteristiğini

belirleyen temel etmendir.  $c_1$  ve  $c_2$  bireyin, kendi en iyi performansı ve bilgilendiricilerinin en iyi performansına uygulanan etki katsayılarıdır.  $c_1$  parametresi sürü zekası metodolojisinin çıkış noktası olan sosyal metafordan yola çıkılarak “bilişsel” parametre,  $c_2$  parametresi ise “sosyal” parametre olarak adlandırılır (Laskari ve diğ., 2002). Bu değerler, her yinelemede,  $[0, c_{\max}]$  aralığında düzgün dağılıma uygun olarak rastgele seçilirler. Dolayısıyla  $c_1$  ve  $c_2$  ifadeleri yerine denklemlerde  $UNIF(0, c_{\max})$  gösterimi de tercih edilebilir (Formül 2.27).

$$\begin{aligned} v_{id} &\leftarrow v_{id} + UNIF(0, c_{\max})(P_{id} - x_{id}) + UNIF(0, c_{\max})(g_{id} - x_{id}) \\ x_{id} &\leftarrow x_{id} + v_{id} \end{aligned} \quad (2.27)$$

Temel hareket denkleminin doğal bir neticesi olarak yüksek hızlarda bazı bireylerin  $[x_{\min}, x_{\max}]$  olarak belirlenen arama uzayının dışına çıkabileceği görülmektedir. Bu sorunu ortadan kaldırmak için herhangi bir yinelemede arama uzayının dışına çıkan parçacıkların konumları en yakın sınır değerine çekilir. Ayrıca bireyin bir sonraki yinelemede tekrar arama uzayının dışına çıkmasının engellenmesi için bireyin hız vektörü sıfırlanır. Bu düzenleme formül 2.28’de gösterilmiştir.

$$x_d \notin [x_{\min}, x_{\max}] \Rightarrow \left\{ \begin{array}{l} v_d \leftarrow 0 \\ x_d < x_{\min} \Rightarrow x_d \leftarrow x_{\min} \\ x_d > x_{\max} \Rightarrow x_d \leftarrow x_{\max} \end{array} \right\} \quad (2.28)$$

Sonuç olarak standart PSO algoritmasının parametreleri tablo 2.1’de özetlenmiştir.

Tablo 2.1: Standart PSO algoritmasının parametreleri

Parametre	Açıklama	Tavsiye Edilen Değer
$c_{\max}$	Rastlantısal parametre	1,47 (Trelea, 2003)
$N$	Sürü büyüklüğü	20 – 40 (Clerc, 2006)
$K$	Komşuluk sayısı	3 (Clerc, 2006)

### Sözde Kod

Algoritmanın sözde kodu şekil 2.10’ da gösterilmiştir. Bu algortmada sonlandırma kriteri olarak önceden belirlenmiş  $t_{\max}$  yinelemeye ulaşana kadar algoritma çalıştırılmaktadır.

```

Sürünün başlangıç durumunu oluştur: rasgele  $x_i, v_i$ 

Belirle:  $x_{\min}, x_{\max}, v_{\max}$ 

YAP: Sonlandırma kriteri gerçekleşmediği sürece ( $t \leq t_{\max}$ )

    {Her bir i. birey için
      {
        Eğer  $f(x_i) \leq f(p_i) \Rightarrow p_i = x_i$ 

         $g_i = MIN(x_{komşuluk})$ 

        Her bir d boyutu için
          {
            Hızı hesapla
            Konumu hesapla
            Eğer arama uzayının dışına çıkıldı ise
               $x_d \leftarrow [x_{\min}, x_{\max}]$ 
            }
          }
      }

    t=t+1
  }

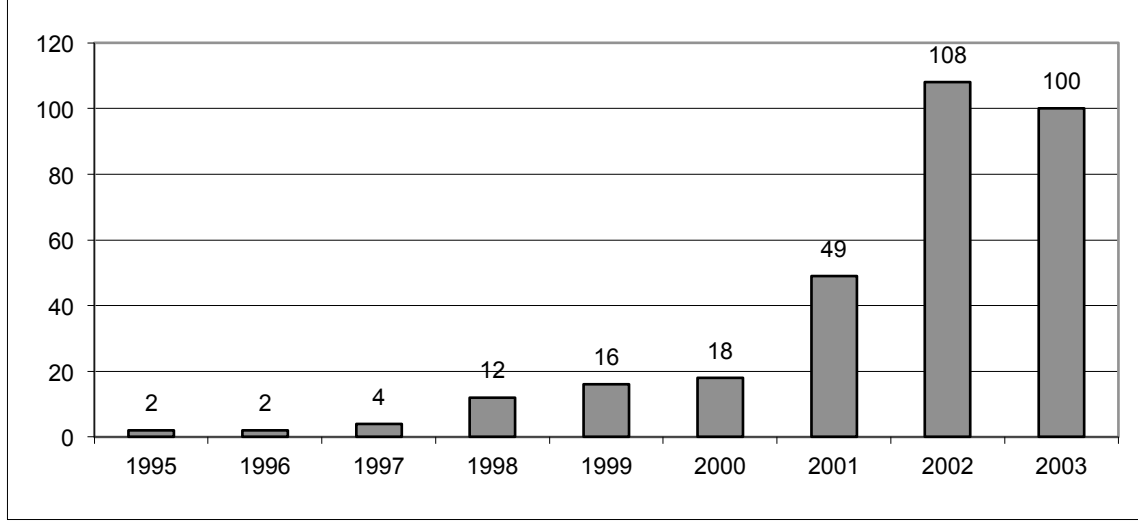
```

Şekil 2.10: Parçacık Sürü Optimizasyonu Algoritması

### 3.6.3. Uygulama Alanları

1995 yılında Kennedy ve Eberhart’ in çalışmalarıyla ortaya çıkan parçacık sürü optimizasyonu algoritması takip eden yıllar içerisinde araştırmacılar tarafında ilgi görmüş ve çeşitli alanlara uygulanmıştır. Yapılan araştırmaların bir kısmı algoritmayı geliştirmek ve algoritma parametreleri üzerinde inceleme yaparak optimizasyon başarımını arttırmaya yönelik, diğer bir kısmı ise mevcut ya da özelleştirilmiş algoritmaların çeşitli problemlere uygulanması şeklindedir.

Şekil 2.11’ de yıllar itibari ile parçacık sürü optimizasyonu tekniği ile ilgili yayınlanmış çalışmaların sayıları görülmektedir. (Hu ve diğerleri, 2004)



Şekil 2.11: Parçacık sürü optimizasyonu tekniği ile ilgili yıl bazında yayınlanmış çalışma sayısı

Söz konusu metodoloji çeşitli mühendislik alanlarında, farklı problemler üzerinde kendi başına ya da mevcut diğer yöntemlerle birlikte karma çözümler oluşturmak üzere kullanılmıştır. Yapılan çalışmalar içerisinde literatürde sıkça referans gösterilen ve kendi alanında özgün olan belli başlı çalışmalar incelendiğinde yöntemin uygulanan problemlerde etkin ve hızlı çözümler sunduğu görülmektedir.

Xia ve Wu (2005), parçacık sürü optimizasyonu tekniği ile tavlama benzetimini birlikte kullanan melez bir yöntem geliştirerek, bunu çok amaçlı esnek atölye tipi çizelgeleme problemleri üzerinde incelemişler ve yöntemin uygulanabilirliğini ortaya koymuşlardır.

Utkan (2006) karesel atama problemlerinden tesis içi yerleşim problemini parçacık sürü optimizasyonu metodolojisi ile ele alarak, tesis içi yerleşiminde etkin sonuçlar veren bir yöntem geliştirmiştir.

Thai (2007) çeşitli PSO algoritmalarını tesis içi yerleşim problemlerinde uygulayarak, bu tip problemler için en etkili modelin hangisi olduğunu tartışmış ve kritik başarı faktörlerini incelemiştir.



Yoshida ve diğ erleri (2000) voltaj güvenliğini sağlamak üzere reaktif güç ve voltaj kontrol sistemlerinde parçacık sürü optimizasyonu tekniğini uygulamışlar ve ortaya koydukları modeli reaktif tabu arama metodolojisi ile karşılaştırmışlardır.

Ujjinn ve Bentley (2003) internet tabanlı bir öneri sistemi geliştirmişlerdir. PSO' nun veri madenciliği alanında ilk uygulamalarından biri olan bu çalışmada kullanıcıların kişisel tercihlerini analiz eden bir algoritma kullanılmıştır.

Bunların dışında PSO yöntemi, yapay sinir ağlarının eğitimi (Settles ve Rylander, 2002), Robot Kontrolü (Kaewkamnerdpong ve Bentley, 2005) , doküman yönetimi (Cui ve diğ ., 2005), portföy yönetimi (Filho, 2006), telekomünikasyon sistemleri (White, 1997) alanlarında da uygulanmıştır.

Kennedy (2000) çalışmasında PSO algoritmasının hız vektörü güncelleme fonksiyonunda bazı değişiklikler yapmıştır. Her bir bireyin kendi geçmiş en iyisini ve komşuluk kümesindeki ya da tüm sürüdeki en iyiyi göz önüne alarak yaptığı hız vektörü güncellemesi işlemi bireyleri öbekleyerek ve en iyi değerler olarak, her bireyin kendi öbeğinin geçmiş en iyi öbek merkezini kullanarak yeniden hesaplamıştır. Bu yaklaşım ile sürüdeki bireyler arama davranışlarına göre öbeklenerek, öbek içerisinde zayıflıkların etkisinin azaltılması amaçlanmıştır.

Li (2004) birden fazla optimum noktası olan problemlerin çözümü için bir PSO algoritması geliştirmiştir. Bu algoritma sürüdeki bireyleri birbirleri ile yakınlık ilişkilerine göre türlere ayırarak her türün kendi etrafındaki alanda optimizasyon yapmasını öngörmektedir. Bu sayede algoritmanın paralel olarak birden fazla optimum nokta bulması sağlanmıştır.

Shi ve Eberhart (2001) bulanık sayıları kullanan bir PSO algoritması geliştirmişler ve bu algoritmanın performansını standart algoritma ile karşılaştırmışlardır.

### 3.6.4. PSO Yöntemi ile Evrimsel Hesaplama Yöntemleri Arasındaki İlişki

Langdon ve Poli (2007) PSO yöntemi ile Diferansiyel Evrim, Evrimsel Programlama ve Genetik programlama arasındaki performans farklılığını incelemiştir. Bu çalışma neticesinde diferansiyel evrim yönteminin, ajanların büyük arama uzaylarında yayılma ve hızlı hareket etme konusunda PSO' nun gerisinde kaldığı görülmüştür. Ancak hız kısıtlaması tanımlanmamış bir PSO modelinin küresel optimumu bulma konusunda genetik programlamanın gerisinde kalabileceği, bazı durumlarda PSO yönteminin küresel optimuma çok yaklaşip ardından hız kısıtlaması olmaması durumunda ötesine geçtiği görülmüştür. Genetik programlama ise yineleme kuşakları boyunca aramasını bu noktada odaklamayı başarabilmiştir.

Castro (2002) evrimsel algoritmalar, sürü zekası algoritmaları ve yapay bağışıklık sistemleri algoritmalarını incelemiş ve çıkış felsefelerini karşılaştırmıştır. Angeline (1998) parçacık sürü optimizasyonu tekniği ile evrimsel optimizasyon algoritmalarının bazı temel metaforlarını bir araya getirerek seçilim mekanizması içeren bir parçacık sürü optimizasyonu tekniği geliştirmiştir. Ortaya konan model dört test fonksiyonunun üçünde standart sürü zekası algoritmasına göre daha iyi ya da çok yakın sonuçlar üretmiştir.

### 3.6.5. Performansı Artırmaya Yönelik Düzenlemeler

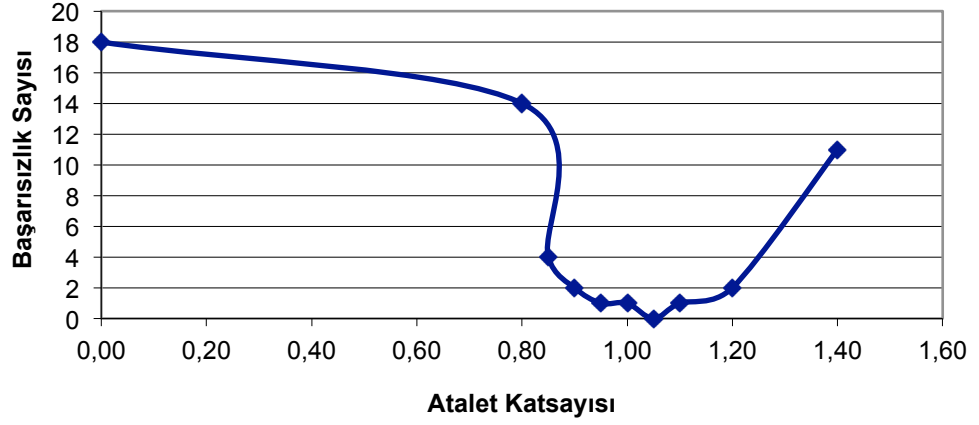
#### *Atalet Katsayısı*

Standart PSO algoritmasının çeşitli test fonksiyonlarının en uygun değerlerini bulma konusundaki performansı Shi ve Eberhart (1999)' ın çalışmasında incelenmiştir. Bunun yanı sıra standart algoritmanın performansını iyileştirmek amacıyla çeşitli fikirler ortaya konmuştur.

PSO algoritması üzerinde ilk yapılan değişikliklerden biri algoritmanın yayılma hızını iyileştirmek amacıyla bir atalet katsayısını modele eklemek olmuştur (Shi ve Eberhart, 1998). Atalet katsayısı algoritmanın en son yinelemedeki hızına etki eden bir ölçekleme faktörüdür. Atalet katsayısının ( $w$ ) yer aldığı değiştirilmiş hız hesaplaması formül 2.21' de gösterilmiştir.

$$v_d \leftarrow wv_d + c_1(p_d - x_d) + c_2(g_d - x_d) \quad (2.21)$$

Görüldüğü gibi  $w$ ' nin değerinin 1 olması durumunda orijinal hız denklemi elde edilir. Shi ve Eberhart (1998)  $w$  katsayısının etkilerini  $[0, 1.4]$  aralığında, 4 farklı test fonksiyonu üzerinde, en çok 4000 yinelemeli, 30'ar deneme ile incelemiştir. Şekil 2.12' de atalet katsayısının çeşitli değerleri için ortalama başarısızlık adetleri görülmektedir. Bu çalışmanın sonuçları  $w \in [0.8, 1.2]$  olarak seçilmesi durumunda algoritmanın en iyi noktalara daha hızlı yakınsadığını ancak  $w > 1.2$  durumunda hata oranının arttığını göstermiştir.



Şekil 2.12: Atalet katsayısının çeşitli değerleri için ortalama başarısız deney sayısı (Shi ve Eberhart, 1998).

Atalet katsayısı ( $w$ )' nin parçacık sürü optimizasyonu algoritmasının yakınsama davranışı üzerinde önemli bir etkisi vardır (Eberhart ve Shi, 2000). Atalet katsayısı bireyin önceki hızlarının o anki hızına olan etkisini kontrol etmek için kullanılır. Bu sayede  $w$  parametresi sürünün arama yeteneğinin küresel ya da yerel karakteristiklere ne ölçüde sahip olacağını belirler. Büyük bir atalet katsayısı değeri küresel araştırmayı, küçük değerler ise yerel aramayı tetikler. Uygun belirlenmiş bir  $w$  katsayısı sürünün küresel ve yerel arama yeteneği arasında denge sağlayarak en uygun çözüme ulaşılmasına yardımcı olur. Eberhart ve Shi' nin çalışmaları atalet katsayısı  $w$  için 0,7298 ve aynı zamanda  $c_1=c_2=1,49618$  değerlerinin etkin bir yakınsama davranışı ortaya çıkardığını göstermiştir. Bunun yanı sıra araştırmalar atalet katsayısının, arama

sürecinin başında yüksek bir değer olarak atanarak başlangıçta küresel aramanın sağlanması ardından kademe kademe düşürülerek süzölmüş sonuçlara ulaşılmasının da etkili bir yöntem olduğunu göstermiştir (Suganthan, 1999, Naka ve diğ., 2001).

#### *Hız Kısıtı*

$w$  katsayısı için 1'den büyük bir değer belirlenmesi durumunda ilgili birey için sürekli artan bir hızdan bahsedilebilir. Bu durum bireylerin daha sık çözüm uzayının dışına çıkmaları ihtimalini doğurur. Bireylerin hareketlerini kısıtlayarak çözüm uzayının dışına çıkmalarını engellemek için PSO algoritmasında hız vektörlerinin alabileceği hızın en büyük değerini belirleyen bir  $v_{max}$  parametresi tanımlanmıştır.  $v_{max}$  modelin her boyutu için farklı değeri olan bir en büyük hız vektörüdür. Örneğin arama uzayının herhangi bir  $d$  boyutunun genişliği  $[0, 5]$  aralığı ise ve bu boyutta en büyük hız  $[2,5]$  olarak tanımlanmışsa  $v$  vektörünün değeri 2,5' in üzerine çıktığında tekrar 2,5 değerine çekilir.

Atalet katsayısı bir önceki yinelemede elde edilen hızın ne oranda yeni yinelemeye aktarılacağını belirlemek için kullanılır.  $c_1 = 0$  ve  $c_2 = 0$  olarak kabul edilirse,  $w > 1$  olması durumunda bireylerin önceden belirlenen  $v_{max}$  hızına doğru hızlanacağı görülebilir.  $w < 1$  olması durumunda ise bireyler zamanla hızları sifira inecek kadar yavaşlayarak hareket edeceklerdir.  $c_1, c_2 \neq 0$  olması durumunda ise algoritmanın davranışını tahmin etmek zordur.

Başka bir çalışmada ise  $v_{max}$  ile atalet katsayısı arasındaki bağlantı incelenmiştir (Shi ve Eberhart, 1998). Bu çalışma neticesinde  $v_{max} = x_{max}$  olması durumunda  $w = 0.8$ ' in daha iyi sonuçlar doğurduğu görülmüştür. Ancak en iyi performans atalet katsayısının zamanla 0.9' dan 0.4' e düşürüldüğü 1500 yinelemeli modellerde sağlanmıştır.

PSO Algoritmasının performansını inceleme ve geliştirmeye yönelik olarak yapılan çalışmaların bir kısmı ise sürüdeki bireylerin yörünge davranışlarını analiz ederek arama uzayındaki en iyi noktalara doğru hareket etmelerini sağlayacak temel davranış faktörlerini anlamak ve yakınsamalarını hızlandırmaktır. Bu konu ile ilgili yapılan ilk çalışmalar Ozcan ve Mohan (1998,1999)' a aittir. Çalışmalarında tek boyutlu tek bireyli bir sürünün yörünge davranışlarından yola çıkılarak, bulgular, çok boyutlu, çok bireyli

PSO sistemlerine genelleştirilmiştir. Bu çalışmaların en önemli sonucu bireylerin yörüngelerinin periyodik sinüs dalgaları şeklinde hareket ettiğinin ortaya çıkması olmuştur. En uygun değeri arama işlevinin ise rastgele olarak diğer bir dalgayı yakalayıp, kendi frekans ve şiddetini ona göre düzenlemek ile sağlandığı görülmüştür.

