

**ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

YÜKSEK LİSANS TEZİ

Alper Tuna KAVLAK

**BULANIK C-ORTALAMALAR, OLABİLİRLİKLI C-
ORTALAMALAR VE KARMA KÜMELEME
ALGORİTMALARININ ETKİNLİKLERİNİN
KARŞILAŞTIRILMASI**

ZOOTEKNİ ANABİLİM DALI

ADANA-2017

**ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**BULANIK C-ORTALAMALAR, OLABİLİRLİK Lİ C-ORTALAMALAR
VE KARMA KÜMELEME ALGORİTMALARININ ETKİNLİKLERİNİN
KARŞILAŞTIRILMASI**

Alper Tuna KAVLAK

YÜKSEK LİSANS TEZİ

ZOOTEKNİ ANABİLİM DALI

Bu Tez 25/05/2017 Tarihinde Aşağıdaki Jüri Üyeleri Tarafından
Oybirliği/Oyçokluğu ile Kabul Edilmiştir.

.....
Prof. Dr. Zeynel CEBECİ
DANIŞMAN

.....
Prof. Dr. G. Tamer KAYAALP
ÜYE

.....
Prof. Dr. Ercan EFE
ÜYE

Bu Tez Enstitümüz Zooteknî Anabilim Dalında hazırlanmıştır.
Kod No:

Prof. Dr. Mustafa GÖK
Enstitü Müdürü

Bu Çalışma Ç. Ü. Araştırma Projeleri Birimi Tarafından Desteklenmiştir.
Proje No: FYL-2015-5186

Not: Bu tezde kullanılan özgün ve başka kaynaktan yapılan bildirişlerin, çizelge ve fotoğrafların kaynak gösterilmeden kullanımı, 5846 sayılı Fikir ve Sanat Eserleri Kanunundaki hükümlere tabidir.

ÖZ

YÜKSEK LİSANS TEZİ

**BULANIK C-ORTALAMALAR, OLABİLİRLİKLİ C-ORTALAMALAR
VE KARMA KÜMELEME ALGORİTMALARININ ETKİNLİKLERİNİN
KARŞILAŞTIRILMASI**

Alper Tuna KAVLAK

**ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ZOOTEKNİ ANABİLİM DALI**

Danışman : Prof. Dr. Zeynel CEBECİ
Yıl: 2017, Sayfa: 394
Jüri : Prof. Dr. Zeynel CEBECİ
: Prof. Dr. G. Tamer KAYAALP
: Prof. Dr.Ercan EFE

Bu çalışmada olasılıklı ve olabilirlikli yumuşak bölümleyici kümeleme algoritmalarından Bulanık C-ortalamalar (BCO), Olabilirlikli C-ortamalar (OCO), bunların hibritleri olarak Bulanık Olabilirlikli C-Ortamalar (BOCO), Olabilirlikli Bulanık C-ortalamalar (OBCO) ile İtmeli Olabilirlikli C-Ortamalar (iOCO) ve Yönetimsiz Olabilirlikli Bulanık Kümeleme (UOBCO) algoritmalarının kümeleme başarımları karşılaştırılmıştır. Algoritma başarımlarını test etmek için kare, elips, daire ve konkav şekilli gürültülü ve gürültüsüz 8 adet sentetik veri seti ve 4 gerçek veri seti kullanılmıştır. Kümeleme geçerlilik indeksleri olarak Bölümleme Entropisi (PE), Bölümleme Katsayısı (PC), Değiştirilmiş Bölümleme Katsayısı (MPC), Xie-Beni (XB), Kwon, Tang-Sun (TS) ve Bulanık siluet indeksleri (FS) kullanılmıştır. Yapılan analizlere göre UOBCO algoritması daha düşük işlem zamanında geçerliliği yüksek kümeleme sonuçları ürettiğinden bulanık ve gürültülü büyük veri setlerinde kümeleme analizi için önerilmiştir.

Anahtar Kelimeler: Kümeleme analizi, kümeleme algoritması, bulanık kümeleme, olabilirlikli kümeleme, kümeleme geçerlilik indeksleri.

ABSTRACT

MSc THESIS

COMPARISON OF CLUSTERING PERFORMANCES OF FUZZY C-MEANS, POSSIBILISTIC C-MEANS AND SOME FUZZY AND POSSIBILISTIC HYBRID ALGORITHMS

Alper Tuna KAVLAK

ÇUKUROVA UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ANIMAL SCIENCE

Supervisor : Prof. Dr. Zeynel CEBECİ

Year: 2017, Pages: 394

Jury : Prof. Dr. Zeynel CEBECİ

: Prof. Dr. G. Tamer KAYAALP

: Prof. Dr.Ercan EFE

In this study, we compared the performance of Fuzzy C-Means (FCM) and Possibilistic C-Means (PCM) and Possibilistic C-Means with Repulsion (PCMR), and their hybrids such as Fuzzy Possibilistic C-Means (FPCM), Possibilistic Fuzzy C-Means (PFCM), Possibilistic C-Means with Repulsion (PCMR) and Unsupervised Possibilistic Fuzzy C-Means (UPFCM). Four real data sets and eight synthetic data sets consists of various square, ellipse, circle and concave shaped clusters with some noises were used for testing the performance of the algorithms. Partition Entropy (PE), Partition Coefficient (PC), Modified Partition Coefficient (MPC), Xie-Beni (XB), Kwon, Tang-Sun (TS) and Fuzzy Silhouette (FS) were used as clustering validity indices for finding the optimal numbers of clusters in the analyzed data sets. According to the obtained results, UPFCM is proposed for partitioning of fuzzy and noisy large data sets because of its computational efficiency and success to find the optimal clustering results.

Key Words: Clustering analysis, clustering algorithms, fuzzy clustering, possibilistic clustering, clustering validity indices.

GENİŞLETİLMİŞ ÖZET

Bu çalışmada olasılıklı ve olabilirlikli yumuşak bölümleyici kümeleme algoritmalarından Bulanık C-ortalamlar (BCO), Olabilirlikli C-ortalamlar (OCO), bunların hibritleri olarak Bulanık Olabilirlikli C-Ortalamlar (BOCO), Olabilirlikli Bulanık C-ortalamlar (OBCO) ile İtmeli Olabilirlikli C-Ortalamlar (İOCO) ve Yönetimsiz Olabilirlikli Bulanık Kümeleme (UOBCO) algoritmalarının kümeleme başarımları karşılaştırılmıştır. Algoritma başarımlarını test etmek için kare, elips, daire ve konkav şekilli gürültülü ve gürültüsüz 8 adet sentetik veri seti ve 4 gerçek veri seti kullanılmıştır. Kümeleme geçerlilik indkesleri olarak Bölümleme Entropisi (PE), Bölümleme Katsayısı (PC), Değiştirilmiş Bölümleme Katsayısı (MPC), Xie-Beni (XB), Kwon, Tang-Sun (TS) ve Bulanık siluet indeksleri (FS) kullanılmıştır. Dahası üyelik ve tipiklik derecelerinin kombine edilmesine dayanan Uzatılmış Bulanık İndeksleri (Extended Fuzzy Indices) de kullanılmıştır.

Analizler öncesi küme ve özellik sayısı belirlenen gerçek ve yapay veri setlerine seçilen kümeleme algoritmaları ve kümeleme geçerlilik indeksleri uygulanmıştır. Algoritmaların veri setleri üzerindeki başarımlarının yani sıra hem genelleştirilmiş bulanık kümeleme indekslerinin hem de uzatılmış indeks değerlerinin sonuçları çizelgelerle ve şekillerle gösterilmiştir.

Yapılan analizler sonucunda veri setleri üzerine uygulanan algoritmaların seçtiği küme sayıları 5.1., 5.3. ve 5.5.'te gösterilmiştir. İndekslerden alınan sonuçlar dâhilinde PE ve uzatılmış versiyonu olarak EPE (Uzatılmış Bölümleme Entropisi) 'nin hemen algoritma sonuçlarına göre farklı tepki vermediği ve küme sayısını tüm veri setleri için 2 olarak önermiş ve bu nedenle çalışılan veri setlerinde kümeleme doğruluğunu ortaya koymada yetersiz kaldığı anlaşılmaktadır. Sentetik gürültüsüz veri setlerinden elde edilen iterasyon sonuçlarına göre İOCO algoritması kümeleme analizini göre 11-16 gibi çok az sayıda iterasyonla tamamlarken OCO algoritması 81-177 iterasyonla en yüksek iterasyon sayısına sahip algoritma olmuştur. OBCO da İOCO da olduğu gibi düşük iterasyon

değerleri göstermektedir. Bu durum gürültülü veri setlerinde de çok farklı bir durum göstermemektedir. Genel anlamda kümeleme algoritmaları RDS1-Iris (Iris Gerçek Veri Seti) veri seti üzerinden çok kısa surede ve az iterasyon sayısı ile sağlam sonuçlar verse de bu durum diğer gerçek veri setleri için aynı olmadığı görülmektedir.

Amaç fonksiyonlar açısından, değerler iyi kümeleme sonuçlarına yaklaşırken küçülmekte ve uzaklaştıkça büyümektedirler. Bu nedenle amaç fonksiyonunun en küçük olduğu kümelemeyi en iyi kümeleme olarak değerlendirdiğimizde UOBOCO algoritmasındaki Çizelge 4.171 ve Şekil 4.109'daki grafiğe göre SDSC (Çizelge 3.1'e göre daire şekilli-gürültüsüz veri seti) için en iyi kümelemenin $c=8$ için elde edildiği görülecektir. Sentetik ve gerçek veri setlerinde geçerlilik indeksleri ile yapılan kümeleme geçerliliklerine göre algoritmalar arasında farklılıklar bulunmadığı; ancak mevcut algoritmaların konkav ve yuvarlak şekilli kümeler içeren SDSX (Çizelge 3.1'e göre konkav şekilli-gürültüsüz veri seti) ve SDSXN (Çizelge 3.1'e göre konkav şekilli-gürültüsüz veri seti) veri setlerinde iyi çalışmadığı anlaşılmaktadır. .

Karma algoritmalarından OBCO ve BOCO algoritmaları benzer başarımlara sahip olmakla birlikte BOCO algoritmasının nispeten daha iyi sonuçlar verdiği görülmüştür. Ek Bölümünde yer alan grafikler algoritmaların performanslarını gürültü ve aykırı değerler bakımından karşılaştırılmasında daha hassas yorumlamalar yapılmasını sağlayabilecektir

Sonuç olarak gerek daha önce çalıştırılan bir BCO algoritmasından üyelik dereceleri matrisi istememesi, gerek gürültülü ortamlarda iyi çalışması ve gerekse diğerlerine göre benzer çalışma performansı göstermesi açısından UOBOCO algoritmasının kullanılması önerilmektedir.