Trelea (2003) model parametrelerinin aldıkları değer ile bireylerin yörüngelerindeki genişleme-yakınsama davranışları arasındaki ilişkiyi incelemiştir.

Clerc ve Kennedy (2002) ve Trelea (2003) 'nin çalışmalarında yapılan matematiksel analizler, algoritmanın önceden belirlenmesi gereken  $c_1$  ve  $c_2$  parametrelerinin birbirlerinden bağımsız olarak belirlenmemesi gerektiğini göstermektedir. Bu çalışmalarda [0,7 1,47] ve [0,8 1,62] değer çiftlerinin genellikle en iyi sonuçları veren değerler olduğu sonucuna varılmıştır.

Shi ve Eberhart (1998) ise atalet katsayısı için en büyük ve en küçük değerleri;

$$\begin{aligned} w_{\max} &= 0.9 \\ w_{\min} &= 0.4 \end{aligned}$$

olarak önermişlerdir.

### 3.6.6. 1'' Tip Parçacık Sürü Optimizasyonu Modeli

Clerc ve Kennedy (2002) PSO algoritmasını geliştirmek amacıyla parametre değerlerinin model çıktısı üzerindeki etkilerini matematiksel olarak ele almışlardır. Standart modeldeki üç rastgele değişken değerini ikiye ( $c_{\max}, \chi$ ) indiren bir algoritma geliştirmişlerdir. Bu modelde hız güncelleme formülü, işlemin tamamına etki eden bir  $\chi$  'daraltma' katsayısı ile çarpılır (formül 2.30).

$$v_{id} \leftarrow \chi(v_{id} + c_1\phi(p_{id} - x_{id}) + c_2\phi(g_{id} - x_{id})) \quad (2.30)$$

Formüldeki parametreler ise;

$$\chi = \frac{2}{\left| 2 - c_{\max} - \sqrt{c_{\max}^2 - 4c_{\max}} \right|} \quad (2.31)$$

$$c_{\max} = c_1 + c_2, \quad c_{\max} > 4$$

olarak hesaplanır.

Shi ve Eberhart (2001) çalışmalarında, en iyi çözüme  $c_1 = c_2 = 2.05 \Rightarrow c_{\max} = 4.1$  ve dolayısıyla  $\chi = 0.7298$  kullanarak ulaşmışlardır. Daraltma katsayısının kullanım amacı, bu modelde hız vektörünün değerini sınırlayacak bir  $v_{\max}$  sabitine ihtiyaç duyulmamasıdır. Bu model, standart modelden genellikle daha iyi sonuçlara daha hızlı ulaşmıştır. Ancak bazı test problemlerinde bireylerin arama uzayının uç noktalarına gidişi engellenememiş ve model başarısız olmuştur. Bunun üzerine Shi ve Eberhart standart modeldeki hız kısıtlayıcı sabitin  $v_{\max} = x_{\max}$  olarak modele dahil edilmesi gerektiğini söylemiş ve bu şekilde tüm test fonksiyonlarında optimum değere yakınsama performansını iyileştirmişlerdir.

Kennedy ve Rui (2002) çalışmasında 5 test fonksiyonu üzerinde çeşitli komşuluk topolojilerini denemiş ve tip 1'' türü algoritmalarda parametreler için optimum değerleri;

$$c_{\max} = 2.01$$

$$\chi = 0.729844$$

olarak hesaplamıştır.

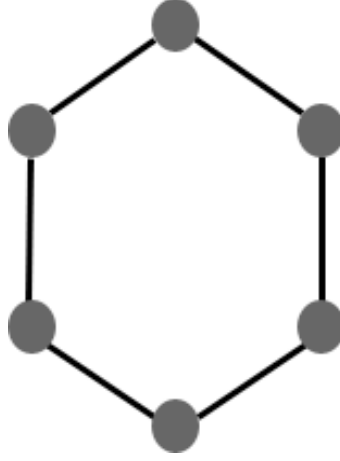
### 3.6.7. Sürünün En İyisi İçin Önerilen Yöntemler:

Çözüm uzayı içerisindeki bireyler, mevcut hareket doğrultusunu gösteren kendi hız vektörleri, aynı bireyin geçmiş en iyi konumunu (kendi hafızası) gösteren konum vektörü ve bireyin komşuları ya da sürünün tümü içerisinde o ana dek bulunan en iyiyi gösteren konum vektörlerinin üçünün birleşimi uyarınca hareket ederler. Bireyin komşuluk kümesi içerisindeki ya da tüm sürüdeki diğer bireylerle etkileşime girmesi sosyal ağlardaki bilginin akışı metaforu ile benzeşim göstermektedir (Kennedy, 1999).

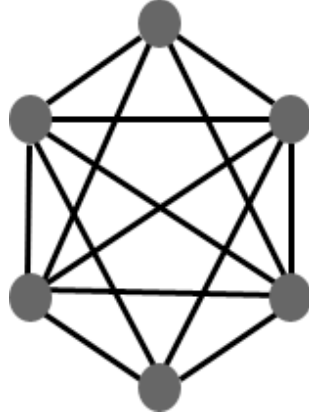
Bir grup içersisindeki iletişim ve özellikle grubun performansı, sosyal ağın yapısından yüksek oranda etkilenirken sosyal ağlardaki bilgi akışı da ağın bazı temel özelliklerinden etkilenmektedir. İlk etken ağdaki bireyler arasındaki bağlantı derecesidir. Bireylerin komşuluk kümelerinin en iyisinden etkilendikleri sürü zekâsı optimizasyonu modellerinde her bir birey  $K$  adet komşusu arasından en iyi  $f$  değerine sahip olana bakacaktır. Dolayısıyla bu modelde ki  $K$  parametresi, bireyin, sürüdeki tüm bireylerin en iyisine baktığı yöntem ile komşuluk kümesine baktığı yöntem arasındaki temel farklılığı yaratmaktadır.

Yapılan çalışmalarda komşuluk durumu üzerinde çeşitli alternatif yöntemler geliştirilmiştir. Komşuluk şu şekillerde ifade edilebilir (Nedjah ve Mourelle, 2006);

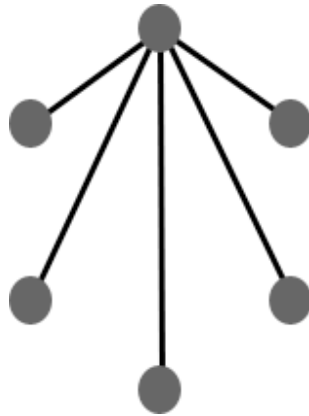
1. *i*. bireyin komşuları: bireyin kendisi ve bireye en yakın olan  $K$  adet diğer bireyden oluşur. Eğer  $K=2$  ise bu yapıya halka yapısı denir (şekil 2.13). Bu yaklaşımda bireyler arasındaki bağlantılar oluşturulurken iki farklı yol izlenebilir.
  - $K$  adet birey algoritmanın başlangıcında sürü içerisinden rastgele olarak seçilir.
  - $K$  adet birey her yinelemede sürü içerisinden rastgele olarak seçilir
2. Sürüdeki tüm bireyler birbirleri ile bağlantılıdır. Bu yapıya *gbest* yapısı da denir (şekil 2.14).
3. Odak Yapısı: Sürüdeki bir birey diğer tüm bireylere bağlıdır ve diğer tüm bireyler yalnızca bu bireye bağlıdır (şekil 2.15) (Bergh, 2001).
4. Bireyin komşuları, bireyin çevresindeki diğer bireylerden belirli bir yapıda (yıldız, piramit, kutu) seçilirler (Kennedy ve Rui, 2002; Rui, 2004).



Şekil 2.13: Komşuluk ilişkisi olarak halka yapısı ( $K=2$ )



Şekil 2.14: Komşuluk ilişkisi olarak *gbest* yapısı



Şekil 2.15: Komşuluk ilişkisi olarak odak yapısı



Bu komşuluk yapılarından 1 numaralı yapıda sürü içerisinde birbirinden uzak olan bireylerin aynı zamanda birbirlerinden bağımsız oldukları da görülür. Bu durum sürünün bir parçasının bir yerel optimuma yakınsarken diğer bir parçanın da başka bir yerel optimuma yakınsayabileceği ya da aramaya devam edebileceği sonucunu doğurur. Bu yapıda etkileşim bir komşuluktan diğerine doğru hareket edecek ve bu sırada eğer bir birey problemin gerçek optimumunu (genel optimum) bulursa, tüm bireyleri kendine doğru çekmeye başlayacaktır.

Odak yapısı tüm bilginin sürüde önceden seçilmiş olan tek bir birey üzerinden geçiş yaptığı yapıyı temsil etmektedir. Bu odak birey sürüdeki tüm diğer bireylerin performanslarını değerlendirerek kendini en iyi olan noktaya doğru yönlendirmektedir. Burada odak birey bir çeşit tampon ya da filtre görevi görerek iyi çözümlerin sürüye yayılımını yavaşlatan bir etki sergilemektedir.

Kennedy (1999), bağlantı sayısının az olduğu modellerin çok boyutlu problemlerde daha iyi sonuçlara ulaşırken, birbirleriyle bağlantısı fazla olan bireylerden oluşan modellerin düşük boyutlu problemlerde daha etkin olduğunu göstermiştir.

Kennedy ve Rui (2002) çeşitli komşuluk yapılarının modelin performansı üzerindeki etkilerini araştırdıkları çalışmalarında söz konusu yaklaşımlar arasında en kötü sonuçların tüm bireylerin diğer tüm bireyler ile bağlantıda olduğu durumda ortaya çıktığını göstermişlerdir.

### **3.6.8. PSO Tabanlı Öbekleme Analizi Yöntemleri**

Literatürde PSO yönteminin kullanıldığı çeşitli öbekleme çalışmaları bulunmaktadır. Ancak PSO' nun kullanımı genellikle mevcut veya değiştirilmiş bir öbekleme algoritmasının sonuçları üzerinde en uygun şekle sokma şeklindedir. Ji ve diğerleri (2004) mobil ağların optimizasyonu ile ilgili çalışmalarında ağırlıklandırılmış öbekleme algoritmasının sonuçları üzerine PSO yöntemini uygulayarak mobil ağların dağıtımını gerçekleştirmişlerdir. Mobil ağlardaki bağlantı noktaları altyapısız ağ elemanları olduğundan enerji tasarrufu mobil ağların optimizasyonunda önemli bir ölçüttür.

Dolayısıyla ağı idare eden optimizasyon algoritmasının işlem gücü gereksinimi düşük olmalı ve hesaplama ihtiyacı ağdaki elemanlar arasında dengeli dağıtılmış olmalıdır. PSO merkezi olmayan, yinelemeli yapısıyla bu tip problemlerin çözümünde etkili bir yöntem olarak ortaya konmuştur.

Correa ve diğ. (2006) kesikli parçacık sürü optimizasyonu algoritmasını kullanarak, biyolojik veri içeren veri kümelerinde öbikleme analizi ile örnek türlerinin kategorizasyonu problemini ele almışlardır. Bu çalışmada bireylerin konum vektörlerini oluşturan her bir boyut, incelenen veri kümesinin sahip olabileceği çeşitli karakteristikleri temsil eder. Bu karakteristik özellikler birer tamsayı olarak tanımlanmıştır ve bireyler, arama uzayında, farklı karakteristik yapı gruplarını temsil eden aday çözümler oluşturacak şekilde bu tamsayı değerlerinin değişimi ile hareket ederler.

Chen ve Ye (2004) PSO yönteminin öbikleme analizinde kullanımını ele almış ve 4 farklı veri kümesinde PSO öbikleme algoritmasını test etmişlerdir. Geliştirilen modelde her bir birey olası bir çözümü ifade etmektedir. Dolayısıyla 3 öbekli 2 boyutlu bir problemde bireylerin  $3 \times 2 = 6$  adet boyutu vardır. Bu boyutlar her iki tanesi bir öbek merkezini gösterecek şekilde bölümlendirilmiştir. Uygunluk fonksiyonu olarak her öbekteki bireylerin öbek merkezine öklidyen uzaklığını minimize eden bir fonksiyon seçilmiştir. Algoritma öbek merkezlerinin konumlarındaki değişim miktarı önceden belirlenen bir değerin altına düşene kadar çalıştırılmaktadır. Bu metot standart PSO algoritması ile kullanılarak ortaya konan öbikleme yönteminin performansı incelenmiştir. Algoritmanın parametre değerleri  $c_1 = c_2 = 1,5$  ve  $w = 0,75$  olarak kabul edilmiştir.

Merwe ve Engelbrecht (2003) çalışmalarında melez bir öbikleme yöntemi sunmuşlardır. Klasik K-Ortalamlar yönteminin sonuçlarına PSO' nun uygulandığı bir model geliştirilmiştir. Yapılan analizler PSO tabanlı melez öbikleme mekanizmasının klasik K-Ortalamlar yönteminden daha iyi sonuçlar ortaya koyduğunu göstermiştir.

Cui ve diğ. (2005) doküman öbikleme problemini parçacık sürü optimizasyonu yöntemi ile ele almışlardır. Doküman niteliklerinin birer boyut olarak tanımlandığı vektör

uzayında her bir doküman bir nokta ile temsil edilecektir. Geliştirilen algoritmanın amacı ise bu vektör uzayındaki öbek merkezlerini tespit ederek; öbek merkezlerinin belirttiği nitelikteki dokümanları birbirinden ayrı olarak sınıflandıran öbekleri tespit etmektir. Sunulan modelde bireylerin boyutları öbek merkezlerinin konumlarını göstermektedir. Dolayısıyla  $k$  adet niteliğe göre yapılacak bir öbekleme analizi çalışmasında, dokümanlar,  $c$  adet öbeğe ayrılmak isteniyorsa her bir bireyin  $k \times c$  adet boyutunun olması gerekir. Her bireyin ayrı bir aday çözümü temsil ettiği PSO algoritmasında en iyi çözüme ulaşan bireyin boyutları sonuç öbeklerin merkezlerini gösterecektir. Algoritmanın hesaplama süresini kısaltmak için bu çalışmada, Merwe ve Engelbrecht (2003)' in çalışmasından farklı olarak, PSO algoritması belli bir yineleme sayısı kadar çalıştırıldıktan sonra elde edilen sonuç öbek merkezleri K-Ortalamalar algoritmasında girdi olarak verilerek en iyi öbek merkezleri tespit edilmeye çalışılmıştır.

Omran ve diğerleri (2002, 2005) PSO tabanlı bir resim tanıma ve sınıflandırma yöntemi geliştirmişlerdir. İz tanıma ve resim sınıflandırma problemleri öbek sayısının önceden bilinmediği bir problem tipi olduğundan ortaya koydukları model dinamik olarak öbek sayısını ayarlayan bir iki değerli (1-0) PSO modelidir.

### 3.6.9. PSO Tabanlı Tesis Yeri Seçimi Yöntemleri

Parçacık sürü optimizasyonu yönteminin tesis yeri seçimi problemlerine uygulandığı çeşitli çalışmalar mevcuttur. Bu çalışmalar genellikle tesis yeri seçimi problemini bir atama problemi olarak ele almaktadırlar.

Guner ve Sevkli (2008) ile Sevkli ve Güner (2006) kapasite kısıtsız tesis yeri seçimi problemi için PSO tabanlı bir çözüm önerisi sunmuştur. Standart PSO algoritmasını temel alan bu çalışmada  $n$  adet tesisin  $m$  adet müşteriye hizmet verdiği bir sistemde minimum maliyeti sağlayacak tesis / tesislerin seçilmesi hedeflenmiştir. Bu amaçla oluşturulan PSO modelinde her bir birey  $n$  boyutlu olarak tanımlanmıştır. Bu boyutlarda modelin alabileceği değerler 1 veya 0 olarak sınırlandırılarak, ilgili tesisin açık ya da kapalı olması durumu temsil edilmiştir. Her bir bireyin aday bir çözümü temsil ettiği PSO algoritması en iyi açık/kapalı tesisler bileşimini bulmak için önceden belirlenen bir yineleme sayısına ulaşıncaya kadar çalıştırılmıştır.

Bu tez çalışmasında sunulan odak bireyli sürü zekası optimizasyonu algoritması bahsedilen çalışmalardan farklı olarak bir atama algoritması değildir. Sürekli uzayda tanımlanmış bir maliyet fonksiyonunun minimum değerine ulaşmayı sağlayacak önceden belirlenmiş sayıda tesisin, arama uzayındaki en uygun konumlarını bulmaya çalışan bir algoritmadır.

Yapıcıoğlu ve diğerleri (2004) yarı-istenen tesisler için tesis yeri seçiminde PSO yöntemini kullanmışlardır. Bu çalışmada Tip1'' modeli üzerinde tesisin ulaştırma maliyeti ve yerleşim yerlerine yakınlığının ortaya çıkardığı zararın etkisini kapsayan iki kriterli bir amaç fonksiyonu kullanılmıştır. Geliştirilen model karşılaştırılan diğer sezgisel yöntemlerden daha iyi sonuçlar vermiştir.

Yano ve diğ. (2008) tesis yeri seçimi problemi için melez parçacık sürü optimizasyonu algoritmasını önermişlerdir. Tesisler ile tesislerin hizmet sağladıkları müşterileri arasındaki toplam mesafeyi en küçükmeye dayalı bu algorithmada her tesise eşit sayıda müşterinin atanmasını sağlayacak bir model geliştirilmiştir. Merwe ve engelbracht (2003) ve Cui ve diğ. (2005)' in çalışmalarına benzer bir şekilde oluşturulan parçacık sürü optimizasyon modelinde bireylerin boyutları öbek merkezlerinin konumlarını göstermektedir. Dolayısıyla  $n$  adet tesisin  $X$ - $Y$  koordinat düzlemindeki konumlarını gösteren her bir bireyin konum vektörü  $2n$  adet boyuttan oluşmaktadır.

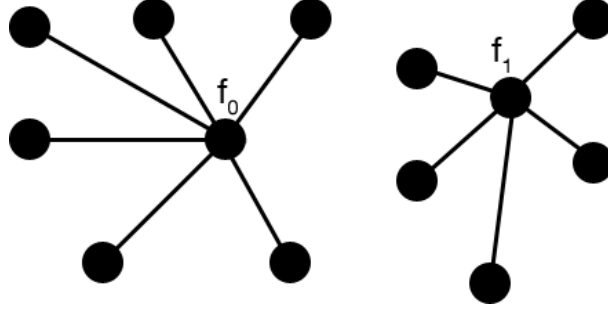
## 4. MALZEME VE YÖNTEM

### 4.1. ODAK BİREYLİ SÜRÜ OPTİMİZASYONU ÖBEKLEME YÖNTEMİ

Bu çalışmada sunulan odak bireyli sürü optimizasyonu (OBŞO) (Swarm Optimization with Focal Particles) algoritması belirli sayıdaki veri elemanının öbeklenmesi amacıyla geliştirilmiştir. Odak bireyli sürü optimizasyonu (OBŞO) algoritması yeni bir öbekleme tekniğidir. Sürü zekası yöntembiliminin temel metaforlarına bağlı kalarak, sürüdeki bireyler arasındaki iletişimin yapısını birden fazla odak birey üzerinden sağlayan bu yaklaşımın iyi bir öbekleme analizi tekniği olduğu ve tedarik zinciri yönetiminde yer seçimi problemlerinin çözümünde başarıyla uygulanabileceği ortaya konmuştur.

Literatürde odak bireyin bir komşuluk yapısı olarak tanımlandığı çeşitli çalışmalar mevcuttur (Sierra ve Coello, 2006, Kennedy ve Rui, 2002). Ancak sürünün bireyleri içerisinde birden fazla odak bireyin aynı anda bulunduğu ve bireyler arasındaki iletişim ağının öbekleme analizinin temel hedeflerine ulaşacak doğrultuda dinamik olarak güncellendiği bir çalışma yer almamaktadır.

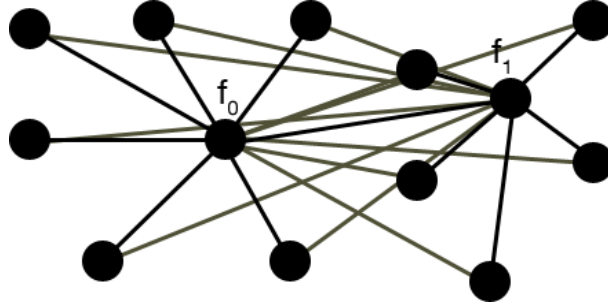
Odak bireyli sürü optimizasyonu algoritması temel olarak önceden belirlenmiş miktarda odak bireyin sürü içerisinde seçildiği ve tüm iletişimin bu odak bireyler üzerinden gerçekleştirildiği bir sürü zekası modelidir. Modelin komşuluk yapısı şekil 3.1' de görülmektedir.



Şekil 3.1:  $l=13$  olan bir modelde OBSO yönteminin algoritmanın herhangi bir  $t \geq 1$  yinelemesindeki komşuluk yapısı.  $f_0$  ve  $f_1$  odak bireyleri temsil etmektedir.

#### 4.1.1. Modelin Yapısı

OBSO yönteminin başlangıç anında  $l$  adet bireyden oluşan bir sürü tanımlanır. Sürü içerisinde rastgele seçilen  $c$  adet birey, odak bireyler olarak kabul edilir. Ardından bireyler arasındaki bilgi alış verişinin yapısını belirleyecek olan bilgi bağlantıları tanımlanır. OBSO yönteminin başlangıç durumunda tüm bireyler tüm odak bireyler ile bağlantıdadır (Şekil 3.2).



Şekil 3.2:  $t = 0$  anında OBSO algoritmasında komşuluk yapısı

Sürüyü oluşturan bireylerin konumları arama uzayındaki muhtemel öbek merkezi çözümlerini temsil eden  $d$  boyutlu vektörlerle tanımlanmıştır. Odak bireylerin yapısal olarak sürüdeki diğer bireylerden herhangi bir farklılığı bulunmamakla birlikte en son yineleme ile birlikte sonuç öbek merkezlerinin konumlarını temsil eden vektörler odak bireyler tarafından temsil edilirler. OBSO algoritmasının parametreleri şu şekilde tanımlanmıştır;

Arama uzayı için;

$d$  : Boyut sayısı

Öbeklenecek olan veri elemanları için;

- $z_i$  :  $i$ . veri elemanının konum vektörü  
 $z_{id}$  :  $i$ . veri elemanının konum vektörünün  $d$ . boyutu  
 $S_Z$  : Veri elemanları kümesi  
 $m$  : Veri elemanı sayısı  
 $c$  : Öbek sayısı, Odak birey sayısı

OBSO yöntemi değişkenleri;

- $l$  : Birey sayısı  
 $b_i$  :  $i$ . birey  
 $S_S$  : Sürüyü oluşturan bireyler kümesi  
 $S_S = \{b_1, b_2, \dots, b_l\}$   
 $x_i$  :  $i$ . bireyin konum vektörü  
 $x_i = \{x_{i1}, x_{i2}, \dots, x_{id}\}$   
 $v_i$  :  $i$ . bireyin hız vektörü  
 $v_i = \{v_{i1}, v_{i2}, \dots, v_{id}\}$   
 $p_i$  :  $i$ . bireyin o ana kadar gösterdiği en iyi performansın konum vektörü  
 $p_i = \{p_{i1}, p_{i2}, \dots, p_{id}\}$   
 $g_i$  :  $i$ . bireyin komşuları içerisinde en iyi performansa sahip olanın en iyi performansının konum vektörü  
 $g_i = \{g_{i1}, g_{i2}, \dots, g_{id}\}$   
 $J(x)$  :  $x$  vektörüne göre amaç fonksiyonu değeri

Öbekleme parametreleri;

$S_i$  :  $i$ . öbeği oluşturan veri elemanları kümesi

$$S_i \in S_z$$

$f_i$  :  $i$ . odak birey

$S_F$  : Odak bireyler kümesi

$$S_F \in S_S$$

$$S_F = \{f_1, f_2, \dots, f_c\}$$

$t$  : İterasyon indisi

$l$  adet bireyden oluşan sürünün, başlangıç anında, ilk  $c$  adet bireyi odak bireyler olarak tanımlanır. Her bir odak birey ayrı birer öbek merkezini temsil edecektir.

Algoritmanın birey sayısı belirlenirken  $l \geq 2c$  olmalıdır. Zira her bir odak birey kendisi dışında en az bir arama yapacak bireye ihtiyaç duyar.

#### 4.1.2. Başlangıç İşlemleri

İlk olarak öbeklenecek veri kümesi belirlenmelidir. Bu aşamada arama uzayının sınır değerleri veri kümesindeki en küçük ve en büyük değerler olarak belirlenir (formül 3.1).

$$\begin{aligned} x_{\min} &= \text{MIN}(z_{id}) \\ x_{\max} &= \text{MAX}(z_{id}) \end{aligned} \quad (3.1)$$

Clerc (2002) daraltma katsayılı modelinde hız vektörünün en büyük değerini  $v_{\max} = x_{\max}$  olarak tanımlamıştır. OBSO algoritmasında da bireylerin arama uzayı dışına çıkışını kontrol altında tutmak amacıyla aynı kabul yapılmıştır. Veri elemanlarının tanımlanmasından sonra sürüyü oluşturan bireylerin başlangıç konum ve hız vektörleri formül 3.2' ye göre oluşturulur.

$$\begin{aligned} x_{id} &= x_{\min} + ((x_{\max} - x_{\min}) \times \varphi) \\ v_{id} &= \frac{x_{\max} - x_{\min}}{2} \times \varphi \end{aligned} \quad (3.2)$$

Formül 3.2' de yer alan  $\varphi$  parametresi  $[0 \ 1]$  aralığında düzgün dağılıma uygun rastgele bir değerdir. Başlangıç durumunda sürünün ilk  $c$  adet bireyi odak bireyler olarak



tanımlanır (formül 3.3). Bilgi bağlantıları sürüdeki tüm bireyler için, başlangıç durumunda, tüm odak bireylerle bilgi alış verişi gerçekleşecek şekilde tanımlanır.

$$S_F = \{b_1, b_2, \dots, b_c\} \quad (3.3)$$

Sürüyü oluşturan bireylerin başlangıç konum ve hızları formül 3.2'ye göre belirlendikten sonra bu konumlara göre her bir bireyin amaç fonksiyonu değeri  $J(x_i)$  hesaplanır. Amaç fonksiyonu, veri elemanlarını, öbekleme yapılmak istenen kritere göre sürüdeki bireylerin, alternatif öbek merkezi konumları olarak değerlendirildiği herhangi bir fonksiyon olabilir. Bu çalışmada OBSO yönteminin bir öbekleme algoritması olarak performansını incelemek amacıyla amaç fonksiyonu olarak niceleme hatası fonksiyonu ve Dunn indeksi; tedarik zincirinde tesis yeri seçiminde tesis yerlerinin ve tesis öbeklerinin tespiti amacıyla da toplam ulaştırma maliyeti fonksiyonu olarak belirlendiği farklı problem tipleri ele alınmıştır.

Algoritmanın işleyişi ardışık yinelemelerden oluşmaktadır. Başlangıç işlemleri tamamlandıktan sonra sonlandırma kriteri gerçekleşene kadar her yinelemede, öncelikle her bir  $i$  bireyinin bilgi alış verişinde olduğu diğer bireyler değerlendirilerek bireyin  $g_i$  vektörü formül 3.4' deki gibi güncellenir.

$$g_{id} = \begin{cases} p_{id} & | J(p_i) \leq J(g_i) \\ g_{id} & | J(p_i) > J(g_i) \end{cases} \quad (3.4)$$

Ardından formül 3.5 kullanılarak her bir  $i$  bireyinin hız vektörü hesaplanır.

$$v_{id} \leftarrow \begin{cases} c_1 \varphi(p_{id} - x_{id}) + c_2 \varphi(g_{id} - x_{id}) & | b_i \in S_F \\ \chi(v_{id} + c_1 \varphi(p_{id} - x_{id}) + c_2 \varphi(g_{id} - x_{id})) & | b_i \notin S_F \end{cases} \quad (3.5)$$

Formül 3.5'de görüldüğü gibi odak bireylerin hız vektörleri hesaplanırken, kendi mevcut hızları dikkate alınmaz. Odak bireyler yalnızca komşuluklarındaki diğer bireylerin en iyi konumlarından ve kendi geçmiş en iyi konumlarından etkilenirler. OBSO yönteminde, odak bireylerin arama uzayında rastgele hareketinin tanımlanmadığı ve tanımlandığı iki farklı durum ele alınabilir.

Hız vektörleri hesaplanan bireylerin bir sonraki aşamada yeni konumları formül 3.6'daki gibi tespit edilir.

$$x_{id} = x_{id} + v_{id} \quad (3.6)$$

Sürüdeki tüm bireylerin yeni konumları hesaplandıktan sonra her bir birey için amaç fonksiyonu değeri hesaplanır. Bu aşamada ele alınan her birey için öncelikle bireyin tüm bilgi bağlantıları iptal edilir. Ardından birey yalnızca kendisine en yakın odak birey ile bilgi alış verişinde bulunacak şekilde bilgi bağlantısı, bu odak bireyle, iki yönlü olarak tekrar kurulur (formül 3.7).

$$\begin{aligned} f' &= \text{MIN}(\text{dist}(b, S_F)) \\ b &\rightarrow f' \\ f' &\rightarrow b \end{aligned} \quad (3.7)$$

Formül 3.7'deki *dist* fonksiyonu *b* bireyi ile odak bireyler arasındaki uzaklığı hesaplamaktadır ve formül 3.8' de gösterilmiştir.

$$\text{dist}(b, S_F) = \sqrt{\sum_{\forall x_j \in S_F} \sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (3.8)$$

Şekil 3.3' de OBSO yönteminin sözde kodu verilmiştir. OBSO yönteminin komşuluk yapısının standart algoritmadan farkı bireyler ile odak bireyler arasındaki bilgilendirme bağlantılarının kurulma şeklidir. Standart algoritmada önceden tanımlanmış bir komşuluk yapısı çerçevesinde bireyler arasındaki bağlantıları kurulurken, OBSO yönteminde, ilk olarak tüm bireylerin tüm odak bireylerle bağlantısı gerçekleştirilmekte, yinelemeler başlamadan önce bir kez tüm bireylerin *J* amaç fonksiyonu değeri hesaplanarak, mevcut bütün bağlantılar silinmekte ve ardından her bir birey yalnızca kendisine en yakın odak birey ile bilgi bağlantısı kuracak şekilde komşuluk yapısı yeniden oluşturulmaktadır. Algoritmanın devam eden yinelemeli aşamalarında da standart algoritmadan farklı olarak her *J* amaç fonksiyonu değerlendirmesinin ardından bireylerin mevcut bilgi bağlantıları silinerek komşuluk yapısı yeniden tanımlanmaktadır.

```

Veri elemanlarını yükle:  $Z_i$ 
 $x_{\min}, x_{\max}, v_{\max}$  değerlerini hesapla
Sürünün başlangıç durumunu oluştur:  $x_i, v_i$ 
Odak bireyleri tanımla:  $S_F = \{b_1, b_2, \dots, b_c\}$ 
Komşuluk yapısını tanımla:
{ Her bir  $f$  odak bireyi için
  { Her bir  $b$  bireyi için
     $f \rightarrow b$ 
     $b \rightarrow f$ 
  }
}
Başlangıç durumunu oluştur:
{ Her bir  $i$ . birey için
  Amaç fonksiyonu değerini hesapla:  $J(x_i)$ 
  Bireyin tüm bilgi bağlantılarını kaldır
  Bireyi kendisine en yakın konumdaki odak bireye ( $f$ )
  bağla
     $f \rightarrow b_i$ 
   $b_i \rightarrow f$ 
  Başlangıç anındaki amaç fonksiyonu değerini bireyin en
  iyi değeri olarak belirle:  $p_i = x_i$ 
}
YAP: Sonlandırma kriteri gerçekleşmediği sürece:  $t \leq t_{\max}$ 
{{ Her bir  $i$ . birey için
   $g_i = \text{MIN}(x_{\text{komşuluk}})$ 
}
{ Her bir  $i$ . birey için
  { Her bir  $d$  boyutu için
    Formül 3.5' e göre hızı hesapla
    Eğer  $v_{id} > v_{\max} \Rightarrow v_{id} = v_{\max}$ 
    Formül 3.6' ya göre konumu hesapla
    Eğer arama uzayının dışına çıkıldı ise
       $x_{id} \leftarrow [x_{\min}, x_{\max}]$ 
    }
  }
{ Her bir  $i$ . birey için
  Amaç fonksiyonu değerini hesapla:  $J(x_i)$ 
  Bireyin tüm bilgi bağlantılarını kaldır
  Bireyi kendisine en yakın konumdaki odak bireye ( $f$ )
  bağla
     $f \rightarrow b_i$ 
     $b_i \rightarrow f$ 
  Eğer  $J(x_i) < J(p_i) \Rightarrow p_i = x_i$ 
}
}
 $t = t + 1$ 

```

Şekil 3.3: OBSO yönteminin sözde kodu

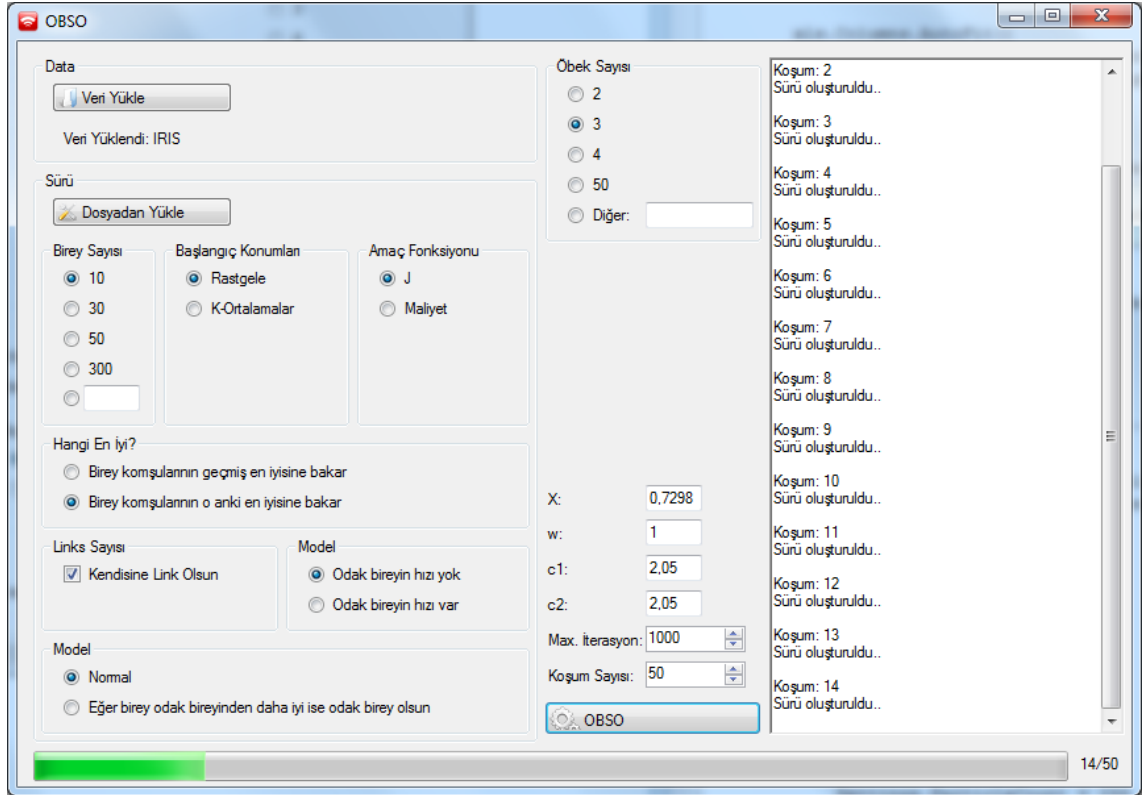
## 5. BULGULAR

Literatürde bir öbikleme analizi yönteminin başarımını incelemek amacıyla çeşitli geçerlilik indeksleri tanımlanmıştır. Geçerlilik indeksleri, öbikleme neticesinde ortaya çıkan öbeklerin kendi içlerinde ve birbirleri arasındaki farklılıkları değerlendirerek, birden fazla yöntemin birbirleri ile karşılaştırılabilmesine olanak tanıyan değişkenlerdir.

Bu çalışmada OBSO öbikleme yönteminin başarımını incelemek amacıyla  $V$  geçerlilik indeksi (Omran ve diğ., 2005, Merwe ve Engelbrecht, 2003) ve Dunn indeksi (Bezdek ve Pal, 1995) hesaplanmıştır. OBSO yöntemi üç farklı test veri kümesi üzerinde K-Ortalamalar, Bulanık C-Ortalamalar (BCO), Gustafson-Kessel (GK) ve PSO Melez Öbikleme (PSO) (Merwe ve Engelbrecht, 2003) yöntemleri ile karşılaştırılmıştır.

OBSO yönteminin ve PSO (Merwe ve Engelbrecht, 2003) yönteminin uygulamaları Microsoft Visual Studio .NET 2008 platformunda Visual Basic 2008 dili ile yazılmış olan bir program aracılığı ile gerçekleştirilmiştir. Şekil 4.1' de bu programın örnek bir ekran görüntüsü yer almaktadır. Program, ilk olarak girdileri okumakta, ardından sürünün parametreleri belirlenerek, yineleme ve koşum sayıları girilmektedir. Model çalıştırıldığında çıktı olarak her bir bireyin son yineleme sonucundaki konum vektörleri ve en iyi yineleme bilgileri verilmektedir.

BCO ve GK algoritmaları MATLAB üzerinde geliştirilmiş olan "Fuzzy Clustering and Data Analysis Toolbox For Use with MATLAB" (Balasko ve diğ., 2005) kullanılarak çalıştırılmıştır. BCO - COG ve BCO - CP melez yöntemleri ise MATLAB ve ardından sonuçların Microsoft Excel yardımı ile sırasıyla ağırlık merkezinin bulunması ve konveks programlama ile optimizasyonunun yapılması şeklinde ele alınmıştır.



Şekil 4.1: OBSO ve PSO algoritmaları için geliştirilen programın örnek ekran görüntüsü

Test verisi olarak kullanılan veri kümelerinden birincisi Iris veri kümesi, ikinci veri kümesi İstanbul'un Avrupa yakasındaki 51 ilçe/beldenin koordinat bilgileri ve üçüncü veri kümesi ise 400 rastgele belirlenmiş noktadan oluşan bir rastgele kümedir.