Yapılan analizlere göre UOBOCO algoritması daha düşük işlem zamanında geçerliliği yüksek kümeleme sonuçları ürettiğinden bulanık ve gürültülü büyük veri setlerinde kümeleme analizi için önerilmiştir.

TEŐEKKÜR

“Bulanık C-Ortalamalar, Olabilirlikli C-ortalamalar ve Karma Kùmeleme Algoritmalarının Etkinliklerinin Karşılaştırılması” konulu bu yüksek lisans tezi konusunu veren, yapıcı ve yönlendirici fikirleri ile bana daima yol gösteren danışman hocam Sayın Prof. Dr. Zeynel CEBECİ’ ye sonsuz teşekkürlerimi sunarım.

Yüksek Lisans çalışmalarım esnasında tüm bölüm olanaklarından yararlanmamı sağlayan Ç.Ü. Ziraat Fakóltesi Zootekni Bölüm Başkanlığı’na, maddi destek veren Ç.Ü. Bilimsel Araştırma Projeleri Birimi’ne (Proje no: 5186) içten teşekkürlerimi sunarım.

Yüksek Lisans çalışmalarım suresince maddi ve manevi desteklerini hiçbir şekilde esirgemeyen, her zaman bana destek olan annem Nurdan KAVLAK’a, babam Yaşar Ceyhan KAVLAK’a, ağabeyim Vahap KAVLAK’a ve ayrıca yakın arkadaşım Utku UZUN’ a da teşekkürlerimi sunarım.

İÇİNDEKİLER

SAYFA

ÖZ.....	I
ABSTRACT	II
GENİŞLETİLMİŞ ÖZET	III
TEŞEKKÜR	V
İÇİNDEKİLER.....	VI
ÇİZELGELER DİZİNİ.....	VIII
ŞEKİLLER DİZİNİ.....	XXII
SİMGELER VE KISALTMALAR	XXX
1. GİRİŞ.....	1
2. ÖNCEKİ ÇALIŞMALAR	7
3. MATERYAL VE METOT.....	25
3.1. Materyal.....	25
3.1.1. Sentetik Veri Setleri.....	25
3.1.2. Gerçek Veri Setleri	29
3.2. Metot	31
3.2.1. Bölümleyici Kümeleme.....	31
3.2.2. Bulanık Ve Olabilirlikli Kümeleme Algoritmaları.....	46
3.3. Kümeleme Geçerlilik İndeksleri.....	58
3.3.1. Bulanık Kümeleme Geçerlilik İndeksleri	60
3.3.2. Genelleştirilmiş Ve Uzatılmış Geçerlilik İndeksleri.....	64
3.4. Veri Setlerinin Oluşturulması Ve Analizler	69
3.4.1. Veri Setleri Ve Kümeleme Sonuçlarının İncelenmesi.....	69
3.4.2. Algoritma Parametrelerinin Seçilmesi.....	69
3.4.3. Kümeleme Başarımlarının Karşılaştırılması.....	71
4. ARAŞTIRMA BULGULARI VE TARTIŞMA.....	73
4.1. BCO Algoritması.....	73

4.1.1. Sentetik Veri Setleri.....	73
4.1.2. Gerçek Veri Setleri	90
4.2. OCO Algoritması.....	98
4.2.1. Sentetik Veri Setleri.....	98
4.2.2. Gerçek Veri Setleri	123
4.3. BOCO Algoritması.....	135
4.3.1. Sentetik Veri Setleri.....	135
4.3.2. Gerçek Veri Setleri	161
4.4. OBCO Algoritması.....	173
4.4.1. Sentetik Veri Setleri.....	173
4.4.2. Gerçek Veri Setleri	198
4.5. İOCO Algoritması	210
4.5.1. Sentetik Veri Setleri.....	210
4.5.2. Gerçek Veri Setleri	237
4.6. UOBO Algoritması.....	249
4.6.1. Sentetik Veri Setleri.....	249
4.6.2. Gerçek Veri Setleri	274
5. TARTIŞMA VE SONUÇLAR.....	287
6. KAYNAKLAR.....	293
ÖZGEÇMİŞ.....	305
EKLER	306

ÇİZELGELER DİZİNİ**SAYFA**

Çizelge 3.1.	Sentetik Veri Setleri Ve Özellikleri	26
Çizelge 3.2.	Gerçek Veri Setleri Ve Özellikleri.....	29
Çizelge 3.3.	İki özellikli (p1 ve p2) örnek bir veri setinde (X16) nesnelere ait gözlem değerleri	33
Çizelge 3.4.	K-Ortalamalar Kümelemesine Göre C=2 Küme Sayısında Üyelik Dereceleri Matrisi	34
Çizelge 3.5.	K-Ortalamalar Kümelemesine Göre C=3 Küme Sayısında Üyelik Dereceleri Matrisi	35
Çizelge 3.6.	K-Ortalamalar Kümelemesine Göre C=4 Küme Sayısında Üyelik Dereceleri Matrisi	35
Çizelge 3.7.	K-Ortalamalar Kümelemesine Göre C=5 Küme Sayısında Üyelik Dereceleri Matrisi	35
Çizelge 3.8.	BCO'na Göre C=2 Küme Sayısında Üyelik Dereceleri Matrisi.....	37
Çizelge 3.9.	BCO'na Göre C=3 Küme Sayısında Üyelik Dereceleri Matrisi.....	38
Çizelge 3.10.	BCO'na Göre C=4 Küme Sayısında Üyelik Dereceleri Matrisi.....	39
Çizelge 3.11.	BCO'na Göre C=5 Küme Sayısında Üyelik Dereceleri Matrisi.....	40
Çizelge 3.12.	X16 veri setinde farklı küme sayılarına göre BCO kümelemesine göre indeks değerleri.....	41
Çizelge 3.13.	OCO Kümelemesine Göre C=2 Küme Sayısında Tipiklik Dereceleri Matrisi	42
Çizelge 3.14.	OCO Kümelemesine Göre C=3 Küme Sayısında Tipiklik Dereceleri Matrisi	43

Çizelge 3.15.	OCO kümelemesine göre $c=4$ küme sayısında tipiklik dereceleri matrisi	44
Çizelge 3.16.	OCO kümelemesine göre $c=5$ küme sayısında tipiklik dereceleri matrisi	45
Çizelge 3.17.	X16 veri seti için küme sayısına göre OCO algoritmasından elde edilen genelleştirilmiş ve uzatılmış indeks değerleri	46
Çizelge 4.1.	SDSC veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	73
Çizelge 4.2.	SDSC veri seti için BCO algoritması başarıımı.....	73
Çizelge 4.3.	SDSCN veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	76
Çizelge 4.4.	SDSCN veri seti için BCO algoritması başarıımı.....	76
Çizelge 4.5.	SDSE veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	78
Çizelge 4.6.	Çizelge 4.6. SDSE veri seti için BCO algoritması başarıımı.....	78
Çizelge 4.7.	SDSEN veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	80
Çizelge 4.8.	SDSEN veri seti için BCO algoritması başarıımı	80
Çizelge 4.9.	SDSR veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	82
Çizelge 4.10.	SDSR veri seti için BCO algoritması başarıımı.....	82
Çizelge 4.11.	SDSRN veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	84
Çizelge 4.12.	SDSRN veri seti için BCO algoritması başarıımı	84
Çizelge 4.13.	SDSX veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	86
Çizelge 4.14.	SDSX veri seti için BCO algoritması başarıımı.....	86
Çizelge 4.15.	SDSXN veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	88

Çizelge 4.16.	SDSXN veri seti için BCO algoritması başarıımı.....	88
Çizelge 4.17.	RDS1-Iris veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	90
Çizelge 4.18.	RDS1-Iris veri seti için BCO algoritması başarıımı	90
Çizelge 4.19.	RDS2-Quail veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	92
Çizelge 4.20.	RDS2-Quail veri seti için BCO algoritması başarıımı.....	92
Çizelge 4.21.	RDS3-Wine veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	94
Çizelge 4.22.	RDS3-Wine veri seti için BCO algoritması başarıımı	94
Çizelge 4.23.	RDS4-Glass veri setinde BCO algoritması ile küme sayısına göre indeks değerleri.....	96
Çizelge 4. 24.	RDS4-Glass veri seti için BCO algoritması başarıımı.....	96
Çizelge 4.25.	SDSC veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	98
Çizelge 4.26.	SDSC veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	99
Çizelge 4.27.	SDSC veri seti için OCO algoritması başarıımı.....	99
Çizelge 4.28.	SDSCN veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	101
Çizelge 4.29.	SDSCN veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	102
Çizelge 4.30.	SDSCN veri seti için OCO algoritması başarıımı.....	102
Çizelge 4.31.	SDSE veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	105
Çizelge 4.32.	SDSE veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	105
Çizelge 4.33.	SDSE veri seti için OCO algoritması başarıımı.....	106

Çizelge 4.34.	SDSEN veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	108
Çizelge 4.35.	SDSEN veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	108
Çizelge 4.36.	SDSEN veri seti için OCO algoritması başarıımı	109
Çizelge 4.37.	SDSR veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	111
Çizelge 4.38.	SDSR veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	111
Çizelge 4.39.	SDSR veri seti için OCO algoritması başarıımı.....	112
Çizelge 4.40.	SDSRN veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	114
Çizelge 4.41.	SDSRN veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	114
Çizelge 4.42.	SDSRN veri seti için OCO algoritması başarıımı.....	115
Çizelge 4.43.	SDSX veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	117
Çizelge 4.44.	SDSX veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	117
Çizelge 4.45.	SDSX veri seti için OCO algoritması başarıımı	118
Çizelge 4.46.	SDSXN veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	121
Çizelge 4.47.	SDSXN veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	121
Çizelge 4.48.	SDSXN veri seti için OCO algoritması başarıımı	122
Çizelge 4.49.	RDS1-Iris veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	124
Çizelge 4.50.	RDS1-Iris veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	124

Çizelge 4.51.	RDS1-Iris veri seti için OCO algoritması başarımı	125
Çizelge 4.52.	RDS2-Quail veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	127
Çizelge 4.53.	RDS2-Quail veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	127
Çizelge 4.54.	RDS2-Quail veri seti için OCO algoritması başarımı.....	128
Çizelge 4.55.	RDS3-Wine veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	129
Çizelge 4.56.	RDS3-Wine veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	130
Çizelge 4.57.	RDS3-Wine veri seti için OCO algoritması başarımı.....	130
Çizelge 4.58.	RDS4-Glass veri setinde OCO algoritması ile küme sayısına göre indeks değerleri.....	133
Çizelge 4.59.	RDS4-Glass veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	133
Çizelge 4.60.	RDS4-Glass veri seti için OCO algoritması başarımı.....	134
Çizelge 4.61.	SDSC veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	136
Çizelge 4.62.	SDSC veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	136
Çizelge 4.63.	SDSC veri seti için BOCO algoritması başarımı	137
Çizelge 4. 64.	SDSCN veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	140
Çizelge 4.65.	SDSCN veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	140
Çizelge 4.66.	SDSCN veri seti için BOCO algoritması başarımı	141
Çizelge 4.67.	SDSE veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	143