### 5.1. V İndeksi

OBSO yönteminin öbekleme performansını değerlendirmek amacıyla formül 4.1' de verilen nicelendirme hatası indeksi (V) kullanılmıştır (Omran ve diğ., 2005, Merwe ve Engelbrecht, 2003).

$$V = \frac{Intra}{Inter}$$

$$Intra = \frac{\sum_{j=1}^C [\sum_{z \in C_j} dist(z, m_j) / |C_j|]}{N_C} \quad (4.1)$$

$$Inter = \min_{i \neq j} \{dist(m_i, m_j)\}$$

*Intra*, öbek bazında, her bir veri elemanının atandığı öbek merkezine olan uzaklıkları toplamının öbekteki veri elemanı sayısına bölünmesi ile elde edilen değerlerin ortalamasıdır. *Inter* ise birbirine en yakın iki öbek merkezi arasındaki mesafeyi gösterir. Formül 4.1’de yer alan *dist* fonksiyonu iki vektör arasındaki mesafeyi hesaplar ve formül 4.3’de gösterilmiştir.  $|C_j|$ ,  $j$ . öbekteki veri elemanı sayısını ifade etmektedir.  $m_j$  ise  $j$ . öbek için tespit edilmiş olan öbek merkezidir (formül 4.2).

$$m = \forall x \in S_F \quad (4.2)$$

Herhangi bir birey( $b_i$ ) için amaç fonksiyonu değeri hesaplanırken, ilk olarak bu bireye en yakın olan odak birey ( $f$ ) tespit edilir. Ardından öbeklenmek istenen her bir veri elemanının ( $z$ ),  $f$  dışındaki tüm diğer odak bireyler ve  $b_i$  bireyi ile arasındaki mesafeler hesaplanır (formül 4.3). Her bir veri elemanı, hesaplanan mesafeler içerisinde en yakın olan odağın yada bireyin temsil ettiği odağın öbeğine atanır.

$$dist = \begin{cases} f \neq f' \Rightarrow dist(f, z) \\ f = f' \Rightarrow dist(b, z) \end{cases} \quad (4.3)$$

## 5.2. Dunn İndeksi

Dunn indeksi öbekler arası en küçük uzaklık ile öbek içi en büyük uzaklığın oranı olarak ifade edilir (Bezdek ve Pal, 1995). Herhangi iki  $X_i$  ve  $X_j$  öbekleri arasındaki en küçük mesafe, farklı öbeklerdeki herhangi bir veri elemanı çiftinden birbirine en yakın olan iki tanesinin arasındaki mesafe olarak formül 4.4’deki gibi hesaplanır.

$$dist(X_i, X_j) = \min_{\substack{z_i \in X_i \\ z_j \in X_j}} \{d(z_i, z_j)\} \quad (4.4)$$

Öbek içi en büyük mesafe ise bir öbekteki veri elemanı çiftlerinden birbirine en uzak iki tanesi arasındaki mesafedir ve formül 4.5’de gösterilmiştir.

$$diam(X_i) = \max_{z_i, z_j \in X_i} \{d(z_i, z_j)\} \quad (4.5)$$

Dunn indeksi tüm öbeklerin birbirleri ile karşılaştırılarak en küçük  $\frac{dist(X_i, X_j)}{\max\{diam(X)\}}$  değerinin bulunması ile hesaplanır (formül 4.6).

$$Dunn = \min_{1 \leq i \leq N_c} \left\{ \min_{\substack{1 \leq j \leq N_c \\ j \neq i}} \left\{ \frac{dist(X_i, X_j)}{\max_{1 \leq k \leq N_c} \{diam(X_k)\}} \right\} \right\} \quad (4.6)$$

### 5.3. Test Veri Kümelerinde Sonuçlar

K-Ortalamlar, BCO, GK, PSO Melez ve OBSO yöntemleri üç farklı veri kümesinde çeşitli öbek sayıları için çalıştırılarak sonuçlar incelenmiştir. Her bir yöntem 30'ar kez çalıştırılmış ve tüm koşumların ortalama değeri ile standart sapması verilmiştir. BCO ve GK algoritmaları için bulanıklaştırma faktörü Zalik' in (2006) çalışmasında belirtilen  $m = 1,7$  değeri; OBSO algoritması için ise  $\chi = 0,7298$  ve  $c_1 = c_2 = 2,05$  değerleri kabul edilmiştir. OBSO yönteminde odak bireylerin de hızlarının tanımlandığı model kullanılmıştır.

#### 5.3.1. I. Test Veri Kümesi (IRIS Veri Kümesi)

Iris veri kümesi her biri 50 örnekten oluşan 3 sınıfı barındırır. Bu sınıfların her biri farklı bir Iris bitki türünü temsil etmektedir. Sınıflardan biri diğerlerinden kolaylıkla ayırt edilebilirken diğer iki sınıfı birbirinden ayırt etmek güçtür. Örneklere ait ölçülen özellikler şunlardır;

- Çanak uzunluğu
- Çanak kalınlığı
- Taç yaprağı uzunluğu
- Taç yaprağı kalınlığı

Bu veri kümesinde öbekleme analizi ile toplam 150 örneğin ait oldukları sınıflar tespit edilmeye çalışılır. Öbekleme analizi ile elde edilen 3 öbeğin merkez değerleri ile her bir örneğe ait ölçülen değerler karşılaştırılır. Her örnek kendisine en yakın öbeğe atanır ve örneklerin gerçek sınıfları ile öbekleme analizi neticesinde atandıkları sınıf karşılaştırılır. Formül 4.7'deki hata değeri öbekleme analizi yönteminin başarımını ölçmek için kullanılır (Balasko ve diğ., 2005).

$$\text{Hata Oranı} = \frac{N_{\text{hatalı}}}{N} \times 100 \quad (4.7)$$

Formül 4.7’de  $N$  toplam örnek adedi olan 150,  $N_{hatalı}$  ise öbikleme analizi neticesinde hatalı atanan örnek sayısını temsil etmektedir.

Iris verisi üzerinde aşağıdaki öbikleme analizi yöntemleri kullanılmıştır;

- K-Ortalamlar
- Bulanık C-Ortalamlar
- Gustafson-Kessel
- OBSO

OBSO yönteminde amaç fonksiyonu olarak  $V$  indeksi (formül 4.1) kullanılmıştır. Bu amaç fonksiyonu diğer öbikleme yöntemleri ile aynı temel prensip olan öbek içindeki farklılıkların minimum ve öbekler arası farklılıkların maksimum olduğu duruma ulaşmayı hedeflediği için karşılaştırma amacına uygun bir fonksiyondur. OBSO yönteminin başarımlarını test etmek için 30’ ar koşul yapılmıştır.

Tablo 4.1: Iris veri kümesinde öbikleme yöntemlerinin tahmin hatasına göre başarımları  
(Küçük değerler daha iyi)

Yöntem	Yanlış Tahmin Sayısı	Hata Oranı
K-Ortalamlar	16	10,66%
BCO	16	10,66%
GK	7	4,66%
PSO Melez En İyi	16	10,66%
PSO Melez Ortalama	16	10,66%
OBSO En İyi	16	10,66%
OBSO Ortalama	17,43	11,62%



Tablo 4.2’de ise karşılaştırılan öbikleme analizi yöntemlerinin Dunn indeksine göre aldıkları değerler görülmektedir.

Tablo 4.2: Iris veri kümesinde öbikleme yöntemlerinin Dunn indeksine göre başarımı (Büyük değerler daha iyi)

Yöntem	Dunn İndeksi
K-Ortalamlar	0,098
BCO	0,104
GK	0,048
PSO Melez	0,106
OBSO Ortalama	0,112

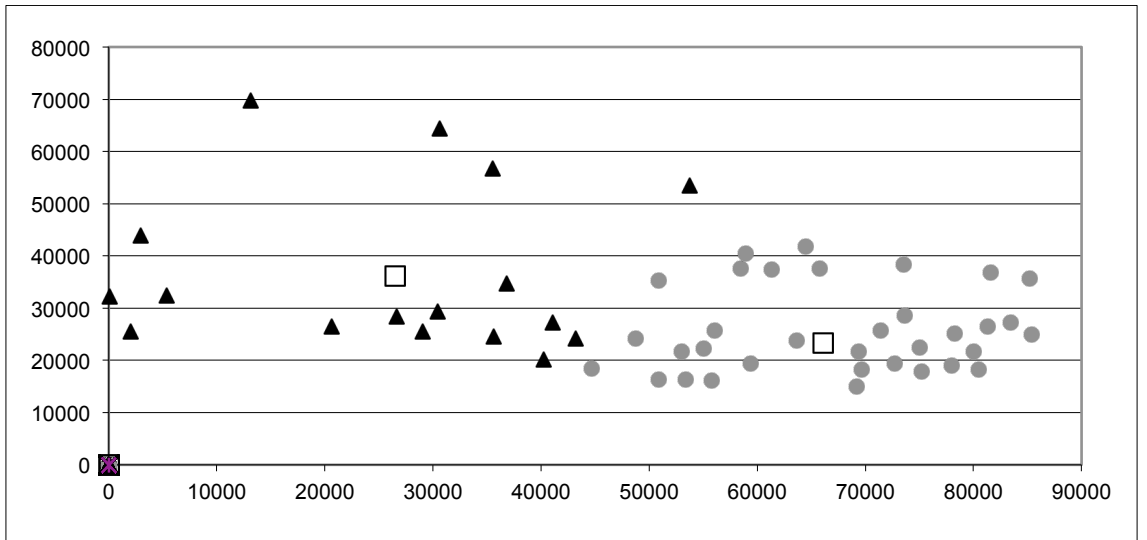
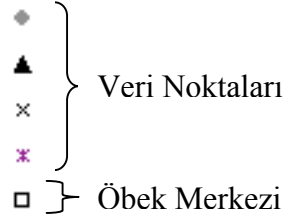
Merwe ve diğ. (2003) Iris veri kümesini geliştirdikleri parçacık sürü optimizasyonu tabanlı öbikleme algoritmasını test etmek için kullanmıştır. Bu çalışmada, K-Ortalamlar yöntemi, sürü zekası tabanlı öbikleme analizi yöntemi ve öbek merkezlerinin başlangıç durumunu K-Ortalamlar analizinin sonuçlarından alan melez sürü zekası tabanlı öbikleme analizi yöntemi niceleme hatası (formül 4.1)’ e göre karşılaştırılmıştır. Merwe ve diğ. (2003)’ nin ulaştıkları sonuçlar ile OBSO algoritmasının sonuçları tablo 4.3’ de gösterilmiştir. Tablodaki değerler 30 koşum sonucunda elde edilen sonuçların ortalama ve standart sapmalarını yansıtmaktadır.

Tablo 4.3: Iris veri kümesinde öbikleme yöntemlerinin  $V$  indeksine göre başarımı (Küçük değerler daha iyi)

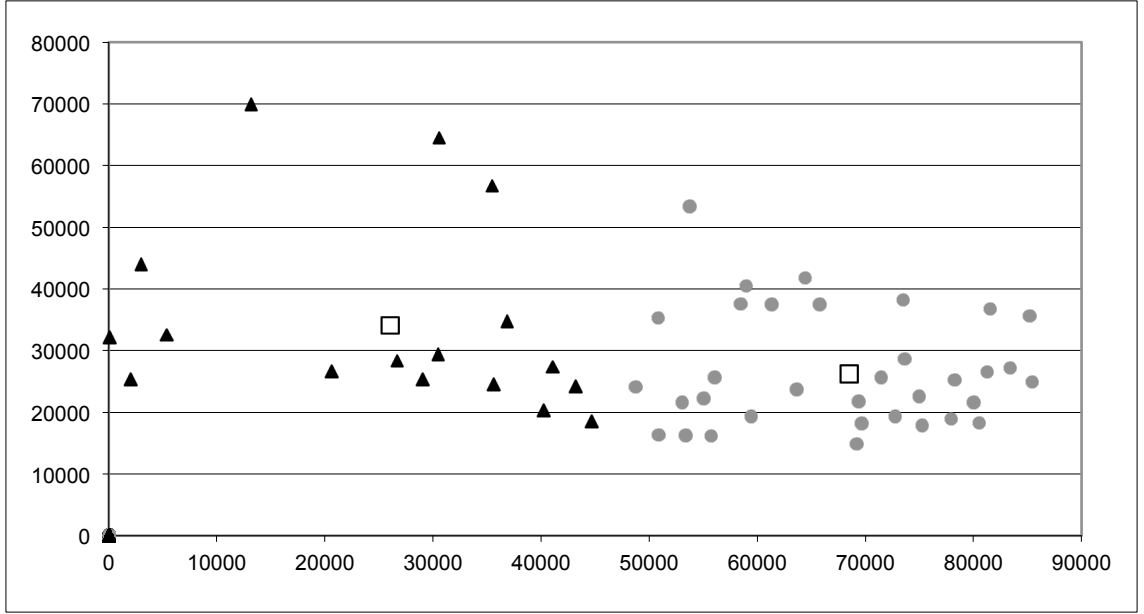
Yöntem	$V$ İndeksi
K-Ortalamlar (Merwe ve diğ, 2003)	0,649 $\pm$ 0,146
PSO (Merwe ve diğ, 2003)	0,774 $\pm$ 0,094
PSO Melez (Merwe ve diğ, 2003)	0,633 $\pm$ 0,143
OBSO	0,623 $\pm$ 0,023

### 5.3.2. II. Test Veri Kümesi

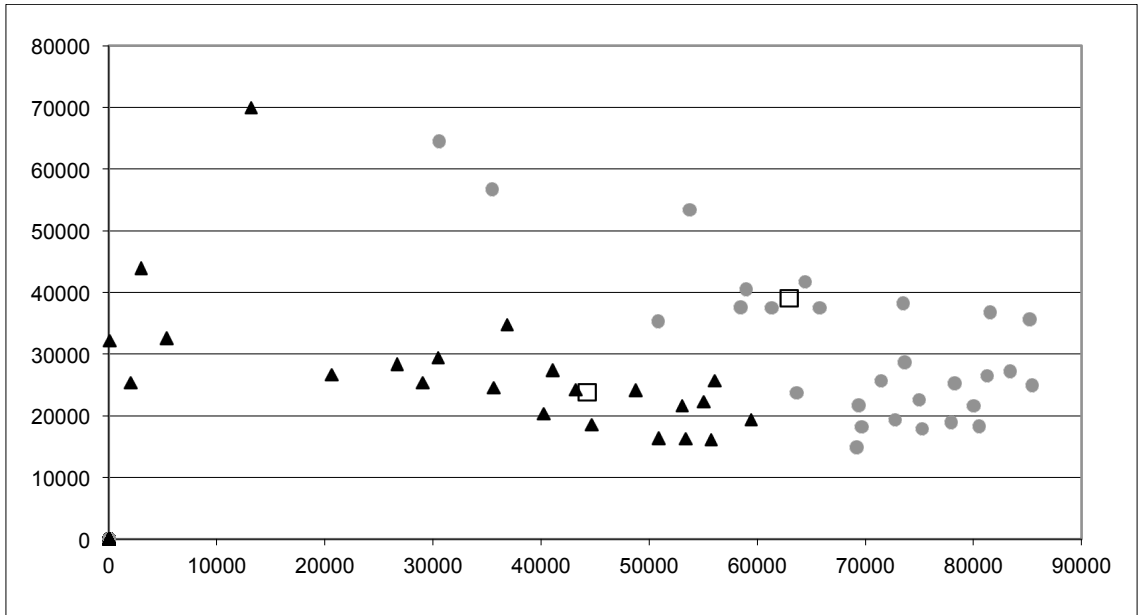
II. test veri kümesi İstanbul Avrupa yakasındaki 51 adet ilçe/belde merkezinin koordinatlarından oluşmaktadır. Karşılaştırılan öbekleme analizi yöntemleri neticesinde öbek yapıları takip eden şekillerde görülmektedir. Şekillerde kullanılan simgeler hemen aşağıda tanımlanmıştır.



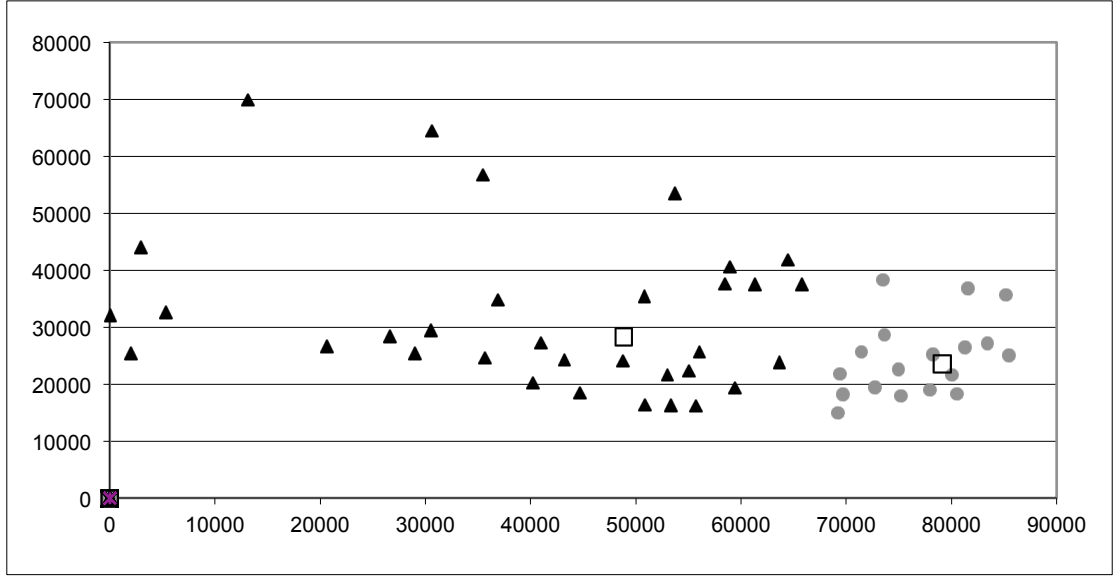
Şekil 4.2: 51 noktalı veri kümesinin K-Ortalamlar – 2 öbek sonuçları



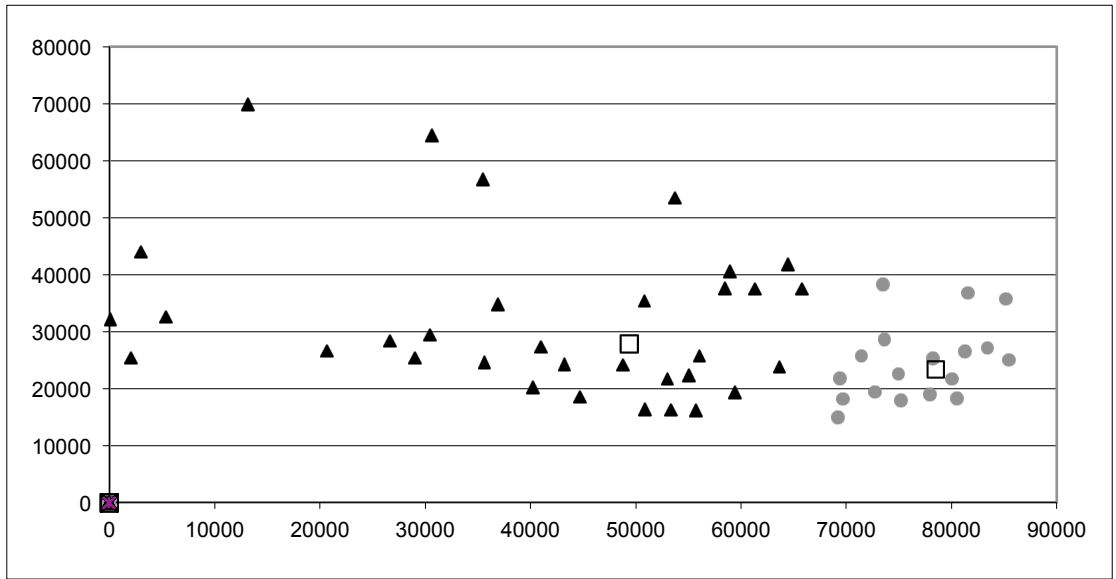
Şekil 4.3: 51 noktalı veri kümesinin BCO – 2 öbek sonuçları



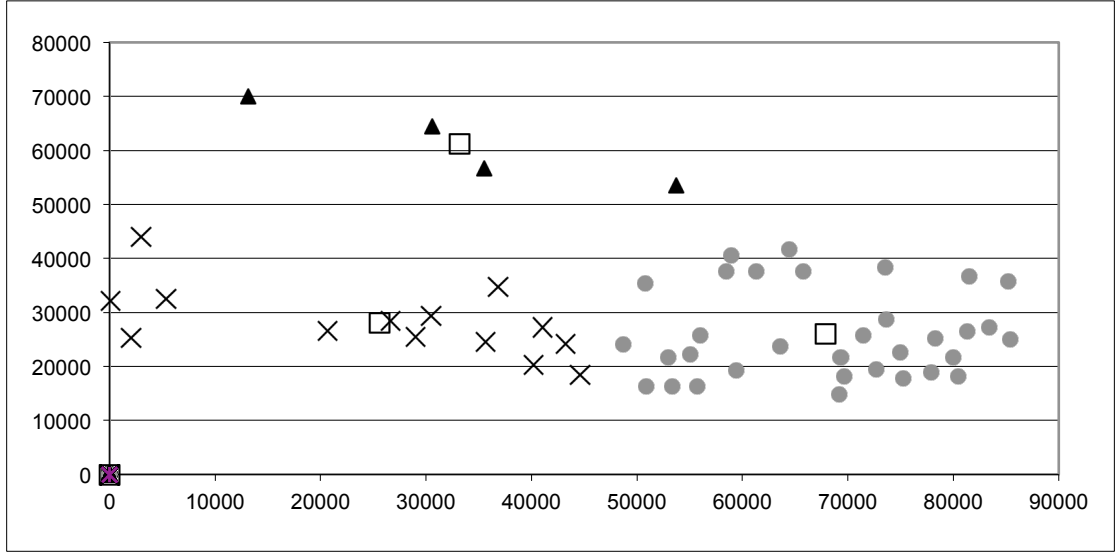
Şekil 4.4: 51 noktalı veri kümesinin GK – 2 öbek sonuçları



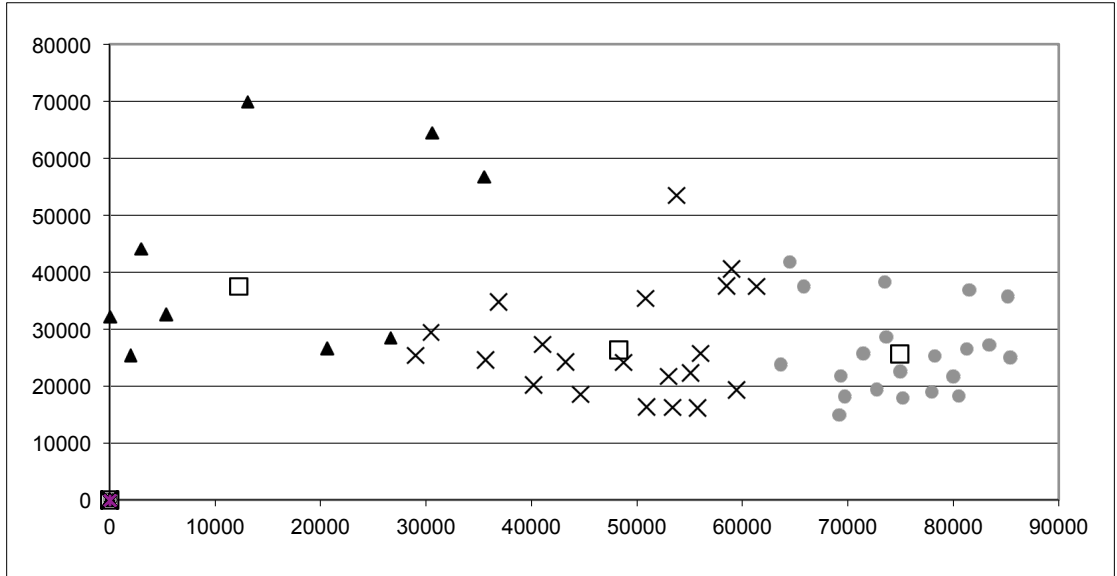
Şekil 4.5: 51 noktalı veri kümesinin PSO – 2 öbek sonuçları



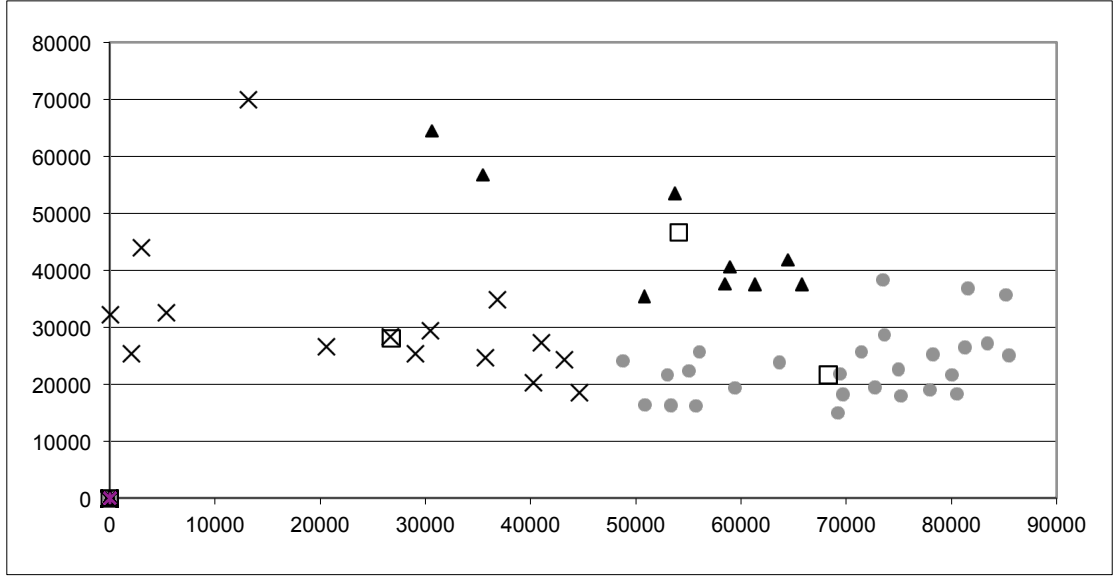
Şekil 4.6: 51 noktalı veri kümesinin OBSO – 2 öbek sonuçları



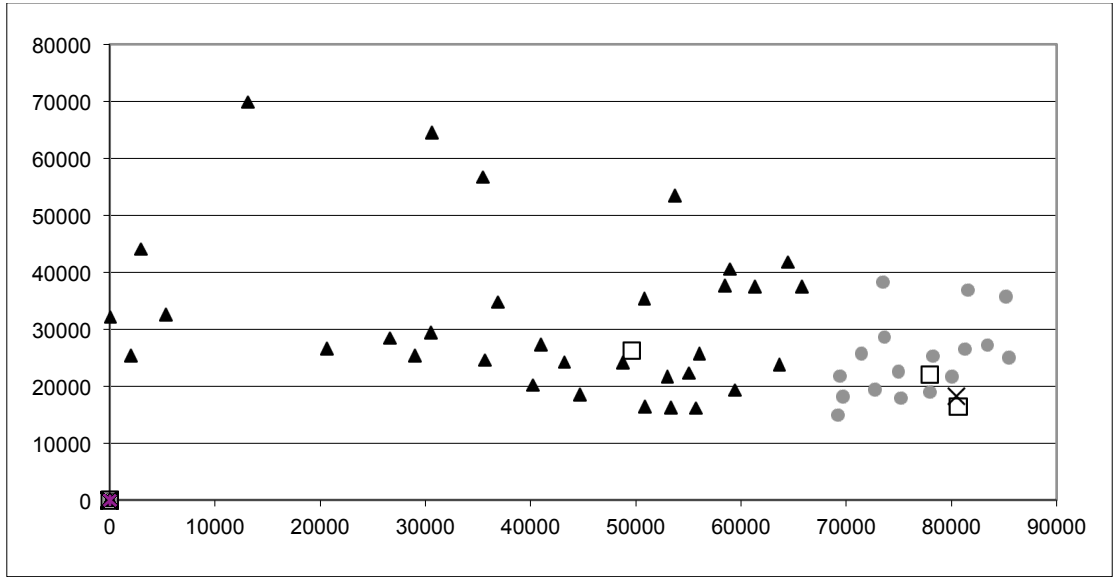
Şekil 4.7: 51 noktalı veri kümesinin K-Ortalamalar – 3 öbek sonuçları



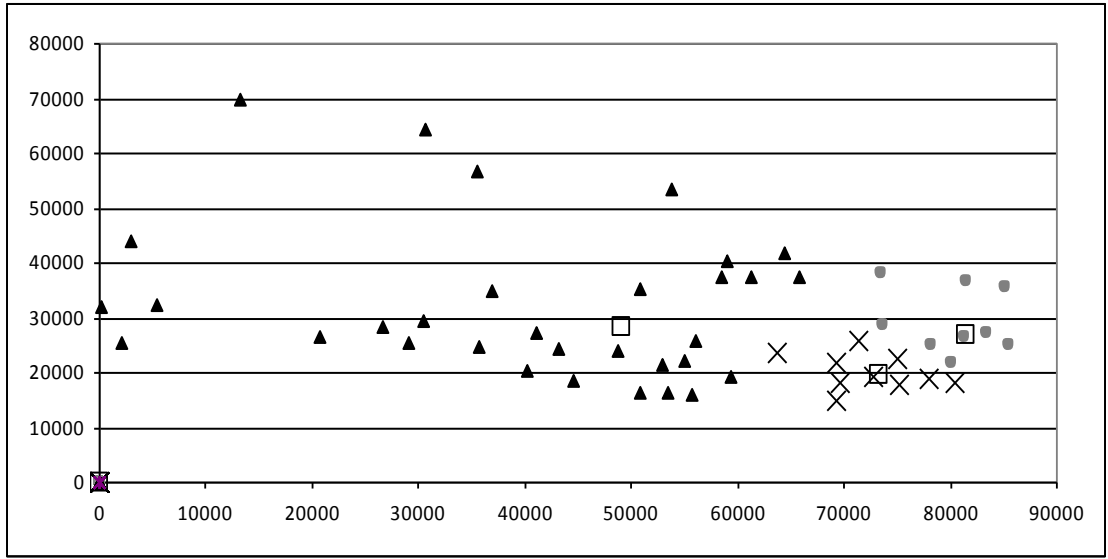
Şekil 4.8: 51 noktalı veri kümesinin BCO – 3 öbek sonuçları



Şekil 4.9: 51 noktalı veri kümesinin GK – 3 öbek sonuçları



Şekil 4.10: 51 noktalı veri kümesinin PSO – 3 öbek sonuçları



Şekil 4.11: 51 noktalı veri kümesinin OBSO – 3 öbek sonuçları

Tablo 4.4’de 51 noktalı veri kümesinde elde edilen  $V$  indeksi değerlerine ve tablo 4.5’de ise Dunn indeksi değerlerine yer verilmiştir.

Tablo 4.4: 51 noktalı veri kümesinde öbekleme yöntemlerinin  $V$  indeksine göre başarımı (Küçük değerler daha iyi)

Yöntem	Öbek Sayısı	
	2	3
K-Ortalamalar	0,382	0,404
BCO	0,378	0,507
GK	0,768	0,491
PSO Melez En İyi (Merwe ve diğ, 2003)	0,465	0,804
PSO Melez Ortalama (Merwe ve diğ, 2003)	$0,494 \pm 0,006$	$2,27 \pm 0,848$
OBSO En İyi	0,467	0,281
OBSO Ortalama	$0,483 \pm 0,006$	$0,442 \pm 0,113$

Tablo 4.5: 51 noktalı veri kümesinde öbekleme yöntemlerinin Dunn indeksine göre başarımı (Büyük değerler daha iyi)

Yöntem	Öbek Sayısı	
	2	3
K-Ortalamlar	0,067	0,113
BCO	0,092	0,078
GK	0,089	0,091
PSO Melez En İyi (Merwe ve diğ, 2003)	0,089	0,088
PSO Melez Ortalama (Merwe ve diğ, 2003)	0,078 ± 0,012	0,045 ± 0,01
OBSO En İyi	0,089	0,113
OBSO Ortalama	0,089 ± 0	0,093 ± 0,013

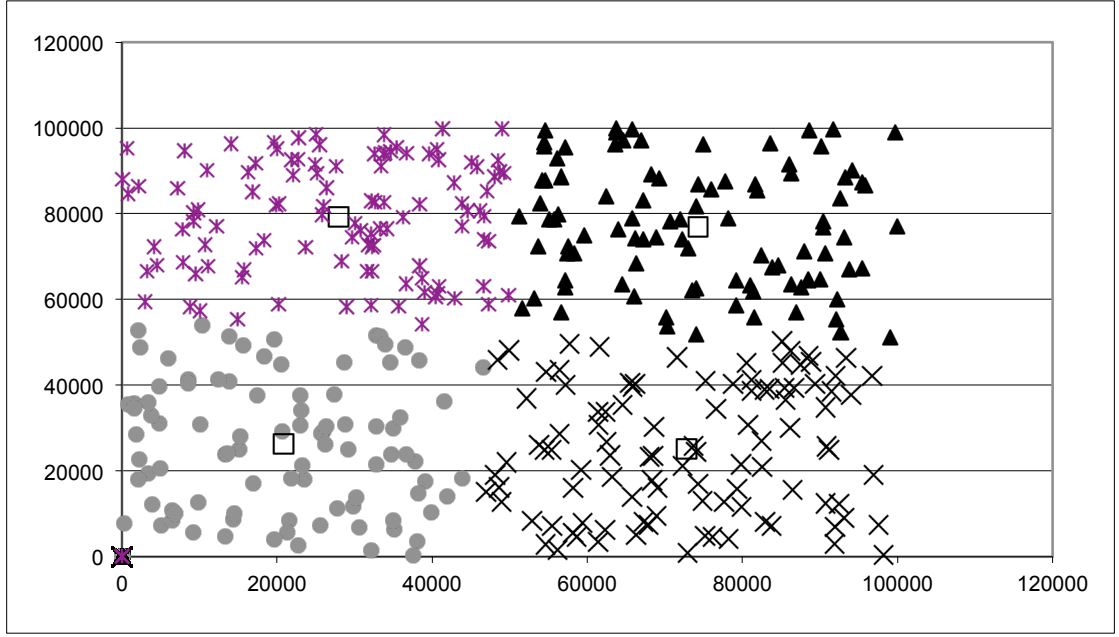
51 noktalı veri kümesi için yapılan analiz neticesinde hemen aşağıdaki sonuçlara ulaşılmıştır;

- 2 Öbekli çözümlerde, ortalamada ve en iyi çözümlerde, OBSO yöntemi Dunn indeksine göre K-Ortalamlar ve PSO Melez yöntemlerinden daha iyi, GK ile aynı ve BCO yönteminden daha kötü sonuçlar sergilemiştir.  $V$  indeksine göre ise PSO Melez ve GK algoritmalarından daha iyi, diğer yöntemlerden daha kötü sonuçlar ortaya çıkmıştır.
- 3 Öbekli çözümlerde, ortalamada, OBSO yöntemi K-ortalamlar dışındaki tüm yöntemlerden daha iyi bir başarıml göstermiştir. En iyi çözümlerde ise K-ortalamlar ile aynı ve diğer tüm yöntemlerden daha iyi sonuçlar ortaya çıkmıştır.
- 3 Öbekli en iyi OBSO çözümü diğer tüm yöntemlerden daha iyi sonuç vermiştir.

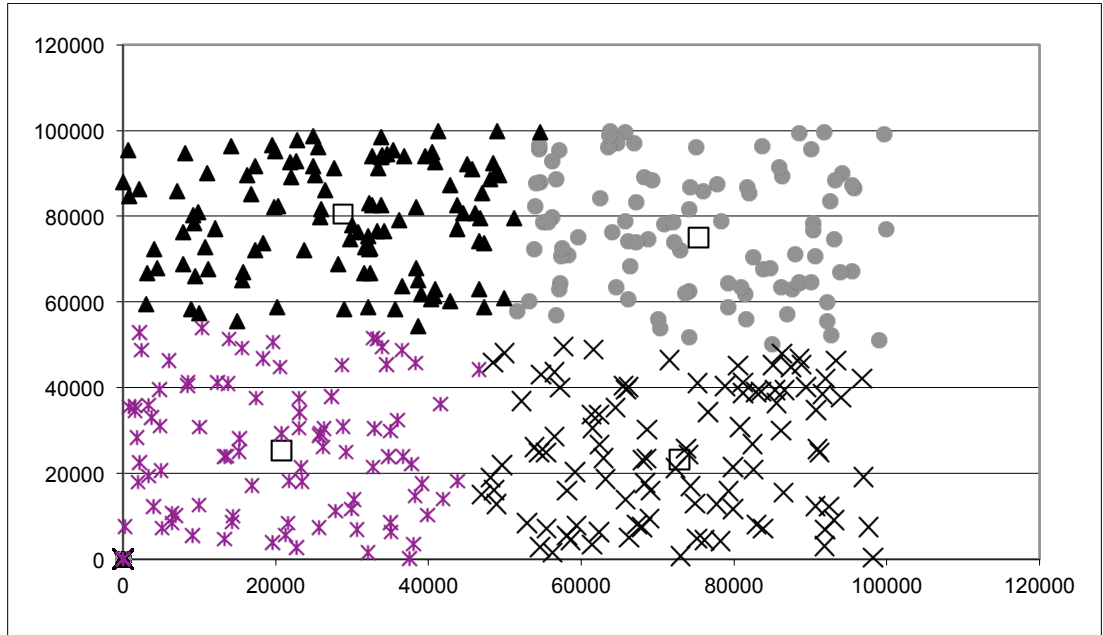
### 5.3.3. III. Test Veri Kümesi

III. test veri kümesi 400 adet rastgele belirlenmiş noktadan oluşmaktadır. Karşılaştırılan öbekleme analizi yöntemleri neticesinde öbek yapıları takip eden şeklerde görülmektedir.