Çizelge 4.68.	SDSE veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	143
Çizelge 4.69.	SDSE veri seti için BOCO algoritması başarımı	144
Çizelge 4.70.	SDSEN veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	146
Çizelge 4.71.	SDSEN veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	147
Çizelge 4.72.	SDSEN veri seti için BOCO algoritması başarımı	147
Çizelge 4.73.	SDSR veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	149
Çizelge 4.74.	SDSR veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	150
Çizelge 4.75.	SDSR veri seti için BOCO algoritması başarımı	150
Çizelge 4.76.	SDSRN veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	153
Çizelge 4.77.	SDSRN veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	153
Çizelge 4.78.	SDSRN veri seti için BOCO algoritması başarımı	154
Çizelge 4.79.	SDSX veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	156
Çizelge 4.80.	SDSX veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	156
Çizelge 4.81.	SDSX veri seti için BOCO algoritması başarımı.....	157
Çizelge 4.82.	SDSXN veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	159
Çizelge 4.83.	SDSXN veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	159
Çizelge 4.84.	SDSXN veri seti için BOCO algoritması başarımı.....	160

Çizelge 4.85.	RDS1-Iris veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	162
Çizelge 4.86.	RDS1-Iris veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	162
Çizelge 4.87.	RDS1-Iris veri seti için BOCO algoritması başarıımı	163
Çizelge 4.88.	RDS2-Quail veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	165
Çizelge 4.89.	RDS2-Quail veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	165
Çizelge 4.90.	RDS2-Quail veri seti için BOCO algoritması başarıımı	166
Çizelge 4.91.	RDS3-Wine veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	168
Çizelge 4.92.	RDS3-Wine veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	168
Çizelge 4.93.	RDS3-Wine veri seti için BOCO algoritması başarıımı	169
Çizelge 4.94.	RDS4-Glass veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri.....	171
Çizelge 4.95.	RDS4-Glass veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	171
Çizelge 4.96.	RDS4-Glass veri seti için BOCO algoritması başarıımı	172
Çizelge 4.97.	SDSC veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri.....	174
Çizelge 4.98.	SDSC veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	174
Çizelge 4.99.	SDSC veri seti için OBCO algoritması başarıımı	175
Çizelge 4.100.	SDSCN veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri.....	178
Çizelge 4.101.	SDSCN veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	178

Çizelge 4.102. SDSCN veri seti için OBCO algoritması başarımı	179
Çizelge 4.103. SDSE veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri.....	180
Çizelge 4.104. SDSE veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	181
Çizelge 4.105. SDSE veri seti için OBCO algoritması başarımı	181
Çizelge 4.106. SDSEN veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri.....	184
Çizelge 4.107. SDSEN veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	184
Çizelge 4.108. SDSEN veri seti için OBCO algoritması başarımı	185
Çizelge 4.109. SDSR veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri.....	186
Çizelge 4.110. SDSR veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	187
Çizelge 4.111. SDSR veri seti için OBCO algoritması başarımı	187
Çizelge 4.112. SDSRN veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri.....	189
Çizelge 4.113. SDSRN veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	190
Çizelge 4.114. SDSRN veri seti için OBCO algoritması başarımı	190
Çizelge 4.115. SDSX veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri.....	193
Çizelge 4.116. SDSX veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	193
Çizelge 4.117. SDSX veri seti için OBCO algoritması başarımı.....	194
Çizelge 4.118. SDSXN veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri.....	196

Çizelge 4.119. SDSXN veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	196
Çizelge 4.120. SDSXN veri seti için OBCO algoritması başarımı.....	197
Çizelge 4.121. RDS1-Iris veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri.....	199
Çizelge 4.122. RDS1-Iris veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	199
Çizelge 4.123. RDS1-Iris veri seti için OBCO algoritması başarımı	200
Çizelge 4.124. RDS2-Quail veri setinde OBCO algoritması ile küme sayısına göre indeks değerler.....	202
Çizelge 4.125. RDS2-Quail veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	202
Çizelge 4.126. RDS2-Quail veri seti için OBCO algoritması başarım	203
Çizelge 4.127. RDS3-Wine veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri.....	205
Çizelge 4.128. RDS3-Wine veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	205
Çizelge 4. 129. RDS3-Wine veri seti için OBCO algoritması başarımı	206
Çizelge 4. 130. RDS4-Glass veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri.....	208
Çizelge 4.131. RDS4-Glass veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	208
Çizelge 4.132. RDS4-Glass veri seti için OBCO algoritması başarımı.....	209
Çizelge 4.133. SDSC veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	211
Çizelge 4.134. SDSC veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	211
Çizelge 4.135. SDSC veri seti için İOCO algoritması başarımı	212

Çizelge 4.136. SDSCN veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	215
Çizelge 4.137. SDSCN veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	216
Çizelge 4.138. SDSCN veri seti için İOCO algoritması başarıımı	216
Çizelge 4.139. SDSE veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	219
Çizelge 4.140. SDSE veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	219
Çizelge 4.141. SDSE veri seti için İOCO algoritması başarıımı	220
Çizelge 4.142. SDSEN veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	223
Çizelge 4.143. SDSEN veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	223
Çizelge 4.144. SDSEN veri seti için İOCO algoritması başarıımı.....	224
Çizelge 4.145. SDSR veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	226
Çizelge 4.146. SDSR veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	226
Çizelge 4.147. SDSR veri seti için İOCO algoritması başarıımı	227
Çizelge 4.148. SDSRN veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	229
Çizelge 4.149. SDSRN veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	229
Çizelge 4.150. SDSRN veri seti için İOCO algoritması başarıımı	230
Çizelge 4.151. SDSX veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	232
Çizelge 4.152. SDSX veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri	232

Çizelge 4.153. SDSX veri seti için İOCO algoritması başarımı.....	233
Çizelge 4.154. SDSXN veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	235
Çizelge 4.155. SDSXN veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	235
Çizelge 4.156. SDSXN veri seti için İOCO algoritması başarımı.....	236
Çizelge 4.157. RDS1-Iris veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	238
Çizelge 4.158. RDS1-Iris veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	238
Çizelge 4.159. RDS1-Iris veri seti için İOCO algoritması başarımı.....	239
Çizelge 4.160. RDS2-Quail veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	240
Çizelge 4.161. RDS2-Quail veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	241
Çizelge 4.162. RDS2-Quail veri seti için İOCO algoritması başarımı.....	241
Çizelge 4.163. RDS3-Wine veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	243
Çizelge 4.164. RDS3-Wine veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	244
Çizelge 4.165. RDS3-Wine veri seti için İOCO algoritması başarımı.....	244
Çizelge 4.166. RDS4-Glass veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri.....	247
Çizelge 4.167. RDS4-Glass veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	247
Çizelge 4.168. RDS4-Glass veri seti için İOCO algoritması başarımı.....	248
Çizelge 4.169. SDSC veri setinde UOBDO algoritması ile küme sayısına göre indeks değerleri.....	250

Çizelge 4.170. SDSC veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	250
Çizelge 4.171. SDSC veri seti için UOBCO algoritması başarıımı	251
Çizelge 4.172. SDSCN veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri.....	254
Çizelge 4.173. SDSCN veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	254
Çizelge 4.174. SDSCN veri seti için UOBCO algoritması başarıımı	255
Çizelge 4.175. SDSE veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri.....	257
Çizelge 4.176. SDSE veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	257
Çizelge 4.177. SDSE veri seti için UOBCO algoritması başarıımı	258
Çizelge 4.178. SDSN veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri.....	260
Çizelge 4.179. SDSN veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	260
Çizelge 4.180. SDSN veri seti için UOBCO algoritması başarıımı	261
Çizelge 4.181. SDSR veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri.....	263
Çizelge 4.182. SDSR veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	263
Çizelge 4.183. SDSR veri seti için UOBCO algoritması başarıımı	264
Çizelge 4.184. SDSRN veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri.....	265
Çizelge 4.185. SDSRN veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	266
Çizelge 4.186. SDSRN veri seti için UOBCO algoritması başarıımı	266

Çizelge 4.187. SDSX veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri.....	269
Çizelge 4.188. SDSX veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	269
Çizelge 4.189. SDSX veri seti için UOBCO algoritması başarımı.....	270
Çizelge 4.190. SDSXN veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri.....	272
Çizelge 4.191. SDSXN veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	272
Çizelge 4.192. SDSXN veri seti için UOBCO algoritması başarımı.....	273
Çizelge 4.193. RDS1-Iris veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri.....	274
Çizelge 4.194. RDS1-Iris veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	275
Çizelge 4.195. RDS1-Iris veri seti için UOBCO algoritması başarımı.....	275
Çizelge 4.196. RDS2-Quail veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri.....	278
Çizelge 4.197. RDS2-Quail veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	278
Çizelge 4.198. RDS2-Quail veri seti için UOBCO algoritması başarımı.....	279
Çizelge 4.199. RDS3-Wine veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri.....	281
Çizelge 4.200. RDS3-Wine veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	281
Çizelge 4.201. RDS3-Wine veri seti için UOBCO algoritması başarımı.....	282
Çizelge 4.202. RDS4-Glass veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri.....	284
Çizelge 4.203. RDS4-Glass veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri.....	284

Çizelge 4.204.	RDS4-Glass veri seti için UOBACO algoritması başarımı	285
Çizelge 5.1.	Sentetik gürültüsüz veri setlerinde algoritmalara göre küme sayıları.....	288
Çizelge 5.2.	Sentetik gürültüsüz veri setlerinde algoritmalara göre iterasyon sayıları ve hesaplama süreleri (saniye).....	288
Çizelge 5.3.	Sentetik gürültülü veri setlerinde algoritmalara göre küme sayıları.....	289
Çizelge 5.4.	Sentetik gürültülü veri setlerinde algoritmalara göre iterasyon sayıları ve hesaplama süreleri (saniye).....	289
Çizelge 5.5.	Gerçek veri setlerinde algoritmalara göre küme sayıları	290
Çizelge 5.6.	Gerçek veri setlerinde algoritmalara göre iterasyon sayıları ve hesaplama süreleri (saniye)	290