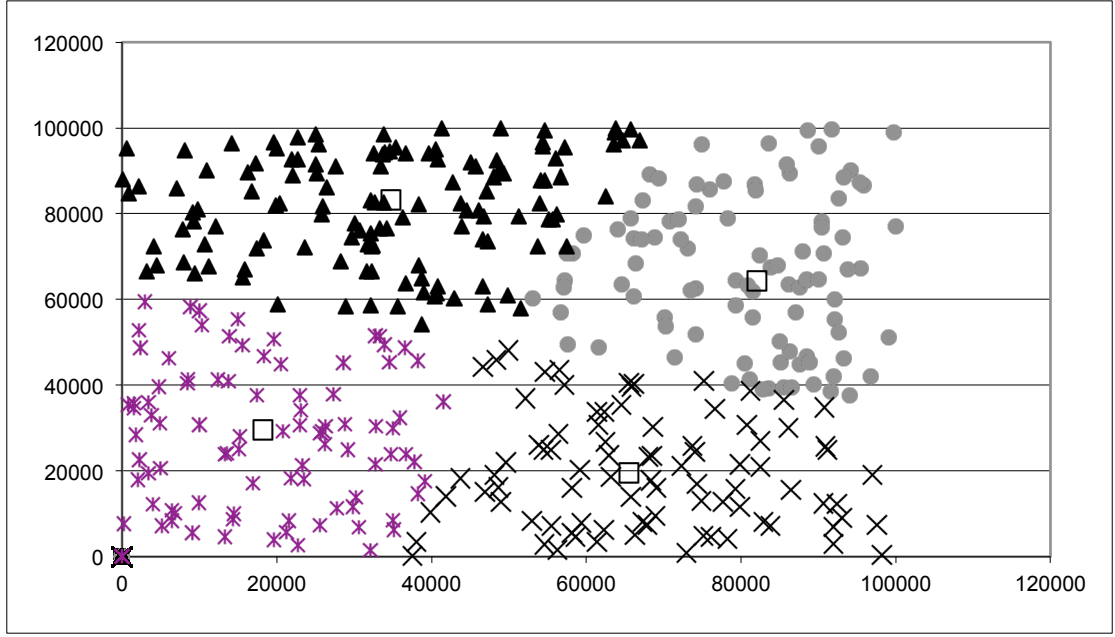




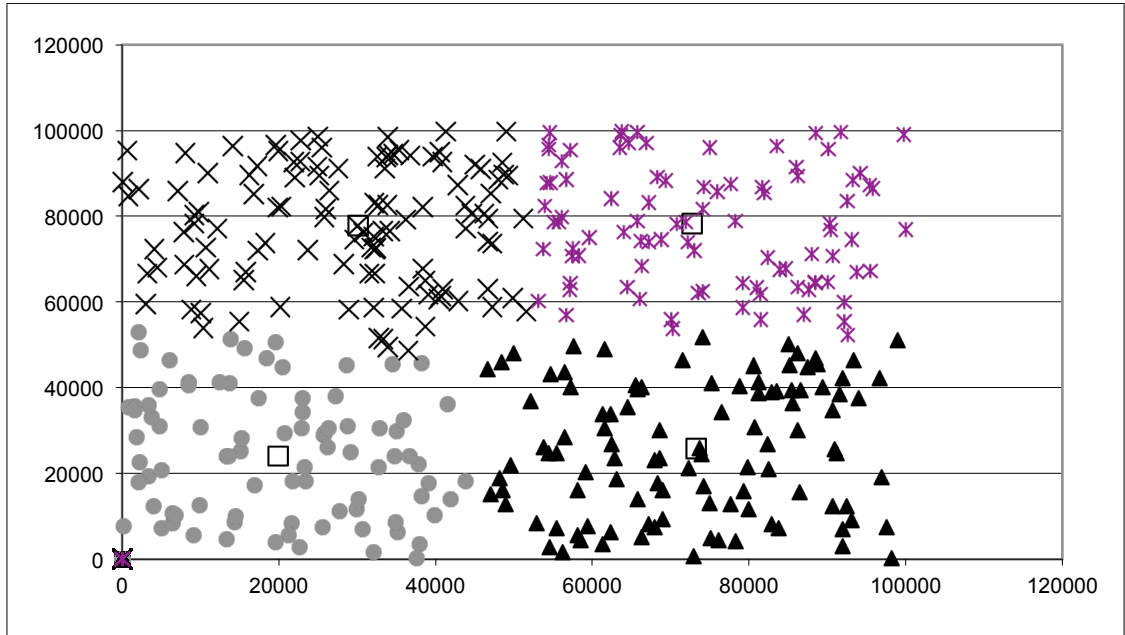
Şekil 4.12: 400 noktalı veri kümesinin K-Ortalamlar – 4 öbek sonuçları



Şekil 4.13: 400 noktalı veri kümesinin BCO – 4 öbek sonuçları



Şekil 4.14: 400 noktalı veri kümesinin GK – 4 öbek sonuçları



Şekil 4.15: 400 noktalı veri kümesinin OBSO – 4 öbek sonuçları

Tablo 4.6: 400 noktalı veri kümesinde öbikleme yöntemlerinin  $V$  indeksine göre başarımı (Küçük değerler daha iyi)

Yöntem	Öbek Sayısı
	4
K-Ortalamalar	0,399
BCO	0,395
GK	0,395
PSO Melez En İyi (Merwe ve diğ, 2003)	0,394
PSO Melez Ortalama (Merwe ve diğ, 2003)	2,469 ± 1,65
OBSO En İyi	0,398
OBSO Ortalama	0,421 ± 0,01

Tablo 4.7: 400 noktalı veri kümesinde öbikleme yöntemlerinin Dunn indeksine göre başarımı (Büyük değerler daha iyi)

Yöntem	Öbek Sayısı
	4
K-Ortalamalar	0,035
BCO	0,035
GK	0,023
PSO Melez En İyi (Merwe ve diğ, 2003)	0,035
PSO Melez Ortalama (Merwe ve diğ, 2003)	0,019 ± 0,008
OBSO En İyi	0,041
OBSO Ortalama	0,032 ± 0,005

400 noktalı veri kümesi için yapılan analiz neticesinde hemen aşağıdaki sonuçlara ulaşılmıştır;

- OBSO yöntemi, ortalamada Dunn indeksine göre GK ve PSO Melez yöntemlerinden daha iyi, K-Ortalamalar ve BCO yöntemlerinden daha kötü başarımlar sergilemiş,  $V$  indeksine göre PSO Melez yönteminden daha iyi, diğer yöntemlerden daha kötü sonuç sergilemiştir.
- OBSO yöntemi, en iyi çözümlerde Dunn indeksine göre tüm diğer yöntemlerden daha iyi sonuç vermiş,  $V$  indeksine göre K-ortalamlar yönteminden daha iyi, diğer yöntemlerden ise kötü ama çok yakın başarımlar göstermiştir.

#### **5.3.4. Tesis Yeri Seçimi Probleminde OBSO Yöntemini uygulanması**

Öbekleme analizi, talep noktalarının coğrafi düzlemdeki koordinatlarının bilindiği problemlerde üretim ve/veya dağıtım tesisleri gibi tedarik zinciri unsurlarının ulaştırma maliyetlerini en aza indirgeyecek şekilde konumlandırılması için kullanılan bir yöntemdir. Tesis yeri seçimi probleminin bir öbekleme problemi olarak ele alınmasının sağladığı bir diğer fayda ise analiz neticesinde tesis – talep noktası eşleşmelerinin doğrudan elde edilmesidir.

Bu çalışmada geliştirilen OBSO yöntemi, amaç fonksiyonu kurulacak tesisler ile talep noktaları arasındaki toplam ulaştırma maliyetini en küçükleyecek bir model kurularak İstanbul'un Avrupa yakasındaki 51 ilçe ve beldeye, bu ilçe ve beldelerin önceden bilinen talep miktarları kadar dağıtım yapacak bir tesis yeri seçimi probleminin çözümü için kullanılmıştır. Aynı problem BCO, BCO ve Ağırlık Merkezi, BCO ve konveks programlama, GK ve PSO Melez yöntemleri ile de çözümlenerek karşılaştırmalı sonuçlar verilmiştir.

Model 51 adet talep noktasından oluşmaktadır. Bu noktaların koordinat ve talep bilgileri tablo 4.8'de görülmektedir.

Tablo 4.8: Tesis yeri seçimi problemi için kullanılan 51 noktanın koordinat ve talep verisi

İlçe Adı	X	Y	Talep
ARNAVUTKÖY	61311	37489	35867
AVCILAR	59425	19301	101632
BAĞCILAR	69353	21702	141768
BAHÇEKÖY	81557	36768	11779
BAHÇELİEVLER	69676	18161	72952
BAHÇEŞEHİR	56037	25653	4100
BAKIRKÖY	69205	14856	103630
BAYRAMPAŞA	74990	22558	40438
BEŞİKTAŞ	85415	24968	42640
BEYLİKDÜZÜ	53369	16218	22550
BEYOĞLU	80017	21669	27875
BİNKILIÇ	13149	69854	5831
BOĞAZKÖY	65785	37535	30991
BOLLUCA	64447	41729	13619
BÜYÜKÇAVUŞLU	2992	43953	10000
BÜYÜKÇEKMECE	48753	24148	8600
CELALİYE-KAMİLOBA	35637	24553	2575
ÇANTA	5379	32489	9000
ÇATALCA	36882	34755	152350
ÇİFTLİKKÖY	35493	56689	5025
DEĞİRMENKÖY	100	32143	5207
DURUSU	53715	53406	5025
EMİNÖNÜ	80478	18277	16316
ESENLER	71423	25720	70179
ESENYURT	55064	22292	66650
EYÜP	78228	25211	70000
FATİH	77942	18983	83658
GAZİOSMANPAŞA	73635	28639	75000
GÖKTÜRK	73496	38256	5025
GÜMÜŞYAKA	2010	25393	4300
GÜNGÖREN	72719	19399	94639
GÜRPINAR	50879	16311	21818
HADIMKÖY	50832	35355	12800
HARAÇCI	58451	37627	10000
KAĞITHANE	81279	26484	30266
KARACAKÖY	30599	64447	2525
KAVAKLI	26678	28396	2575
KIRAÇ	53000	21610	28837
KUMBURGAZ	40220	20231	7593
KÜÇÜKÇEKMECE	63628	23752	100000
MİMARŞINAN	44648	18479	24205
MURATBEY	41033	27290	10283
ORTAKÖY	30471	29364	5025
SARIYER	85197	35632	69914
SELİMPAŞA	29030	25401	10000
SİLİVRİ	20636	26599	144424
ŞİŞLİ	83431	27185	40894
TAŞOLUK	58959	40531	28593
TEPECİK	43218	24194	16450
YAKUPLU	55721	16172	16450
ZEYTİNBURNU	75229	17894	58609

Öbekleme analizi ile söz konusu 51 noktaya en düşük ulaştırma maliyeti ile hizmet sağlayacak olan 2 veya 3 olası tesisin konumları ve tesis-müşteri eşleşmeleri ile bu eşleşmeler neticesinde ortaya çıkan ulaştırma maliyetleri hesaplanacaktır. Tesislerin kapasite kısıtının olmadığı varsayılmıştır. Dolayısıyla belirlenen tesislerin kendi öbeklerindeki tüm talebi karşılayacağı kabul edilmiştir. Ulaştırma maliyetlerinin hesaplanması için formül 4.8'deki yaklaşım uygulanmıştır. Maliyet faktörü, tüm talep noktalarının ( $z$ ) ait oldukları öbek merkezinde bulunması varsayılan tesis ile aralarındaki mesafe ile talep miktarının ( $w_i$ ) çarpımı olarak ifade edilmiştir.

$$TM = \sum_{i=1}^c \sum_{z \in S_i} w_i \times dist(z_i, f_i) \quad (4.8)$$

Tablo 4.9’ da çeşitli öbek sayıları için öbekteleme analizi neticesinde ortaya çıkan toplam maliyetler ve tablo 4.10 ve 4.11’ de ise bu yöntemlerin hesapladığı öbek merkezlerinin koordinatları görülmektedir.

Tablo 4.9: Tesis yeri seçimi probleminde hesaplanan maliyet parametresi değerleri

Yöntem	Öbek Sayısı	
	2	3
K-Ortalamlar	23.962.248	21.732.452
BCO	22.384.575	19.878.228
BCO & COG	20.268.809	17.008.644
BCO & CP	20.247.401	16.886.653
GK	36.005.916	19.707.733
PSO Melez (Merwe ve diğ, 2003)	24.539.308 ± 2.282	25.596.249 ± 3.473
OBSO Ort.	20.204.783 ± 1.018	16.222.899 ± 2.615

Tablo 4.10: Tesis yeri problemi için sonuç öbek merkezi koordinatları

Öbek No	K-Ortalamlar		BCO		BCO & COG		BCO & CP	
	X	Y	X	Y	X	Y	X	Y
1	66125,33	26504,56	68506,76	26215,73	70.152	22.035	69977,83	21942,78
2	23353,93	36103,8	26022,34	34063,25	30.682	30.469	31437,11	30917,42

Öbek No	GK		PSO Melez		OBSO	
	X	Y	X	Y	X	Y
1	62997,52	39054,63	49449,45	27344,79	70179,58	21935,06
2	44261,31	23800,29	78225,82	22717,76	34455,31	32618,49

Tablo 4.11: Tesis yeri problemi için sonuç öbek merkezi koordinatları

Öbek No	K-Ortalamlar		BCO		BCO & COG		BCO & CP	
	X	Y	X	Y	X	Y	X	Y
1	67846,39	26002,58	74912,83	25605,59	73.223	22.114	73124,93	22013,61
2	33239	61099	12290,58	37484,67	20.634	26.602	20636,06	26598,84
3	25638,14	28088,57	48245,81	26354,73	51.518	24.153	52142,6	23677,53

Öbek No	GK		PSO Melez		OBSO	
	X	Y	X	Y	X	Y
1	68266,19	21629,27	75833,83	26494,36	74195,01	21758,58
2	54097,7	46582,06	85197,17	35632,01	57445,85	22527,04
3	26813,01	28039,3	45255,65	28886,31	28415,03	30431,37

## 6. TARTIŞMA VE SONUÇ

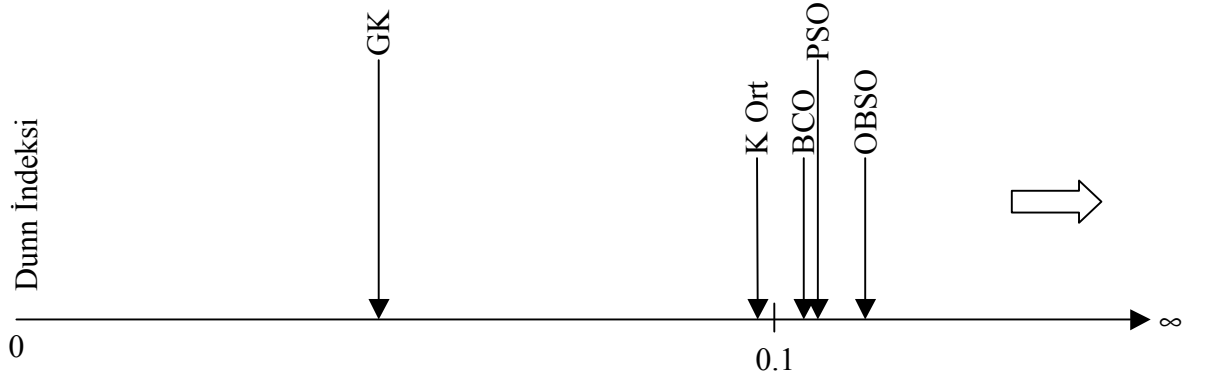
Bu çalışmada sürü zekası tabanlı yeni bir öbekleme analizi yöntemi geliştirilmiş ve bu yöntemin tedarik zinciri yönetiminde tesis yeri seçimi problemlerine uygulanışı incelenmiştir. Odak bireyli sürü zekası optimizasyonu (OBSO) olarak adlandırılan bu yeni yöntem farklı bir komşuluk yapısı ve birden fazla odak bireyin yer aldığı yeni bir sürü zekası algoritmasıdır.

Çalışmanın birinci bölümünde tezin ana hatları ve literatüre sağlanmak istenen katkılara değinilmiştir. Ardından ikinci bölümde tedarik zinciri yönetimi kavramı ve özel olarak tedarik zinciri yönetiminde tesis yeri seçimi problemi tanımlanmış ve çözüm için geliştirilen yöntemler incelenmiştir. Aynı bölümde sürü zekası kavramı ve parçacık sürü optimizasyonu yöntemi anlatılmış, yöntemin temel denklemleri, çalışma prensipleri ve gelişmiş versiyonları sunulmuştur. Üçüncü bölümde, geliştirilen odak bireyli sürü zekası optimizasyonu yöntemi anlatılarak dördüncü bölümde söz konusu yöntemin genel olarak bir öbekleme analizi yöntemi olarak başarımı ve tesis yeri seçimi problemlerinde uygulanması ele alınmış, sonuçlar sunulmuştur.