Şekil 3.1.	Sentetik Veri Setlerinde İlk İki Özelliğin Mevcut Kümelere Göre Dağılımı	27
Şekil 3.2.	Sentetik Veri Setlerinde İlk İki Özelliğin Mevcut Kümelere Göre Dağılımı	28
Şekil 3.3.	Gerçek Veri Setlerinde İlk İki Özelliğin Mevcut Kümelere Göre Dağılımı	30
Şekil 3.4.	X16 Veri Seti P1 Ve P2 Özellik Çifti İçin Serpilme Diyagramı.....	34
Şekil 3.5.	X16 Veri Setinde Farklı Küme Sayılarına Göre BCO Algoritması Kümeleme Sonuçları.....	41
Şekil 3.6.	X16 veri seti için küme sayısına göre OCO algoritması ile kümeleme sonuçları	46
Şekil 3.7.	Gürültü ve aykırı değerlerin temsili gösterimi (Wachs ve ark.,2004)	49
Şekil 4.1.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSC)	74
Şekil 4.2.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSCN)	77
Şekil 4.3.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSE)	79
Şekil 4.4.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSEN)	81
Şekil 4.5.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR)	83
Şekil 4.6.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN)	85
Şekil 4.7.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX).....	87

Şekil 4.8.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSXN).....	89
Şekil 4.9.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS1-Iris).....	91
Şekil 4.10.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail).....	93
Şekil 4.11.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS3-Wine).....	95
Şekil 4.12.	BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS4-Glass).....	97
Şekil 4.13.	OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSC).....	100
Şekil 4.14.	OCO algoritması uzatılmış indeks değerleri (SDSC).....	101
Şekil 4.15.	OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSCN).....	103
Şekil 4.16.	OCO algoritması uzatılmış indeks değerleri (SDSCN).....	104
Şekil 4.17.	OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSE).....	106
Şekil 4.18.	OCO algoritması uzatılmış indeks değerleri (SDSE).....	107
Şekil 4.19.	OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSEN).....	109
Şekil 4.20.	OCO algoritması uzatılmış indeks değerleri (SDSEN).....	110
Şekil 4.21.	OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR).....	112
Şekil 4.22.	OCO algoritması uzatılmış indeks değerleri (SDSR).....	113
Şekil 4.23.	OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN).....	115
Şekil 4.24.	OCO algoritması uzatılmış indeks değerleri (SDSRN).....	116

Şekil 4.25. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX).....	119
Şekil 4.26. OCO algoritması uzatılmış indeks değerleri (SDSX).....	120
Şekil 4.27. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSXN).....	122
Şekil 4.28. OCO algoritması uzatılmış indeks değerleri (SDSXN).....	123
Şekil 4.29. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail).....	125
Şekil 4.30. OCO algoritması uzatılmış indeks değerleri (RDS1-Iris).....	126
Şekil 4.31. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail).....	128
Şekil 4.32. OCO algoritması uzatılmış indeks değerleri (RDS2-Quail).....	129
Şekil 4.33. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS3-Wine).....	131
Şekil 4.34. OCO algoritması uzatılmış indeks değerleri (RDS3-Wine).....	132
Şekil 4.35. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS4-Glass).....	134
Şekil 4.36. OCO algoritması uzatılmış indeks değerleri (RDS4-Glass).....	135
Şekil 4.37. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSC).....	138
Şekil 4.38. BOCO algoritması uzatılmış indeks değerleri (SDSC).....	139
Şekil 4.39. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSCN).....	141
Şekil 4.40. BOCO algoritması uzatılmış indeks değerleri (SDSCN).....	142
Şekil 4.41. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSE).....	145
Şekil 4.42. BOCO algoritması uzatılmış indeks değerleri (SDSE).....	146
Şekil 4.43. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSEN).....	148

Şekil 4.44. BOCO algoritması uzatılmış indeks değerleri (SDSEN).....	149
Şekil 4.45. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR).....	151
Şekil 4.46. BOCO algoritması uzatılmış indeks değerleri (SDSR).....	152
Şekil 4.47. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN).....	154
Şekil 4.48. BOCO algoritması uzatılmış indeks değerleri (SDSRN).....	155
Şekil 4.49. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX).....	157
Şekil 4.50. BOCO algoritması uzatılmış indeks değerleri (SDSX).....	158
Şekil 4.51. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSXN).....	160
Şekil 4.52. BOCO algoritması uzatılmış indeks değerleri (SDSXN).....	161
Şekil 4.53. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS1-Iris).....	163
Şekil 4.54. BOCO algoritması uzatılmış indeks değerleri (RDS1-Iris).....	164
Şekil 4.55. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail).....	166
Şekil 4.56. BOCO algoritması uzatılmış indeks değerleri (RDS2-Quail).....	167
Şekil 4.57. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS3-Wine).....	169
Şekil 4.58. BOCO algoritması uzatılmış indeks değerleri (RDS3-Wine).....	170
Şekil 4.59. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS4-Glass).....	172
Şekil 4.60. BOCO algoritması uzatılmış indeks değerleri (RDS4-Glass).....	173
Şekil 4.61. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSC).....	176
Şekil 4.62. OBCO algoritması uzatılmış indeks değerleri (SDSC).....	177

Şekil 4.63. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSCN)	179
Şekil 4.64. OBCO algoritması uzatılmış indeks değerleri (SDSCN).....	180
Şekil 4.65. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSE)	182
Şekil 4.66. OBCO algoritması uzatılmış indeks değerleri (SDSE).....	183
Şekil 4.67. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSEN)	185
Şekil 4.68. OBCO algoritması uzatılmış indeks değerleri (SDSEN).....	186
Şekil 4. 69. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR).....	188
Şekil 4.70. OBCO algoritması uzatılmış indeks değerleri (SDSR).....	189
Şekil 4.71. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN)	191
Şekil 4.72. OBCO algoritması uzatılmış indeks değerleri (SDSRN).....	192
Şekil 4.73. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX).....	194
Şekil 4.74. OBCO algoritması uzatılmış indeks değerleri (SDSX)	195
Şekil 4.75. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSXN).....	197
Şekil 4.76. OBCO algoritması uzatılmış indeks değerleri (SDSXN).....	198
Şekil 4.77. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS1-Iris).....	200
Şekil 4.78. OBCO algoritması uzatılmış indeks değerleri (RDS1-Iris)	201
Şekil 4.79. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail).....	203
Şekil 4.80. OBCO algoritması uzatılmış indeks değerleri (RDS2-Quail).....	204
Şekil 4.81. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS3-Wine)	206

Şekil 4.82.	OBCO algoritması uzatılmış indeks değerleri (RDS3-Wine).....	207
Şekil 4.83.	OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS4-Glass).....	209
Şekil 4.84.	OBCO algoritması uzatılmış indeks değerleri (RDS4-Glass).....	210
Şekil 4.85.	İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSC).....	213
Şekil 4.86.	İOCO algoritması uzatılmış indeks değerleri (SDSC).....	214
Şekil 4.87.	İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSCN).....	217
Şekil 4.88.	İOCO algoritması uzatılmış indeks değerleri (SDSCN).....	218
Şekil 4.89.	İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSE).....	221
Şekil 4.90.	İOCO algoritması uzatılmış indeks değerleri (SDSE).....	222
Şekil 4.91.	İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSEN).....	224
Şekil 4.92.	İOCO algoritması uzatılmış indeks değerleri (SDSEN).....	225
Şekil 4.93.	İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR).....	227
Şekil 4.94.	İOCO algoritması uzatılmış indeks değerleri (SDSR).....	228
Şekil 4.95.	İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN).....	230
Şekil 4.96.	İOCO algoritması uzatılmış indeks değerleri (SDSRN).....	231
Şekil 4.97.	İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX).....	233
Şekil 4.98.	İOCO algoritması uzatılmış indeks değerleri (SDSX).....	234
Şekil 4.99.	İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSXN).....	236
Şekil 4.100.	İOCO algoritması uzatılmış indeks değerleri (SDSXN).....	237

Şekil 4.101. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS1-Iris).....	239
Şekil 4.102. İOCO algoritması uzatılmış indeks değerleri (RDS1-Iris).....	240
Şekil 4.103. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail)	242
Şekil 4.104. İOCO algoritması uzatılmış indeks değerleri (RDS2-Quail)	243
Şekil 4.105. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS3-Wine)	245
Şekil 4.106. İOCO algoritması uzatılmış indeks değerleri (RDS3-Wine)	246
Şekil 4.107. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS4-Glass)	248
Şekil 4.108. İOCO algoritması uzatılmış indeks değerleri (RDS4-Glass)	249
Şekil 4.109. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSC)	252
Şekil 4.110. UOBCO algoritması uzatılmış indeks değerleri (SDSC).....	253
Şekil 4.111. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı(SDSCN)	255
Şekil 4.112. UOBCO algoritması uzatılmış indeks değerleri (SDSCN).....	256
Şekil 4.113. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSE)	258
Şekil 4.114. UOBCO algoritması uzatılmış indeks değerleri (SDSE).....	259
Şekil 4.115. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSEN)	261
Şekil 4.116. UOBCO algoritması uzatılmış indeks değerleri (SDSEN).....	262
Şekil 4.117. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR)	264
Şekil 4.118. UOBCO algoritması uzatılmış indeks değerleri (SDSR).....	265
Şekil 4.119. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN)	267

Şekil 4.120. UOBCO algoritması uzatılmış indeks değerleri (SDSRN).....	268
Şekil 4.121. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX).....	270
Şekil 4.122. UOBCO algoritması uzatılmış indeks değerleri (SDSX).....	271
Şekil 4.123. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSXN).....	273
Şekil 4.124. UOBCO algoritması uzatılmış indeks değerleri (SDSXN).....	274
Şekil 4.125. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS1-Iris).....	276
Şekil 4.126. UOBCO algoritması uzatılmış indeks değerleri (RDS1-Iris)	277
Şekil 4.127. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail)	279
Şekil 4.128. UOBCO algoritması uzatılmış indeks değerleri (RDS2-Quail).....	280
Şekil 4.129. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS3-Wine)	282
Şekil 4.130. UOBCO algoritması uzatılmış indeks değerleri (RDS3-Wine)	283
Şekil 4.131. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS4-Glass)	285
Şekil 4.132. UOBCO algoritması uzatılmış indeks değerleri (RDS4-Glass).....	286

SİMGELER VE KISALTMALAR

VM	: Veri Madenciliği
KA	: Kümeleme Analizi
KO	: K-Ortalamlar
BCO	: Bulanık C-Ortalamlar
İOCO	: İtmeli Olabilirlikli C-Ortalamlar
BOCO	: Bulanık Olabilirlikli C-Ortalamlar
OCO	: Olabilirlikli C-Ortalamlar
KDE	: Kernel Density Estimation (Kernel'in Yoğunluk Tahmini)
SMV	: Support Vector Clustering (Destek Vektör Kümeleme)
GK-OBCO	: Gustafson-Kessel Olabilirlikli Bulanık C-Ortalamlar
NBRIP:	: National Botanical Research Institute's phosphate growth medium
PE	: Bölümleme Entropisi
PC	: Bölümleme Katsayısı
XB	: Xie-Beni İndeksi
FS	: Bulanık Silliuet İndeksi
HBCO	: Hızlı Bulanık C-Ortalamlar
GOBCO	: Genelleştirilmiş Olabilirlikli Bulanık C-Ortamalar
AU-OCO	: Aralık Uyumlu C-Ortalamlar
BBOCO	: Benzer Bulanık Olabilirlikli C-Ortalamlar
AD-OBCO	: Aralık Değerli Olabilirlikli Bulanık C-Ortalamlar
KBCO	: Kernel'in Bulanık C-Ortalaması
KOCO	: Kernel'in Olabilirlikli C-Ortalamlar
AD-BCO	: Aralık Değerli Bulanık C-Ortalamlar
BYBCO	: Yerel Bilgi C-Ortalamlar
AD-OCO	: Aralık Değerli Olabilirlikli C-Ortalamlar Bulanık
MBOCO	: Modifiye Edilmiş Bulanık Olabilirlikli C-Ortalamlar

MPCAES	: Modified Partition Coefficient And Exponential Separation
TK-OCO	: Tek Kümeli Olabilirlikli C-Ortalamlar
BrBCO	: Br Bulanık C-Ortalamlar
SDSC	: Daire şekilli – Gürültüsüz Yapay Veri Seti
SDSCN	: Daire şekilli – Gürültülü Yapay Veri Seti
SDSE	: Elips şekilli – Gürültüsüz Yapay Veri Seti
SDSEN	: Elips şekilli – Gürültülü Yapay Veri Seti
SDSR	: Kare şekilli – Gürültüsüz Yapay Veri Seti
SDSRN	: Kare şekilli – Gürültülü Yapay Veri Seti
SDSX	: Konkav şekilli – Gürültüsüz Yapay Veri Seti
SDSXN	: Konkav şekilli – Gürültülü Yapay Veri Seti
RDS1-Iris	: İris Veri Seti
RDS2-Quail	: Quail Veri Seti
RDS3-Wine	: Wine Veri Seti
RDS4-Glass	: Glass Veri Seti
X16	: 16 Elemanlı Sentetik Veri Seti
İOBCO	: İtmeli Olabilirlikli Bulanık C-Ortalamlar Algoritması
UOBCO	: Yönetimsiz Olabilirlikli Bulanık C-Ortalamlar Algoritması
OKA	: Olabilirlikli Kümeleme Algoritması
MPC	: Değiştirilmiş Bölümleme Katsayısı
TS	: Tang ve Sun İndeksi
EXB	: Uzatılmış Xie-Beni İndeksi
EKwon	: Uzatılmış Kwon İndeksi
ETS	: Uzatılmış Tang ve Sun İndeksi
EXB	: Uzatılmış Xie-Beni İndeksi
EPE	: Uzatılmış Bölümleme Entropisi
η	: Eta vektörü
ε	: Yaklaşma değeri
γ	: İtme teriminin tartı (ağırlık) faktörü

s : Hesaplama süresi
T : İterasyon sayısı
C : Küme sayısı
N : Nesne sayısı
P : Boyut veya özellik sayısı

1. GİRİŞ

Son yıllarda otomatik veri toplama araç ve sistemlerinin yaygın kullanımı nedeniyle tarım ve çevre verilerinin hacminde çok yüksek artışlar gözlenmektedir. Hava durumu, uzaktan algılama ve coğrafi bilgi sistemleriyle toplanan yüksek hacimli veri uzun zamandan beri kullanımda bulunmaktadır. Hassas tarım uygulamalarından elde edilen verim kayıt verileri tarımsal veri hacminde artışı çok daha hızlandırmıştır. Bunlara ilaveten bitkisel üretim alanlarında, ahır ve kümes gibi hayvansal üretim tesislerinde duygular ve duyurga ağlarının yoğun kullanımı da tarımsal veri artışında çok önemli artışlara yol açmıştır. Tarımsal veri hacminin bu olağanüstü artışı diğer birçok alanda olduğu gibi tarımda da “büyük veri” oluşmasını sağlamıştır. Tarımsal büyük verideki anlamlı bilgi veya yapıları keşfetmek için veri madenciliği (VM) yöntemleri ve tekniklerinin kullanılmasına ihtiyaç ortaya çıkmıştır. Buna karşın, tarım, gıda, çevre ve diğer ilgili alanlarda VM henüz yeterince tanınmamakta ve uygulanmamaktadır. Ancak, birçok VM tekniği arasında en popüler olanlardan biri olarak kümeleme analizi örüntü tanıma, görüntü bölümlenme, genetik, biyoinformatik, web madenciliği ve tüketici pazar araştırmaları gibi diğer birçok alandakine benzer şekilde tarımsal büyük veri madenciliğinde de kullanılmaktadır. Örneğin, genelde VM ve özelde kümeleme analizi (KA) tarımda karar destek sistemlerinin temel bileşenlerinden biri durumundadır (Tiwari ve Misra, 2011).

Bir şemsiye terim olarak kümeleme analizi, veri setlerindeki nesnelere gruplara ayırma veya veri setlerini küme olarak adlandırılan alt veri setlerine ayırma/bölme işlemleri veya teknikleri koleksiyonudur. İstatistiksel olarak kümeler, uygun bir kümeleme algoritması kullanılarak birbirine daha fazla benzeyen nesnelere gruplanmasıyla oluşturulur. Bir başka ifadeyle özellikleri itibarıyla aynı karakteristikleri gösteren nesnelere aynı kümeye atanırlar. Birçok farklı açıdan kategorize edilebilmekle birlikte, kümeleme algoritmalarını hiyerarşik, hiyerarşik olmayan (düz) ve karma yöntemler/teknikler olmak üzere 3

grup altında incelemek mümkündür. Yüzlerce kümeleme algoritması mevcut olmasına karşın pratikte bu algoritmalarından çoğunun kullanımı, karmaşıklık, etkinlik ve istatistik yazılımlarda mevcut olup olmadığına bağlı olarak sınırlı durumdadır.

Kümeleme analizinde iyi bir algoritma seçimi üzerinde çalışılan verinin hacmi, yapısı ve elbette kümelemenin amacına bağlı olarak değişmektedir (Velmurugan, 2012; Bora ve Gupta, 2014). Birçok çalışmada da bildirildiği gibi (örneğin Dong ve ark., 2011; Kuar ve Kuar, 2013) büyük veri setlerinde hiyerarşik yöntemlerle karşılaştırıldığında hiyerarşik olmayan bölümleyici kümeleme algoritmaları, örneğin K-ortalamlar ailesi algoritmalar, daha az işlem zamanı ile etkin ve iyi sonuçlar verebilmektedir. Bunun için K-ortalamlar, MacQueen (1967) tarafından önerildiğinden bu yana, VM uygulamaları ve açıklayıcı veri analizinde en yoğun kullanılan algoritmalarından biri olmuştur.

K-Ortalamlar (veya yumuşak Bulanık C-Ortalamların önerilmesinden sonra Sert C-Ortalamlar olarak da bilinmektedir), bir veri setini c (veya k) sayıda kümeye bölmek üzere kullanılmaktadır. K-ortalamlar (KO), veri setini k kümeye bölen bir algoritma olduğundan k değerinin önceden bilinmesi gereklidir. Birçok çalışmada bildirildiği üzere eğer veri setindeki kümeler iyi ayrılmışlarsa çok iyi sonuçlar verebilmektedir. Ayrıca işlem maliyeti açısından da göreceli olarak etkin olduğu bilinmektedir. Bu avantajlarına rağmen KO, veri setlerindeki kümelerin şekilleri ve dağılımlarından kaynaklanan birçok dezavantaja da sahiptir. Her şeyden önce KO üst üste çakışan yani iyi ayrılmamış kümelerin bulunmasında başarılı olmadığı gibi doğrusal olmayan transformasyonlara karşı da duyarlıdır. Bu nedenle, aynı veri setinin kartezyen ve kutupsal koordinatları ile temsil edilmesi halinde farklı kümeleme sonuçları ortaya çıkabilmektedir. Ayrıca gürültülü ve doğrusal olmayan dağılımlar gösteren kümeler içeren veri setlerinde de çok başarılı olamamaktadır.

KO ile karşılaşılan bu sorunları çözmek üzere, Bezdek (1981), Dunn'un çalışmasına (Dunn, 1973) dayanan ve KO'nun bir uzantısı olan Bulanık C-

Ortalamalar (BCO) adında yeni bir yumuşak kümeleme algoritması önermiştir. Suganya ve Shanthi (2012) ve Ali ve ark. (2008) tarafından yapılan incelemelerde BCO'nun etkinliğini arttırmak ve karşılaşılan bazı sorunları çözmek üzere bir dizi yeni algoritma daha önerilmiştir. Buna karşın BCO algoritması mühendislikten iktisada birçok disiplinde sıkça kullanılmaya başlanmıştır. BCO, bulanık veri analizini hedefleyen bir yumuşak kümeleme algoritmasıdır (Cebeci ve Yıldız, 2015). Yumuşak kümelemede nesnelere yalnız belli bir kümenin üyesi değil fakat aynı zamanda farklı üyelik dereceleriyle birden fazla kümenin üyesidirler. Bu şekilde, kümelerin sınırlarında bulunan nesnelere illa da belli bir kümeye üye olması zorlanmamakta, aksine 0 ve 1 arasında değişen derecelerle birçok kümeye üye olması söz konusu olmaktadır. KO'ya göre daha yüksek bir hesaplama maliyeti olan BCO, yukarıda anılan avantajları nedeniyle birçok disiplinde olduğu gibi tarım ve ormancılık alanında da uygulama alanı bulmuştur (di Martino ve ark., 2007; 2009).