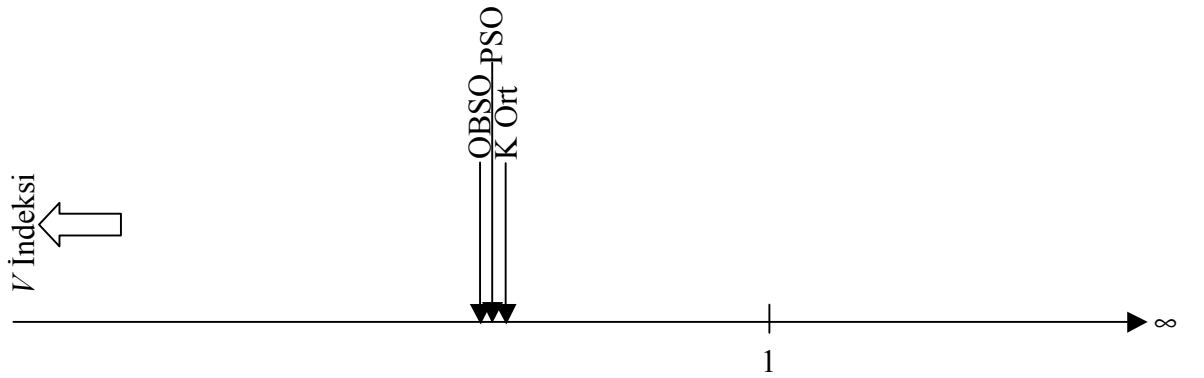
Geliştirilen algoritma, ilk olarak, literatürdeki mevcut öbekleme analizi yöntemleri ile çeşitli performans parametreleri üzerinden karşılaştırılarak yöntemin öbekleme başarımı incelenmiştir. Iris veri kümesi, 51 noktalı İstanbul Avrupa yakası ilçe ve beldeler veri kümesi ve rastgele belirlenmiş 400 noktalı veri kümesi üzerinde yapılan analizler neticesinde Dunn indeksine göre OBSO yönteminin en iyi değerlerinin Iris veri kümesi ve 400 noktalı veri kümesinde, karşılaştırılan diğer öbekleme yöntemlerinin tamamından daha iyi sonuçlar elde ettiği, yine Dunn indeksine göre 51 noktalı veri kümesinde 3 öbekli çözümlemede K-ortalamlar dışındaki öbekleme yöntemlerinden daha iyi sonuçlar elde ettiği, aynı veri kümesinin 2 öbekli çözümlemede ise K-ortalamalardan daha iyi, Bulanık C-ortalamalardan daha kötü ve diğer yöntemlerle aynı



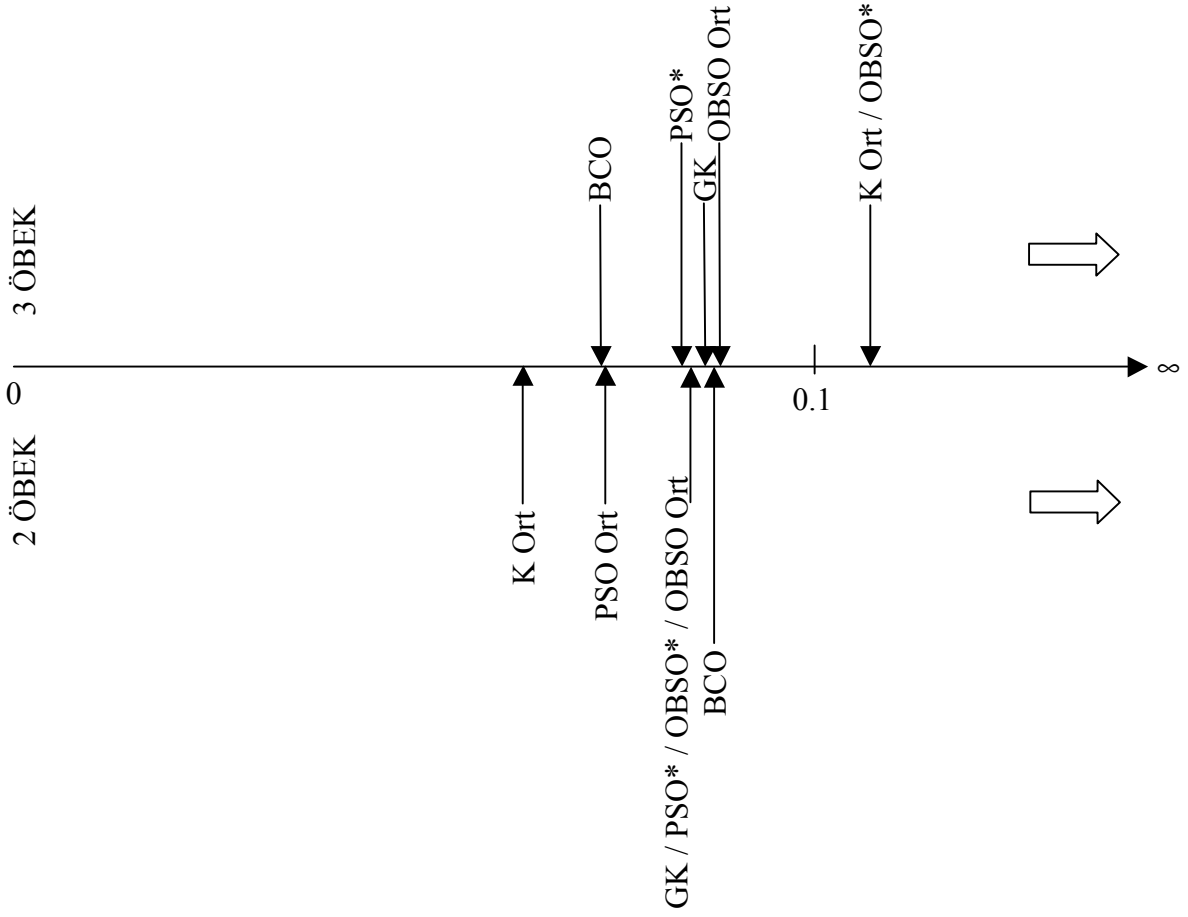
başarım gösterdiği görülmüştür. Şekil 5.1, 5.2, 5.3, 5.4 ve 5.5'de de elde edilen öbekleme analizi sonuçları karşılaştırılmıştır



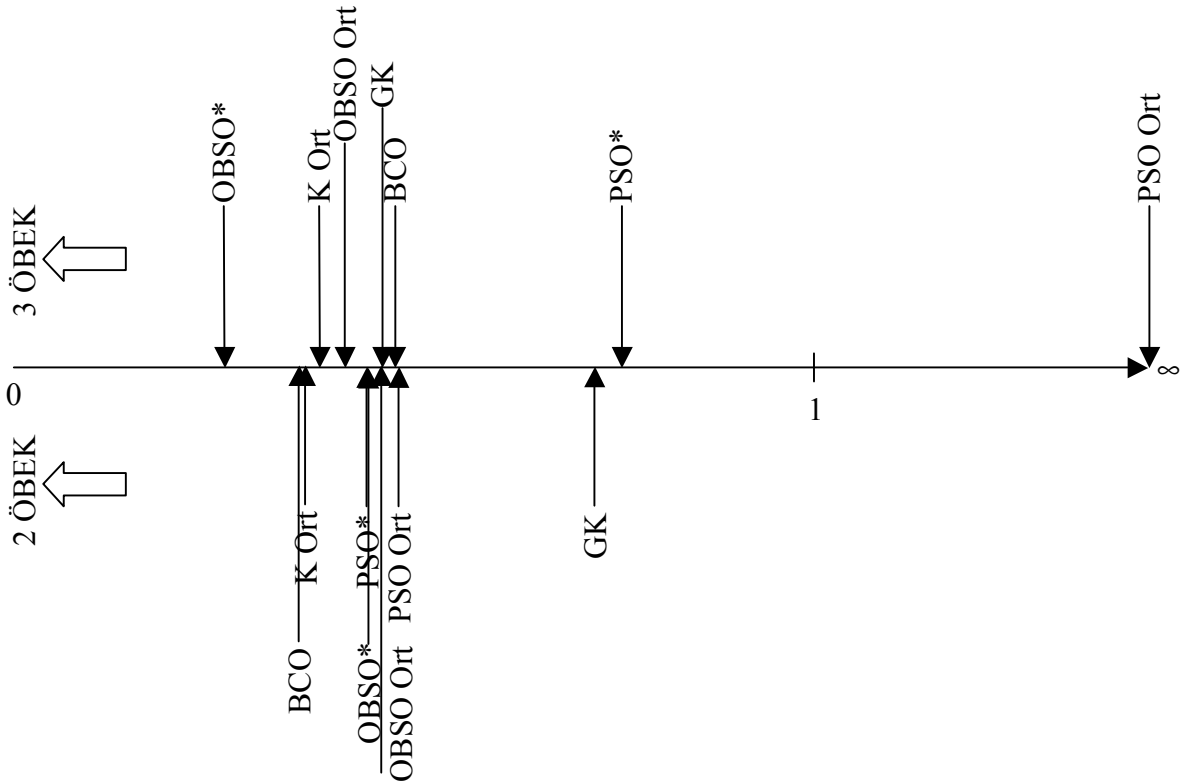
Şekil 5.1: Iris veri kümesinde Dunn indeksine göre başarımlar. (Büyük değerler daha iyi)



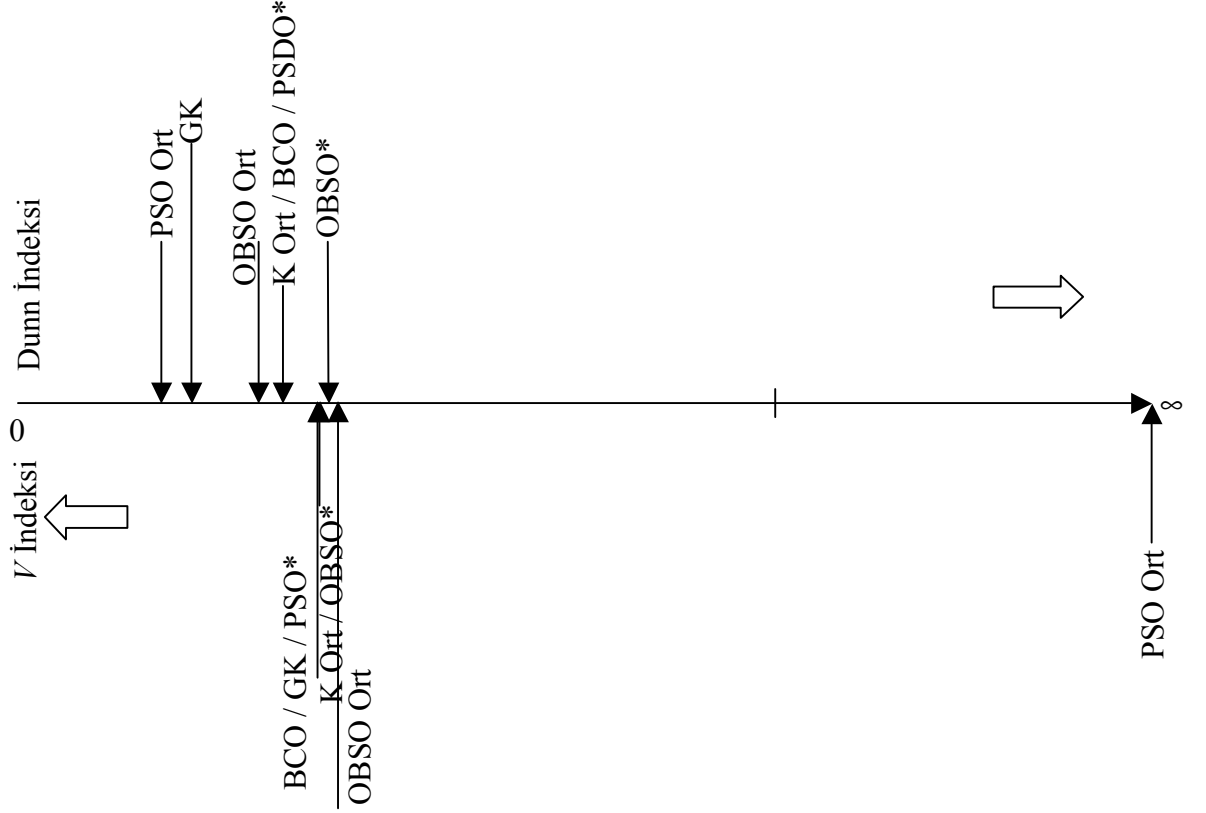
Şekil 5.2: Iris veri kümesinde  $V$  indeksine göre başarımlar. (Küçük değerler daha iyi)



Şekil 5.3: 51 noktalı veri kümesinde Dunn indeksine göre başarımlar. (Büyük değerler daha iyi)



Şekil 5.4: 51 noktalı veri kümesinde V indeksine göre başarımlar. (Küçük değerler daha iyi)



Şekil 5.5: 400 noktalı veri kümesinde başarımların karşılaştırılması.

Tesis yeri seçimi problemlerinde öbikleme analizinin kullanılması coğrafi düzlemde ulaştırma maliyetlerini en küçük kılacak noktaların tespiti ve konumları tespit edilen tesislerin hangi talep noktaları ile eşleşeceğinin belirlenmesini sağlamaktadır. Bu kapsamda, geliştirilen algoritma, örnek bir veri kümesi üzerinde çeşitli öbikleme algoritmaları ve melez algoritmalar ile karşılaştırılarak elde edilen toplam maliyet değerleri karşılaştırılmıştır. Ortaya konan sonuçlar en düşük maliyetli çözümün OBSO yöntemi ile elde edildiğini göstermiştir. OBSO yöntemi ile elde edilen toplam maliyet 2 öbekli çözümlemede K-ortalamlar, Bulanık C-ortalamlar, BCO & Ağırlık merkezi, BCO & Konveks programlama, Gustafson-Kessel ve PSO Melez algoritmalarından sırasıyla %15, %9, %0.3, % 0.2, %43 ve % 17 daha düşük çıkmıştır. 3 öbekli çözümlemede ise söz konusu yöntemlerden, aynı sıra ile %25, %18, %4, %3, %17, %36 daha düşük maliyetlere ulaşılmıştır.

Odak bireyli sürü zekası optimizasyon algoritmasının başarılı bir öbikleme analizi yöntemi olarak ortaya çıkmasının nedeni, arama uzayında hareket eden bireylerin yerel minimumları bulma konusundaki üstünlükleridir. Parçacık sürü optimizasyonu

algoritması yerel ve küresel minimumları bulma konusunda başarılı bir algoritmadır. Bu algoritmanın odak bireyler kavramı ile farklılaştırılması ve amaç fonksiyonu olarak öbekleme analizine yönelik bir fonksiyonun tanımlanması ile bu çalışmada yer alan sonuçlara ulaşılmıştır.

Odak bireyli sürü zekası optimizasyon algoritmasının tesis yeri seçimi problemlerinde başarılı olmasının nedeni, diğer öbekleme algoritmalarından ve melez algoritmalarından farklı şekilde, öbekleme sırasında performans kriteri olarak, kullanıcının belirleyebileceği herhangi bir maliyet denkleminin kullanılabilmesidir. Bu sayede öbekleme analizi özellikle maliyetin en küçüklenmesini hedefleyen amaç fonksiyonuna göre çalışmaktadır.

Dördüncü bölümde de bahsedildiği gibi OBSO yönteminin bir tahmin yöntemi olarak kullanılması ayrı bir çalışmanın konusudur. Benzer şekilde kapasite kısıtı altında tesis yeri seçimi problemlerinin öbekleme analizi ile çözümü ve buna uygun bir algoritmanın geliştirilmesi yapılacak akademik çalışmaların bir sonraki adımı olabilir. OBSO yönteminin performansına etki eden odak bireylerin davranışı ile ilgili farklı yaklaşımların ele alınması, PSO' nun performansını arttırmaya yönelik geliştirilen tekniklerin OBSO yöntemine uygulanması da diğer çalışmalarda incelenebilecek konulardır.

## KAYNAKLAR

- ALBINO, V., GARAVELLI, A.C., GORGOGLIONE, M., 1998, Fuzzy logic in vendor rating: a comparison between fuzzy logic system and neural network, *Fuzzy Economic Review*, 3 (2), 25-48.
- ANGELINE, P., 1998, Using Selection to Improve Particle Swarm Optimization, *Proceedings of the 1998 International Conference on Evolutionary Computation*, Alaska, USA, Piscataway NJ: IEEE Press, 84-89.
- BABUSKA, R., VEEN, P.J., KAYMAK, U., 2002, Improved covariance estimation for Gustafson Kessel clustering. *Proceedings of 2002 IEEE International Conference on Fuzzy Systems*, 1081 - 1085, Honolulu, Hawaii.
- BALASKO B., ABONYI J. and FEIL B., 2005, Fuzzy Clustering and Data Analysis Toolbox For Use with MATLAB. <http://www.fmt.vein.hu/softcomp/fclusttoolbox>
- BALLOU R., 1999, *Business Logistics Management*, Fourth Edition, Prentice-Hall, Inc. Upper Saddle River, New Jersey.
- BERGH, F., 2001, *An Analysis of Particle Swarm Optimizers*, Doktora Tezi, Faculty of Natural and Agricultural Science of University of Pretoria
- BERGH, F., ENGELBRECHT, A.P., 2006, A Study of Particle Swarm Optimization Particle Trajectories, *Information Sciences*, 176, 937-971.
- BEZDEK, J.C., 1981, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York.
- BEZDEK J.C., DUNN J.C., 1975, Optimal fuzzy partitions: A heuristic for estimating the parameters in a mixture of normal distributions. *IEEE Transactions on Computers*, 835-838.
- BHARITKAR, S., KYRIAKIS, C. 2001, Cluster centroid method for room response equalization at multiple locations, *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 55-58.

- BONABEAU, E., MEYER, C., 2001, Swarm Intelligence: A Whole New Way to Think about Business, *Harvard Business review*, Harvard Business School Publishing Corporation.
- CAN T., ÇİLİNGİRTÜRK, M., KOÇAK, H., 2006, Konveks Programlama ile Lojistik Merkezlerinin Tespiti, *Yönetim:İstanbul Üniversitesi İşletme Fakültesi İşletme İktisadi Enstitüsü Dergisi*, Yıl:17, Sayı:54-Haziran, 17-25.
- CASTRO, L.N., 2002, Immune, Swarm and Evolutionary Algorithms Part I: Basic Models, *Proceedings of The 9<sup>th</sup> International Conference on Neural Information Processing*, Vol 3, 1464-1468.
- CHEN, C., YE, F., 2004, Particle Swarm Optimization Algorithm and Its Application to Clustering Analysis, *Proceeding of the IEEE Conference on Networking, Sensing & Control*, Taipei, Taiwan, 789-794.
- CHENG, T.W., GOLDFOF, D.B., HALL, L.O. 1998, Fast fuzzy clustering , *Fuzzy Sets and Systems*, 93, 49-56.
- CHRISTOPHER, M., 1998, *Logistics and supply chain management*, Prentice Hall, İngiltere, 0273630490.
- CLERC, M., KENNEDY, J., 2002, The particle Swarm: Explosion, Stability and Convergence in a Multi-Dimensional Complex Space, *IEEE Transactions on Evolutionary Computation*, Vol 6. No 1, 58-73.
- CLERC, M., 2006, *Particle Swarm Optimization*, ISTE, CA, 1-905209-04-5.
- CORREA, E., FREITAS, A., JOHNSON, C., 2006, A New Discrete Particle Swarm Algorithm Applied to Attribute Selection in a Bioinformatics Data Set, *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, Washington ABD, 35-42
- CUI, X., POTOK, T., PALATHINGAL, P., 2005, Document Clustering using Particle Swarm Optimization, *Proceedings of IEEE 2005 Swarm Intelligence Symposium*, 8-10 Haziran 2005, TN, 185 – 191.
- DÖRİNG C., LESOT M., KRUSE R., 2006, Data analysis with fuzzy clustering methods, *Computational Statistics & Data Analysis*, 51, 192-214.
- DUNN J.C. (1974) . A fuzzy relative of the ISODATA process and its use in detecting compact well-seperated clusters , *Journal of Cybernetcis*, Vol.3,No.3,32-57.
- EBERHART, R.C., SHI, Y., 2000, Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization, *Proceedings of The IEEE Congress on Evolutionary Computation*, San Diego, USA, 84-88.