BCO'nun bulanık veri analizinde daha etkin olduğu ortaya konulmakla birlikte araştırma bulguları tüm veri setleri için bu üstünlüğün aynı kalmadığını ortaya koymaktadır. Zira BCO algoritması ile yapılan analizlerde BCO'nun verilerdeki gürültü ve aşırı değerlere aşırı derece duyarlılık gösterdiği anlaşılmıştır. Bu soruna çözüm arayan algoritmalar geliştirilmiş ancak en çok ilgi çeken Krishnapuram ve Keller (1993)'in Olabilirlikli C-ortalamalar (OCO) algoritması olmuştur. OCO algoritması kümelere üyelik dereceleri yerine kümelere uyumluluk ya da ait olabilirliği gösteren tipiklik değerlerini üreten bir algoritmadır. OCO'nun farklı başlangıç küme merkezlerine farklı tepkiler vermesi nedeniyle örtüşen kümeler (üst üste çakışan) sorununa sahip olduğu görülmüştür. Bu durumda kümelerin bazıları veya hepsi üst üste çakışabilmekte ve bazıları da boş kalabilmektedir. BCO'nun gürültü ve aykırı değerlere karşı sorunlu olması ve OCO'nun özdeş kümeler üretme sorununu çözmek üzere Bulanık Olabilirlikli C-Ortalamalar (BOCO) ve Olabilirlikli Bulanık C-Ortalamalar (OBKO) gibi iki karma algoritma daha geliştirilmiştir. Ancak BOCO'nun büyük verilerde çok küçük

tipiklik dereceleri üretebildiği ve OBCO'nun ise seçilen parametrelere yüksek ölçüde bağlı sonuçlar üretme sorunu gösterdiği anlaşılmıştır. Amaç fonksiyona bir itme (repulsion) terimi eklenerek geliştirilen İtmeli Olabilirlikli C-Ortalamlar (İOCO) algoritması bu soruna çözüm arayan algoritmalar arasında yer almıştır (Timm ve ark., 2001). iOCO ile BCO'nun iyi ayırma gücü ile OCO'nun gürültü ve aşırı değer sorununa çözüm gücü birleştirilerek daha iyi kümeleme başarımı sağlanmıştır (Shapira ve Wachs, 2004). Tüm bunlara ilaveten algoritmaların çoğunda uzaklık ölçüsü olarak kareli Öklid uzaklıklarının kullanılması nedeniyle yuvarlak kümelerde sağlanan başarımların diğer elips ve diğer şekilli kümeler için iyi çalışmadıkları ortaya konulmuş ve Gustafson-Kessel (GK) algoritması bu soruna çözüm aramıştır. Bu algoritma kullanılarak bulanık ve olabilirlikli karmaları da geliştirilmiştir. Örneğin, GK-BOCO'nun üyelik ve tipiklik derecelerine daha fazla önem verme kabiliyetinde olan iyi bir algoritma gibi gözüktüğü bildirilmiştir (Almeida ve Sousa, 2006). Özellikle hız başarımı üzerine yoğunlaşan algoritmalar da geliştirilmiş olup Hızlı Bulanık C-Ortalamlar (HBCO) bunlar arasında yer almaktadır. Almeida ve Sousa (2006) göre HBCO'nun algoritmasının hızlı ve özellik seçimi için iyi bir algoritma olduğu ancak hem altı adet kullanıcı tanımlı parametre gerektirmesi ve hem de parametrelerin ilişkili olması halinde etkinliğinin kesin olmaması gibi dezavantajları bulunmaktadır.

Sonuç olarak bulanık ve olabilirlikli algoritmalar ve bunların farklı karmalarına dayanan çok sayıda algoritma ve teknik geliştirilmesine karşın yapılan testler bunlardan birinin diğerine her durumda üstün olduğunu ortaya koymamıştır. Çünkü algoritmaların etkinliği kullanılan verinin hacmi, veride mevcut kümelerin hacmi, yönü ve şeklinden etkilenmektedir. Ayrıca başlangıç küme merkezleri ve üyelik dereceleri matrisleri ile bulanıklaştırma (fuzzyfier) ve tipikleştirme (typicality) katsayılarının optimal seçimleri kümeleme başarımı üzerinde etkili olmaktadır. Literatürde genellikle KO ve BCO'nun R'da Iris ve Wine gibi bazı iyi tanınmış deneysel veri setlerinde karşılaştırılması üzerine yoğunlaşmış bulunmaktadır (Jipkate ve Gohokar, 2012; Panda ve ark, 2012; Ghosh ve Dubey,

2013; Bora ve Gupta, 2014). Bu yüzden, sert ve yumuşak kümeleme yöntemlerinin farklı büyüklük ve desenler içeren veri setlerinde test edilerek karşılaştırılmasında daha kapsamlı çalışmalara ihtiyaç bulunmaktadır. Bu nedenle, KO, BCO, OCO ve bunların çeşitli karma algoritmalarının (BOCO, OBCO ve iOCO) yapay olarak üretilen farklı küme dağılımları ve küme büyüklerinde gürültülü ve gürültüsüz verilerde test edilmesi veri yapılarına özel önerilerin geliştirilmesinde yararlı olabilecektir.

Bu tez çalışmasında, gürültülü ve gürültüsüz kümeler içeren veri setlerinde sert ve yumuşak bölümleyici kümeleme algoritmalarının başarımları karşılaştırılmaktadır. Klasik K-ortalamlar, Bulanık C-Ortalamlar, Olabilirlikli C-Ortalamlar ile bunların çeşitli karma yaklaşımlarını içeren Bulanık Olabilirlikli C-Ortamalar, Olabilirlikli Bulanık C-Ortalamlar ve İtmeli Olabilirlikli C-Ortalamlar algoritmaları, Yönetimsiz Olabilirlikli Bulanık C-Ortalamlar incelenmekte ve kümeleme etkinlikleri belirlenmektedir. Gürültülü ve gürültüsüz olmak üzere kare, daire ve elips şekilli ve bunların çeşitli düzeyde karışımlarından oluşan kümeler içeren yapay veri setleri üzerinde algoritmaların avantaj ve dezavantajları sınanmaktadır. Bu amaçla, her bir algoritma ile elde edilen sonuçların çeşitli iç indekslere göre değerlendirilerek bazı önerilerin yapılması amaçlanmaktadır.

2. ÖNCEKİ ÇALIŞMALAR

Bezdek ve Ark. (1984), çeşitli jeostatistiksel veri analizlerinin problemlerinde uygulanabilen, sayısal verilerin herhangi bir seti için bulanık bölümlenmeler ve prototipler üreten ve bilinen alt yapıları doğrulamak veya keşfedilmemiş verilerde altyapı önermek için kullanılan yöntemlerden birisi olan Bulanık C-Ortalamalar (BCO) kümeleme algoritmasından bahsedilmiştir. Bu kümeleme yönteminin özellikleri arasında, üç farklı normun (Öklid, Diyagonal veya Mahalonobis) seçilebildiğinden söz edilmektedir. Bunlar; gürültüye duyarlılık, değişken kümelerin kabulü ve kümenin geçerliliğini içeren çeşitli çıktıları kontrol eden ayarlanabilir ağırlık faktörleridir. Bu açıklamalardan sonra sayısal uygulama için ilk olarak Sneath ve Sokal (1973)' in çalışmasında ve Bezdek (1974)'in konuyla ilgili olan makalesinde sıkça kullanılan yapay veri seti üzerinde bir çalışma yapılmaktadır. Ayrıca çalışmada Bezdek (1981)'in konuyla ilgili kitabından da birçok örneğin değerlendirmesiyle birlikte, BCO kümeleme algoritmasının güvenilir bir algoritma olduğu, araştırma çalışmalarıyla birlikte desteklenmektedir.

Pal ve Bezdek (1995), BCO kümeleme algoritması ile üretilen kümeleme başarımlarını değerlendirmek için birçok işlem önermektedir (bu çalışmada bunlar arasında en önemlilerden birisi olan BCO modelinin ağırlıklandırma üssü (m) incelenmektedir). Çalışmada, Xie-Beni (XB) ve uzatılmış Xie-Beni (EXB), Fukuyama-Sugeno (FS), bölümlenme katsayısı (PC) ve bölümlenme entropisi (PE) geçerlilik indeksleri dikkate alınmaktadır. Yazar, düşük ve yüksek değerli m değerini açıkladıktan sonra uygulama bölümünde değerleri elde edip karşılaştırmaktadır. FS endeksinin m 'nin hem yüksek hem de düşük değerlerine karşı duyarlı olduğunu ve bu yüzden sağlam bir indeks olmadığını düşünmektedir. Test edilen indekslerden XB indeksi, kümeler için geniş bir seçim aralığı ve m için en iyi sonucu rakamlarla ifade etmekte ve son olarak, hesaplamalara göre m için en iyi seçimin değer aralığını da ifade etmektedir.

Rezaee ve Ark. (1998), BCO algoritması ile üretilen kümeleme sonuçlarının değerlendirilmesinde kullanılmak üzere yeni bir geçerlilik indeksi önermiştir. Önerilen yeni geçerlilik indeksi, dört farklı geçerlilik indeksi (PE, XB, PC, Fsil) ile karşılaştırılmış ve yeni geçerlilik indeksinin daha iyi sonuç verdiği belirtilmiştir.

Eschrich ve Ark. (2001), BCO algoritmasının görüntü işlemedeki önemini vurgulayarak, BCO'nun daha hızlı çalışma başarımını gösteren yeni bir varyant olan BCO algoritmasını önermişlerdir. Br Bulanık C-Ortalamlar (BrBCO) olarak adlandırılan algoritma, 172 kızılötesi görüntüden alınan veri seti üzerinde orijinal BCO algoritmasına göre daha hızlı kümeleme sonucu üretmiş ancak veri indirgeme çok fazla olduğundan kümeleme başarımı BCO'ya göre azalmıştır.

Halkidi ve Ark. (2001), kümeleme sonuçlarının değerlendirilmesinde kullanılan iç ve dış geçerlilik indekslerini kapsamlı olarak incelemişlerdir. Araştırmacılar ayrıca mevcut geçerlilik indekslerinin başarımlarını bazı veri setleri üzerinde yapılan analiz çalışmaları ile karşılaştırmalı olarak değerlendirmişlerdir. Genel olarak geçerlilik indekslerinin, kümeler çoğunlukla kompakt olduğunda daha iyi çalıştıklarını, ancak farklı şekiller ve yerleşimler gösteren kümelerin değerlendirilmesinde yeterli olmadıklarını ileri sürmüşlerdir.

Pakhira ve Ark. (2003), "PBM" olarak adlandırılan yeni bir geçerlilik indeksinin başarımını, sert kümeleme algoritması olarak K-ortalamlar (KO) ve bulanık kümelemeyi temsilen BCO algoritmasını kullanarak test etmişlerdir. Davies Bouldin indeksi, Dunn indeksi ve XB indeksi gibi iyi bilinen geçerlilik indekslerine karşı yapay ve gerçek veri setleri üzerinde yapılan karşılaştırmalara göre PBM indeksinin kümelerin sayısını diğerlerinden daha iyi tahmin ettiği gösterilmiştir.