- EMAMI, M. R., Turksen I.B., Goldenberg, A. A. (1996). An improved fuzzy modeling algorithm, Part II: System Identification, *Proceedings of 1996 Biennial Conference of the North American Fuzzy Information Processing Society-NAFIPS*, 294-298.
- ESNAF, S., 1998, Incremental Change Analysis-Based Fuzzy Objective Modeling in Medicine: Serum Lithium Concentration Prediction, *Istanbul Universitesi, Isletme Fakultesi Dergisi*, 27, 19-28.
- ESNAF, S., KÜÇÜKDENİZ, T., 2006, A Fuzzy Clustering-Based Hybrid Method For A Multi-Facility Location Problem, *Proceedings of The 5th International Symposium On Intelligent Manufacturing Systems*, 304-317.
- ESNAF Ş., KÜÇÜKDENİZ T., 2007, Kapasite Kısıtlı Atama Problemleri İçin Bulanık Öbekleme Esaslı Bir Çözüm Algoritması, *VI. Endüstri-İşletme Mühendisliği Kurultayı*, 09-10 Kasım 2007, Bursa.
- ESNAF, Ş., KÜÇÜKDENİZ, T., 2008, Fuzzy C-Means and Center of Gravity Combined Model for a Capacitated Planar Multiple Facility Location Problem, *International Conference On Multivariate Statistical Modeling and High dimensional Data Mining*, 19-23 Haziran 2008, Kayseri, 131.
- ESNAF, Ş., KÜÇÜKDENİZ, T., 2009, A Fuzzy Clustering-Based Hybrid Method For A Multi-Facility Location Problem, *Journal of Intelligent Manufacturing*, 20 (2), 259-265
- ESCHRICH, S., KE, J.W., HALL, L.O., GOLDFOF, D.B. 2003, Fast accurate fuzzy clustering through data reduction, *IEEE Transactions on Fuzzy Systems*, 11 (2), 262- 270.
- FILHO, V.A.D., 2006, *Portfolio Management Using Value at Risk: A Comparison Between Genetic Algorithms and Particle Swarm Optimization*, Master Thesis, University Rotterdam Informatics & Economics.
- FOX, M. BARBUCEANU M., TEIGEN R., 2000, Agent-Oriented Supply Chain Management, *The International Journal of Flexible Manufacturing Systems*, 12, 165-188.
- GALVAO, R.D., 2004, Uncapacitated Facility Location Problems: Contributions, *Pesquisa Operacional*, 24 (1), 7-38.
- GIANNOCCARO, I., PONTRANDOLFO, P., 2001, Models for supply chain management: A taxonomy, *Twelfth annual conference of the production and operations management society*, Mart 30-Nisan 2 2001, Orlando FL.
- GUNER, A.R., SEVKLI, M., 2008, A Discrete Particle Swarm Optimization Algorithm for Uncapacitated Facility Location Problem, *Journal of Artificial Evolution and Applications*, Vol 2008.

- GUSTAFSON D.E., KESSEL W.C., 1979, Fuzzy clustering with fuzzy covariance matrix. *In Proceedings of the IEEE CDC*, San Diego, 761-766.
- HEPPNER, F., GRENANDER, U., 1990, A Stochastic Nonlinear Model for Coordinated bird Flocks, *The Ubiquity of Chaos*, AAAS Publications, Washington DC.
- HOULIHAN, J., 1984, Supply Chain Management, *Proceedings of the 19th International Technical Conference*, BPICS.
- HU T., SHEU J., 2003, A fuzzy-based customer classification method for Demand-responsive logistical distribution operations, *Fuzzy Sets and Systems*, 139, 431-450.
- Jİ, C., ZHANG, Y., GAO, S., 2004, Particle Swarm Optimization for Mobile Ad Hoc Networks Clustering, *IEEE International Conference on Networking, Sensing and Control*, 1, 372-375.
- KAEWKAMNERDPONG, B., BENTLEY, P., 2005, Perceptive Particle Swarm Optimisation: An Investigation, *2005 IEEE Swarm Intelligence Symposium*, June 8-10, 2005, Pasadena, California.
- KENNEDY, J., 1999, Small Worlds and Mega-Minds: Effects of Neighborhood Topology on Particle Swarm Performance, *Proceedings of the 1999 Conference on Evolutionary Computation*, Piscataway NJ: IEEE Press, 1931-1938.
- KENNEDY, J., 2000, Stereotyping: Improving Particle Swarm Performance With Cluster Analysis, *Proceedings of the 2000 Congress on Evolutionary Computation*, 2, 1507 – 1512.
- KENNEDY, J., EBERHART, R.C., 1995, Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, NJ, USA, IEEE Service Center, 1942-1948.
- KENNEDY, J., RUI, M., 2002, Population Structure and Particle Swarm Performance, *Proceedings of the Evolutionary Computation on 2002*, 1671-1676.
- KOGAN, J., NICHOLAS, C., TEBOULLE, M., 2006, *Grouping Multidimensional Data: Recent Advances in Clustering*, Springer, Netherlands, 3-540-28348-X.
- KÜÇÜKDENİZ, T., 2004, *Tedarik Zinciri Yönetiminde Bulanık Modelleme*, Yüksek Lisans Tezi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü
- LAMBERT, D.M., COOPER, M.C., PAGH, J.D., 1998, Supply Chain Management: Implementation Issues and Research Opportunities, *The International Journal of Logistics Management*, 9(2), 1-12.



- LANGDON, W.B., POLI, R., 2007, Evolving Problems to Learn about Particle Swarm and other Optimisers, *IEEE transactions on evolutionary computation*, 11 (5), 561-578.
- LASKARI, E.C., PAROPOULOS, K.E., VRAHATIS, M.N., 2002, Particle Swarm Optimization for Integer Programming, *Proceedings of the IEEE 2002 Congress on Evolutionary Computation*, Honolulu (HI), 1582-1587.
- LEVIN Y., BEN-ISRAEL A., 2004, A heuristic method for large-scale multi-facility location problems, *Computers & Operations Research*, 31, 257–272.
- LÌ, X., 2004, Adaptively Choosing Neighborhood Bests Using Species in a Particle Swarm Optimizer for Multimodal Function Optimization, *Proceedings of Genetic and Evolutionary Computation Conference*, 3102, 105-116.
- LİN, F., TAN, G., SHAW, M., 1998, Modeling Supply-Chain Networks by a Multi-Agent System, *Proceedings of the Thirty-First Annual Hawaii International Conference on System Sciences*, 6-9 Ocak 1998, 105-112.
- LU, J., YUAN, X., YAHAGI, T. 2006, A method of face recognition based on fuzzy clustering and parallel neural networks, *Signal Processing*, 86, 2026-2039.
- MERWE V. D., ENGELBRECHT, A. P., 2003, Data clustering using particle swarm optimization. *Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003)*, Canbella, Australia, 215-220.
- NAKA, S., GENJÌ, T., YURA, T., FUKUYAMA, Y., 2001, Practical Distribution State Estimation Using Hybrid Particle Swarm Optimization, *IEEE Power Engineering Society Winter Meeting*, Columbus, USA, 815-820.
- NASCIMENTO, S., SOUSA, F., CASIMIRO, H., BOUTOV, D. 2005, Applicability of fuzzy clustering for the identification of upwelling areas on sea surface temperature images, *UK Workshop on Computational Intelligence*, 5-7 Eylül, 143-148.
- NEDJAH, N., MOURELLE, L.M., 2006, *Swarm Intelligent Systems*, Springer-Verlag, ISBN: 3-54033-868-3
- NEGRESIOS, M., PALHANO, A., 2006, The capacitated centered clustering problem, *Computers & Operations Research*, 33, 1639-1663.
- OMRAN, M., ENGELBRECHT, A. P, SALMAN, A., 2005., Dynamic Clustering Using Particle Swarm Optimization with Application in Unsupervised Image Classification, 199 – 204.
- OMRAN, M., SALMAN, A. and ENGELBRECHT, A. P., 2002. Image classification using particle Swarm optimization, *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning 2002 (SEAL 2002)*, Singapore, 370-374.

- ÖZÇAKAR, N., ESNAF, Ş., 1998, Fuzzy Clustering-Based Part-Family Formation in a Multicell Flexible Manufacturing System, *Istanbul Universitesi, Isletme Fakultesi Dergisi*, 27, 19-28.
- OZCAN, E., MOHAN, C.K., 1998, Analysis of a Simple Particle Swarm Optimization System, *Intelligent Engineering Systems Through Artificial Neural Networks*, 253-258.
- OZCAN, E., MOHAN, C.K., 1999, Particle Swarm Optimization: Surfing The Waves, *Proceedings of The IEEE Congress on Evolutionary Computation*, Piscataway, USA, 69-73.
- QUINN F.J., 2000, "The Clockspeed Chronicles," *Supply Chain Management Review*, 3 (4), 60-64.
- REVELLE C.S., EISELT H.A., 2005, Location analysis: a synthesis and survey, *European Journal of Operational Research*, Vol.165, 1-19.
- REYNOLDS, C.W., 1987, Flocks, herds and schools: a distributed behavioral model. *Computer Graphics* 2(2), 91-108.
- ROSS, D.F., 1998, *Competing Through Supply Chain Management: Creating Market-Winning Strategies through Supply Chain Partnerships*, Chapman & Hall, 0412137216.
- RUI, M., 2004, *Population Topologies and Their Influence in Particle Swarm Performance*, Doktora Tezi, Departamento de Informatica, Universidade do Minho
- SETTLES, M., RYLANDER, B., 2002, Neural Network Learning Using Particle Swarm Optimizers, *Advances in Information Science and Soft Computing*, 224-226.
- SEVKLI, M., GUNER, A.R., 2006, A New Approach to Solve Uncapacitated Facility Location Problems by Particle Swarm Optimization, *Proceedings of 5th International Symposium on Intelligent Manufacturing Systems*, Mayıs 29-31, 237-246
- SHEU J., 2002, A fuzzy clustering-based approach to automatic freeway incident detection and characterization, *Fuzzy Sets and Systems*, 128, 377-388.
- SHEU J., 2006, A novel dynamic resource allocation model for demand-responsive city logistics distribution operations, *Transportation Research Part E: Logistics and Transportation Review*, 42 (6), 445-472.
- SHEU J., 2007, A hybrid fuzzy optimization approach to customer grouping-based logistics distribution to operations, *Applied Mathematical Modeling*, 31, 1048-1066.

- SHI, Y. H., EBERHART, R. C., 1998, Parameter Selection in Particle Swarm Optimization, *Proceedings of The 7<sup>th</sup> Annual Conference on Evolutionary Programming*, San Diego, CA. 591-600.
- SHI, Y., EBERHART, R.C., 1998, A Modified Particle Swarm Optimizer, *Proceedings of IEEE International Conference of Evolutionary Computation*, Anchorage, Alaska.
- SHI, Y., EBERHART, R.C., 1999, Empirical Study of Particle Swarm Optimization, *Proceedings of the 1999 Congress on Evolutionary Computation*, 3, 1950
- SHI, Y., EBERHART, R.C., 2001, Fuzzy Adaptive Particle Swarm Optimization, *Proceedings of The IEEE Congress on Evolutionary Computation*, IEEE Press: Soul, Korea.
- SIERRA, M.R., COELLO, C., 2006, Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art, *International Journal of Computational Intelligence Research*, Vol. 2, No. 3.
- SUGANTHAN, P.N., 1999, Particle Swarm Optimizer with Neighborhood operator, *Proceedings of The IEEE Congress on Evolutionary Computation*, IEEE Press: Piscataway, USA, 1958 – 1962.
- SULE D.R., 2001, *Logistics of facility location and allocation*, Marcel Dekker Inc., New York, 0824704932.
- TILLET J., Rao T.M., Sahin F., and Rao R., 2005, Darwinian Particle Swarm Optimization, *in the Proceedings of the 2<sup>nd</sup> Indian International Conference on Artificial Intelligence (IICAI-05)*, pp. 1474-1487, December 20-22 2005, Pune, India.
- THAI, A.B., 2007, *Evaluation of Using Particle Swarm Intelligence To Produce Facility Layout Solutions*, Master of Engineering, University of Louisville School of Engineering Dept. of Computer Science.
- TOWILL, D.R., 1996, Industrial dynamics modeling of supply chains, *International Journal of Physical Distribution & Logistics Management*, 26 (2), 23-42
- TRELEA, I.C., 2003, The Particle Swarm Optimization, Algorithm: Convergence Analysis and Parameter Selection, *Information Processing Letters*, 85, 317-325.
- TÜRKŞEN, I.B., ESNAF, Ş., 1997, Financial Data Forecasting Using Fuzzy Objective Modeling, *Proc. IV. Congress of The International Society of Fuzzy Management and Economics, Santiago de Cuba*, 2, 1-16.
- TÜRKŞEN, I.B., ESNAF, Ş., 1998, Spare Parts Problem Revisited, *1st National Production Conference*, Eylül 30-31, 334-338.

- UJJIN, S. ve BENTLEY, P. J., 2003, Particle Swarm Optimization Recommender System. *In Proceedings of the IEEE Swarm Intelligence Symposium*,. Indianapolis, 124 - 131.
- UTKAN, F., 2006, *Tesis Yerleşim Problemleri için Takım Zekası Tabanlı Bir Rassal Eniyileme Algoritması*, Yüksek Lisans, Osmangazi Üniversitesi Fen Bilimleri Enstitüsü.
- WHITE, T., 1997, *Routing with Swarm Intelligence, Technical Report SCE-97-15*, Systems and Computer Engineering, Carleton University, September 1997.
- XIA, W., WU, Z., 2005, An Effective Hybrid Optimization Approach for Multi-Objective Flexible Job-Shop Scheduling Problems, *Computers And Industrial Engineering*, Vol 48, 409-425.
- YANO, F., SHAHDOHJI, T., TOYODA, Y., 2008, Modification of Hybridized Particle Swarm Optimization Algorithms Applying to Facility Location Problems, *Proceedings of the 9th Asia Pasific Industrial Engineering & Management Systems Conference*, 3-5 Aralık, Endonezya, 2278 – 2287
- YAPICIOĞLU, H., DOZIER, G., SMITH, A., 2004, Bi-Criteria Model for Locating a Semi-Desirable Facility on a Plane Using Particle Swarm Optimization, *Proceedings of the 2004 congress on evolutionary computation*, 2371
- YOSHIDA, H., KAWATA, K., FUKUYAMA, Y., NAKANISHI, Y., 2000, A Particle Swarm Optimization For Reactive Power and Voltage Control Considering Voltage Stability, *Proceedings of The International Conference on Intelligent System Application to Power System*, Rio de Janeiro, Brazil, 117-121.
- ZALIK, K.R., 2006, Fuzzy C-Means Clustering And Facility Location Problems, *Proceedings of the 10<sup>th</sup> IASTED International conference on Artificial Intelligence and Soft Computing*, 28-30 Ağu. 2006, Palma de Mallorca, İspanya, 256-261

## **ÖZGEÇMİŞ**

Tarık Küçükdeniz, 1979 yılında İstanbul' da doğdu. İlk ve ortaöğreniminin ardından 1997 yılında Kabataş Erkek Lisesi' nden mezun oldu. Aynı yıl girdiği İstanbul Üniversitesi Endüstri Mühendisliği Bölümünün 2001 yılında lisans, 2004 yılında da yüksek lisans programlarını bitirdi. 2001 yılından bu güne dek aynı bölümde araştırma görevlisi olarak görev almaktadır.

1.	GİRİŞ .....	1
1.2.	KATKILAR .....	2
1.3.	TEZİN ANA HATLARI .....	3
2.	GENEL KISIMLAR .....	4
2.1.	TEDARİK ZİNCİRİ YÖNTEİMİ .....	4
2.1.1.	Tedarik Zinciri Kavramı .....	4
2.1.2.	Tedarik Zinciri Yönetimi Kavramı .....	5
2.1.3.	Tedarik Zinciri Yönetimi Bileşenleri .....	6
2.2.	TEDARİK ZİNCİRİNİN MODELLENMESİ .....	6
2.2.1.	Tedarik Zinciri Problemleri .....	7
2.3.	TEDARİK ZİNCİRİ YÖNETİMİ MODELLERİ .....	9
2.3.1.	Kavramsal Modeller .....	9
2.3.2.	Analitik Modeller .....	10
2.3.3.	Yapay Zekâ Tabanlı ve Sezgisel Modeller .....	11
2.3.4.	Simülasyon Modelleri .....	12
2.4.	TESİS YERİ SEÇİMİ PROBLEMİ .....	13
2.4.1.	Kapasite Kısıtsız Çoklu Tesis Yeri Seçimi Problemleri .....	15
2.4.2.	Çoklu Tesis Yeri Seçimi Problemlerinde Öbekleme Analizi .....	16
2.4.3.	Talep Noktalarının Coğrafi Öbeklemesi İçin Bulanık C-Ortalamalar Algoritması .....	18
2.4.4.	Talep Noktalarının Coğrafi Öbeklemesi İçin Gustafson – Kessel Öbekleme Algoritması .....	20
2.4.5.	Tesis Yerlerinin Tespiti için Melez Yöntemler .....	23
2.4.6.	Bulanık Öbekleme - Ağırlık Merkezi Melez Yöntemi .....	23
2.4.7.	Bulanık Öbekleme - Konveks Programlama Melez Yöntemi .....	24
2.4.8.	Ulaştırma Maliyetinin Hesaplanması .....	25
2.5.	SÜRÜ ZEKASI .....	26
2.6.	PARÇACIK SÜRÜ OPTİMİZASYONU YÖNTEMİ .....	28
2.6.1.	Çıkış Noktası .....	28
2.6.2.	Standart Algoritma .....	30
2.6.3.	Uygulama Alanları .....	36
2.6.4.	PSO Yöntemi ile Evrimsel Hesaplama Yöntemleri Arasındaki İlişki .....	39
2.6.5.	Performansı Artırmaya Yönelik Düzenlemeler .....	39
2.6.6.	1'' Tip Parçacık Sürü Optimizasyonu Modeli .....	42
2.6.7.	Sürünün En İyisi İçin Önerilen Yöntemler: .....	43
2.6.8.	PSO Tabanlı Öbekleme Analizi Yöntemleri .....	46
2.6.9.	PSO Tabanlı Tesis Yeri Seçimi Yöntemleri .....	48
3.	MALZEME VE YÖNTEM .....	50
3.1.	ODAK BİREYLİ SÜRÜ OPTİMİZASYONU ÖBEKLEME YÖNTEMİ .....	50
3.1.1.	Modelin Yapısı .....	51
3.1.2.	Başlangıç İşlemleri .....	53
4.	BULGULAR .....	57
4.1.	V İndeksi .....	58
4.2.	Dunn İndeksi .....	59
4.3.	Test Veri Kümelerinde Sonuçlar .....	60
4.3.1.	I. Test Veri Kümesi (IRIS Veri Kümesi) .....	60
4.3.2.	II. Test Veri Kümesi .....	63
4.3.3.	III. Test Veri Kümesi .....	69
4.3.4.	Tesis Yeri Seçimi Probleminde OBSO Yöntemini uygulanması .....	73

5. TARTIŞMA VE SONUÇ .....	77
KAYNAKLAR .....	82
ÖZGEÇMİŞ .....	90
Tablo 2.1: Standart PSO algoritmasının parametreleri .....	35
Tablo 4.2: Iris veri kümesinde öbikleme yöntemlerinin Dunn indeksine göre başarımı (Büyük değerler daha iyi).....	62
Tablo 4.3: Iris veri kümesinde öbikleme yöntemlerinin $V$ indeksine göre başarımı (Küçük değerler daha iyi).....	62
Tablo 4.4: 51 noktalı veri kümesinde öbikleme yöntemlerinin $V$ indeksine göre başarımı (Küçük değerler daha iyi).....	68
Tablo 4.5: 51 noktalı veri kümesinde öbikleme yöntemlerinin Dunn indeksine göre başarımı (Büyük değerler daha iyi).....	69
Tablo 4.6: 400 noktalı veri kümesinde öbikleme yöntemlerinin $V$ indeksine göre başarımı (Küçük değerler daha iyi).....	72
Tablo 4.7: 400 noktalı veri kümesinde öbikleme yöntemlerinin Dunn indeksine göre başarımı (Büyük değerler daha iyi).....	72
Tablo 4.8: Tesis yeri seçimi problemi için kullanılan 51 noktanın koordinat ve talep verisi.....	74
Tablo 4.9: Tesis yeri seçimi probleminde hesaplanan maliyet parametresi değerleri ....	75
Tablo 4.10: Tesis yeri problemi için sonuç öbek merkezi koordinatları.....	75
Tablo 4.11: Tesis yeri problemi için sonuç öbek merkezi koordinatları.....	76