Shapira ve Wachs (2004), Olabilirlikli C-Ortalamlar (OCO) algoritmasında zayıf başlatmalar durumunda ortaya çıkan çakışık kümeler sorununu çözmek üzere, İtmeli Olabilirlikli C-Ortalamlar (İOCO) kümeleme algoritmasını önermişlerdir. İki veri kümesi üzerinde BCO, OCO, OBCO ve İOCO

algoritmalarının başarımları karşılaştırılmıştır. Çalışmada İOCO'nun, diğer algoritmalara kıyasla daha iyi kümeleme sonuçları verdiği ortaya konulmuştur.

Wu ve Yang (2004), "PCAES" olarak adlandırılan yeni bir geçerlilik indeksi önermişlerdir. Önerilen indeksin etkinliği üç farklı gerçek veri seti (Iris, Glass, Vowel) üzerinde test edilmiş ve uygun küme sayısını tahmin etmede daha başarılı olduğu ileri sürülmüştür. Ayrıca gürültülü veri setlerinden elde edilen kümeleme sonuçlarının değerlendirilmesinde daha uygun tahmin edicilerden söz edilmiştir.

Wachs ve Ark. (2004), bu çalışmada temel görüntü tanımlama tekniklerinden birisi olan bulanık kümeleme algoritmalarını incelemektedirler. Genel olarak OCO'nun gürültülü noktalar üzerinde BCO'ya göre daha iyi performans gösterdiğinden bahsedilmektedir. Ancak OCO'nun rastgele kümeler üretmek konusunda istenmeyen bir eğilim içerisinde olduğu bu yüzden de BCO'nun bölme özelliğini OCO'nun sağlam gürültü duyarsızlığı ile birleştirilerek bir itme teriminin eklenmesiyle İOCO algoritmasının başarımlarını değerlendirilmektedir. Ayrıca veri setlerine uygulanan algoritmaların başarımlarını test etmek için bazı geçerlilik indeksleri de uygulanmaktadır. Önerilen algoritma (İOCO) hesaplama açısından yoğun olsa da, kümelenebilirliği denetlemekte daha iyidir. XB indeksi sadece kompakt ve ayrı kümeleri ölçer, bu nedenle güvenilirlik için daha iyi geçerlilik önlemleri dikkate alınarak iyileştirilebilir.

Tang ve Ark. (2005), bulanık kümeleme için önerilen XB ve Kwon geçerlilik indekslerinde küme sayısının artışıyla oluşan monotonik azalma problemini önlemek için yeni bir geçerlilik indeksi önermişlerdir. Önerilen indeks, UCI Machine Learning Repository'deki Butterfly ve Iris veri setleri üzerine uygulanıp diğer bulanık indekslerle karşılaştırılmış ve önerilen indeksin çalışılan veri setleri için diğerlerine göre daha iyi sonuçlar verdiği ortaya konulmuştur.

Cano ve Ark. (2007), bu çalışmada potansiyel olarak örtüşen Olabilirlikli Spektral Kümeleme (OSK) algoritması adında, bulanıklık teknolojisi ile spektral kümeleme bazlı yeni bir metot önermektedirler. Elde edilen metot, Saccharomyces

cerevisiae (bir maya türü) hücrenel döngü ifadesi verileri ve kanserli insanlardan alınan veri setleri üzerinde test edilmektedir. Çalışmada yapılan analizde gen keşfi, genlerin fonksiyonel açıklaması, hastalık tanısı, uyuşturucu keşfi ve tümör alt tiplerinin tespiti gibi durumlar hedeflenmektedir. Çift kümeleme mikrodizi verileri için önerilen algoritma (OSK) araştırma sonucu tablo ve grafik halinde gösterilmektedir. Araştırma sonucunda önerilen algoritma CC'nin ve FLOC'un olasılık algoritmalarından alınan sonuçlara göre daha iyi oldukları gözlenmektedir.

Filippone ve Ark. (2007), bu çalışmada da iki farklı kernel metodlarından olan OCO ve Tek Kümeli Olabilirlikli C-Ortalamalar (TK-OCO) algoritmalarını önermektedirler. Bu algoritmalar Olabilirlikli kümeleme özelliklerini, yoğunluk tahmin edicilerini ve aykırı veriler üzerinde sağlamlık göstermek gibi durumlarını muhafaza etmektedirler. Genel olarak çalışmada Olabilirlikli kümeleme algoritmalarının ana yönlerinden bahsedilmekte ve önerilen algoritmalarla yapılan çalışma sonuçlarıyla değerlendirilmektedir. Yapılan çalışmada yapay veri seti üzerinde üç ayrıık yoğun bölge oluşmaktadır. Oluşan bu bölgelerde kare, dikdörtgen ve daire şeklinde desenler oluşmaktadır. Oluşan bu desenler içerisinde farklı sayı ve oranlarda örüntüler barınmaktadır. Sonuç itibarıyla araştırma sonucunda önerilen algoritmaların (OCO ve TK-OCO) ve bu algoritmalarındaki adımların (içini çıkarma, etiketleme, durulaştırma) çok hızlı bir şekilde uygulanmasıyla birlikte optimal değerleri bulma konusunda da iyi sonuçlar verdiği gözlenmektedir.

El-Melegy ve Ark. (2007), bu çalışmada bulanık ve sert kümeleme algoritmalarıyla elde edilen kümeler arasından en iyisini seçmek için, iki yeni geçerlilik indeksi önermektedirler. Çalışmada, daha önceden bilinen indeksler ile yeni indeksler, MR verileri ile birkaç sentetik görüntü üzerinde değerlendirilmekte ve karşılaştırılmaktadırlar. Veri setleri üzerinde önerilen iki yeni ve 16 farklı geçerlilik indeksi ile birlikte toplam 18 indeks incelenip karşılaştırılmaktadır. Genel olarak gözlemlendiğinde, belli bir indeksin her durumda iyi sonuç vereceği kanaatine varılamamaktadır. Çalışmada en iyi indeksin "T" geçerlilik indeksi olduğu gözlenirse de değişken gürültülü durumlar altında MPC, XB, AIC ve SC

indekslerinin daha iyi olduğu gözlenmektedir. Yapılan çalışmanın ayrıca daha iyi indeksler geliştirme amaçlı bir yol gösterdiğinden de bahsedilmektedir.

Zhang ve Ark. (2007), bu çalışmada daha öncesinde Bezdek tarafından önerilen Bulanık Olabilirlikli C-Ortalamlar (BOCO) kümeleme algoritması tarafından türetilen Benzer Bulanık Olabilirlikli C-Ortalamlar (BBOCO) kümeleme algoritmasından bahsetmektedirler. Temel olarak çalışmada önerilen algoritma ile BOCO kümeleme algoritma arasındaki farklar üzerinde durulmaktadır. Mekanizmalı öğrenme metodlarından biri olan kümeleme yöntemlerinde pratik ve teorik anlamda birçok problemle karşı karşıya kalınmaktadır. Karşılaşılan bu problemler (gürültülü noktaları, aykırı veriler gibi) daha sert ve yumuşak kümeleme algoritmaları baz alınarak türetilen algoritmalar ile giderilmeye çalışılmaktadır. Çalışmanın kalan kısmında önerilen algoritmanın tanımından ve UCI mekanizma öğrenme deposundan alınan 22 veri seti üzerine önerilen algoritma ile BOCO kümeleme algoritmasının uygulamasından bahsedilmektedir. Son olarak da uygulamada BBOCO ile BOCO algoritmaları karşılaştırılmaktadır. Karşılaştırma için iki farklı nesne belirtilmekte ve uzantı olarak öklid uzantısı kullanılmaktadır. Belirlenen iki nesne 4 durum halinde açıklanmaktadır. Oluşturulan durumlar sonuçların doğruluğu açısından formüle edilerek (R) en iyi sonucun nasıl oluşacağı şeklinde fikir verilmekte ve karşılaştırılması istenilen algoritmaların sonuçları da tabloda gösterilmektedir. 22 veri seti üzerinden alınan sonuçlara göre BBOCO ile BOCO kümeleme algoritmalarının arasında çok büyük bir fark olmadığı ancak BBOCO algoritmasının diğer algoritmaya göre bir nebze daha iyi sonuç verdiği ve biraz daha hızlı olduğu gözlenmektedir.

Zhang ve Ark. (2008), bu çalışmasında BCO ve OCO kümeleme algoritmaları için geliştirilen geçerlilik indekslerini farklı veri setleri üzerine uygulayıp karşılaştırmaktadırlar. İlk olarak BOCO' nun uygun küme sayısını belirlemek için XB ve Kwon indeksleri genişletilmektedir. Sonrasında önerilen geçerlilik indeksi, Iris ve üç farklı amaca uygun oluşturulmuş diğer veri setiyle

birlikte küme sayılarına göre ayrılıp sonuçlar elde edilmektedir. Geçerlilik indekslerin uygulanmasıyla alınan sonuçlar ile birlikte her bir veri seti için en uygun küme sayısı verilen tabloda gösterilmektedir. Tabloda görülen değerler arasında en küçük indeks değerine sahip küme sayısı, uygun küme sayısını vermektedir. Önerilen EPESIM geçerlilik indeksinin, sadece üyelik matrisleri ve olabirlikli matrislerinin sonuçlarının elde edilmesinde kullanıldığından, bütün veri noktaları ve küme merkezleri arasındaki zaman tüketiminin hesaplaması hakkında bilgi vermediğinden bununla birlikte bulanık iki küme arasındaki yakınlığın daha farklı şartlar altında alternatif metotların araştırılacağından bahsetmektedir.

Filippone ve Ark. (2009), denetimsiz öğrenme, mekanizmalı öğrenme alanındaki etkisiz verilerin problemlerinin analizi konusunda önemli bir dal olduğu belirtilmektedir. Bu çalışmada Olabirlikli kümelemenin Kernel uzantısından bahsedilmektedir. Büyük boyutlu bir uzay içerisinde örnekler tanımlanmakta ve bu oluşturulan yeni uzaya OCO ile bir model oluşturulmaktadır. Uzayda modellenen üyeler ile uzayın içerisinde genel şekilli modeller üretilmektedir. Araştırmada önerilen algoritmalar ile bazı uygulamalar analiz edilmektedir. İlk olarak örneğin yoğunluk tahmini ve kümelemesi gösterilmekte, daha sonra aykırı veriler belirlenmektedir. Son olarak 3 adet gerçek veri üzerinde OCO, SMV ve KDE' nin kararlılığı ve doğruluğu karşılaştırılmaktadır. Yapılan analizler sonucunda OCO' nun aykırı veriler üzerindeki başarımı ve parametrik olmayan yol ile oluşturduğu yoğunluk nedeniyle diğer algoritmalara kıyasla daha çok önerilmektedir. KDE'de örnekler arası farklılıklar, SVM' de ise optimizasyon problemleri arası benzerlik gözlenmektedir. Bu temel nedenler baz alındığından ve daha sağlıklı sonuçlar alındığından OCO kümeleme algoritması tercih edilmektedir.

Saad ve Alimi (2009), bu çalışmada daha önceden Krishnampuram ve Keller tarafından ortaya konulan BCO ve Pal tarafından geliştirilen BOCO kümeleme algoritmalarından yola çıkarak, Modifiyeli Bulanık Olabirlikli C-ortalama (MBOCO) kümeleme algoritmasını önermektedirler. İlk olarak MBOCO 2 küme veri seti üzerinde daha sonra 3 küme veri seti üzerinde gürültülü

ve gürültüsüz üyeler ile test edilmektedir. Önerilen algoritma daha önceden bilinen ve tanımlanan algoritmalar olan BCO ve BOCO algoritmaları ile veri setleri üzerine uygulanıp sonuçları karşılaştırılmaktadır. Yapılan tüm deneyler sonucunda geliştirilen ve önerilen algoritma MBOCO diğer (BCO ve BOCO) algoritmalara göre daha iyi sonuçlar vermektedir. Elde edilen sonuçlar şekiller üzerinde gösterilmektedir. Önerilen algoritmanın amaç fonksiyonu her kümeye ilişkin veri noktalarına yeni ağırlık ekleyerek ve bir nokta ve sınıf arasındaki mesafenin üs değiştirilerek hassaslığı azaltmaya yönelik bir algoritma geliştirmektedir. Sonuç olarak ise BCO ve BOCO' nun uygulanan sayısal örnekler sonucu önerilen algoritmaya göre daha çok farklılık göstermektedir. MBOCO ile alınan sonuçların doğruluğu ise yine yapılan sayısal deneyler sonucu açıkça görünmektedir.

Rezaee (2010), bu çalışmasında daha önceki çalışmalarda kullanılan kümeleme geçerlilik indeksleriyle ayrılan kümelerin ve bu kümelerin alt kümeleri dâhil oluşturduğu problemleri göz önüne alarak önerdiği yeni bir kümeleme geçerlilik indeksinden bahsetmektedir. Kompakt ve ayırma ölçütünün kullanıldığı bu indeks, üye matrislerdeki bulanıklığı veri setini temsil eden geometrik şekillerle prototipler üzerinden göstermektedir. BCO'dan yola çıkarak genişletilen GK kümeleme algoritması farklı merkezli ve şekilli alt kümeler oluşturmakta ancak bu algoritma ile çok az geçerlilik indekslerinden yararlandığı da belirtilmektedir. "Kim" geçerlilik indeksi, GK için önerilen bir geçerlilik indeksi, örtüşme içindeki her veri noktasının paylaşım derecesini göz önüne alarak, bulanık kümelerin her bir çiftinin çakışmasını hesaplar. Önerilen algoritma yedi farklı veri seti üzerinde daha önceden bilinen geçerlilik indeksleriyle birlikte uygulanıp karşılaştırılmaktadır. Önerilen indeksin diğer indekslerle karşılaştırıldığında üstün etkinliğini ve güvenilirliğinin olduğu değerler ile birlikte görülmektedir.

Gomathi ve Thangaraj (2010), bu çalışmada tıbbi alanda yapılan çalışmaların başında kümeleme tekniklerinin geldiğinden bahsetmektedirler. Çalışmada BCO ve BCO baz alarak geliştirilmiş diğer kümeleme algoritmaları tanımlanmaktadır. Bu çalışmada da BCO ve BOCO kümeleme algoritmaları baz

alınarak yeni bir yaklaşımda bulunmaktadır. Önerilen algoritma BCO algoritmasındaki mesafe ölçümünü değiştirerek bir pikselin diğer pikseller arasında olan gürültülü verilerin etkinliğini kontrol altına almaya çalışmaktadır. Önerilen algoritmalar ile karşılaştırılması istenilen kümeleme algoritmaları formüller ile açıklandıktan sonra gerçek veri seti üzerine Matlab programı üzerindeki çalışmanın sonucunda benzerlik, yanlış pozitif (Gerçek sağlamlar içinden testin hatalı olarak hasta dediği olgulardır) ve yanlış olumsuz (Gerçek sağlamlar içinden testin hatalı olarak sağlam dediği olgulardır) oranı olmak üzere 3 önemli parametre ele alınmaktadır. Sonuç olarak önerilen algoritma ve diğer BCO algoritmaları ele alınan veri seti üzerine uygulandıktan sonra, istenilen 3 parametre ile değerlendirilmektedir. Sonuçta önerilen algoritmanın , şekil ve grafikler üzerinde diğer BCO algoritmalarına göre daha iyi sonuç verdiği gözlenmektedir.

Ouintanilla-Dominguez ve Ark. (2011), çalışmada sayısallaştırılmış mamografi içindeki mikro kalsifikasyonları belirlemeye yönelik yaklaşımları sert ve yumuşak kümeleme teknikleriyle ele almaktadırlar. Elde edilen şekiller üzerinde matematiksel morfoloji kullanılmaktadır. Daha sonra bölümlenen görüntüler üzerine daha önce belirlenen KO, BCO ve OBCO gibi 3 farklı algoritma uygulanıp durumun avantaj ve sakıncalarından bahsedilmektedir. Uygulama için algoritmaların şartları ayrı ayrı belirlenmektedir. KO ve BCO bölgesindeki küme sayısının artışı için çok yararlı olduğu bunun yanı sıra bölge belirlemede de önemli rolü olduğu gözlenmektedir. Sonuç itibarıyla belirlenen metodolojiler farklı bölgeler üzerine uygulanıp test edilmektedir. Bunların yanı sıra bu metodların geliştirilmeye açık olduğu ve ileride yapılacak çalışmalara katkı sağlayacağı söylenmektedir.

Vanisri ve Loganathan (2011), çalışmada model tanımlamada bulanık kümeleme algoritmalarının sıklıkla kullanıldığından bahsetmekte ve yapılan çalışmada Olabilirlikli ve Olabilirlikli yaklaşımlar keşfetmektedirler. Gösterilen ilk örnekte 30 puanlık her bir küme 2 ayrı bileşene ayrılmaktadır. Bu işlemden sonra BCO, OCO, BOCO ve son olarak da İOCO kümeleme algoritmaları

uygulanmaktadır. Uygulama sonunda ilk etapta çıkan sonuçların yakın, sentroitler arası sonuçların ise nerdeyse aynı olduğu gözlenmektedir. Bir sonraki işlemde kümelere puan eklemesi yapılabilmekte ve çıkan sonuçlar arasında çok önemli farklılıklar olduğu gözlenmektedir. Burada BCO ve BOCO'nun en kötü sonuçlar verdiği, diğer iki algoritmaların ise daha yakın ve sağlıklı sonuçlar verdiği gözlenmektedir. Bu yapılan işlemler grafikler üzerinde de gösterilip açıkça saptanmaktadır. Sonuç itibariyle algoritmaların geliştirilmesiyle birlikte oluşan gama faktörünün etkisiyle de İOCO'nun diğer algoritmalara nazaran daha iyi ölçümler gösterdiği ve bunun geliştirilmeye daha da açık olduğu ifade edilmektedir. Ek olarak İOCO özellikle denetimsiz kümeleme yöntemlerinde faydalı bir kalıp olduğu da söylenmektedir.

Alamelumangai ve Devishree (2012), bu çalışmalarıyla elde edilen ultrason resimlerini ön işleme almış, gürültü giderme ve resimler çoğaltma işlemlerini uygulamış ve kanserli bölgeleri belirlemişlerdir. Bu araştırmada yeni bir bölünme tekniği kullanılmaktadır. Bu tekniğin adı İtmeli Bulanık Olabilirlikli C-Ortalama (İBOCO)'dır. Bu teknikte hem BOCO ve hem de diğer C-Ortalamların avantajlarından faydalanılmaktadır. Araştırmada 10 adet ultrason fotoğrafı ele alınıp önerilen teknik uygulanmıştır. Bu algoritma uygulamasıyla resimler daha yoğun olarak kümelenecek olmakta ve bu da kanserli bölgenin nerede olduğu tanısını bulmak açısından oldukça kolaylık sağlamaktadır. Uygulanan veriler doğrultusunda, sonucun önerilen yöntemin diğer geleneksel işlemlere göre çok daha iyi sonuç verdiği ve standart hatanın da daha az olduğu gözlenmektedir.

Krishnan ve Ramamoorthy, (2012), bu çalışmada görüntüyü bölümlere ayırma teknikleriyle beyin görüntüleri üzerinde çalışmaktadırlar. Daha önceleri manyetik rezonans görüntüleme yöntemi uygulanmakta ancak bu yöntem ile görüntü tanısından birçok rahatsız edici görüntüye rastlanılmaktadır. İlk olarak BCO kümeleme algoritması uygulanmaktadır. Bu tekniğin görüntü üzerindeki kenar bölgelerde istenilen sonucu vermemesi sonucu BOCO algoritması uygulanmaktadır. Bu yöntemde görüntü üzerinde rahatsız edici görüntüler

bulundurması üzerinde Modifiye Edilmiş Bulanık Olabilirlikli C-Ortalamlar (MBOCO) kümeleme algoritması önerilmektedir ki yapılan araştırmada bu yöntemle elde edilen sonuç daha önceki algoritmalarından elde edilen sonuçlara göre daha iyi sonuç verdiği görülmektedir. Şekilde açıkça gözlemlendiği gibi en son olarak önerilen MBOCO algoritmasının diğer algoritmalara göre daha iyi sonuç verdiği ortaya çıkmıştır.

Zhang ve Qian (2012), bu çalışmada veri setleri içerisinde en iyi ayrılmış yapıları seçmede yardımcı kümeleme geçerlilik indekslerine katkı olarak yeni bir indeks önermektedirler. “DWSC” adı altında ifade edilen bu yeni geçerlilik indeksinin, örtüşen kümelerin bulunduğu durumlarda çok iyi çalıştığından bahsetmektedirler. Bulanık kümelemedeki ayırım ve kompaktlık bilgilerini birleştirerek, kümeler ve kümeler arası hata kare ağırlığı arasındaki çakışma derecesinin dinamik ağırlığının toplamını tanımlamaktadır. Önerilen yeni indeks için yapılan tanımlamalarla birlikte, bulanık bir bölümün düşük örtüşme derecesine ve küçük bir kompaktlık değerine sahip olması beklenir. 6 yapay ve UCI veri havuzundan alınan 4 gerçek veri setleri (Iris, Pima, Breast Cancer, Wine) üzerinde uygulanan deney sonuçları ile elde edilen sonuçlar, önerilen indeksin diğer indekslerle karşılaştırıldığında üstün etkinliğini ve güvenilirliğini göstermektedir.

Saad ve Alimi (2012), çalışmada kümeleme analizlerinde kullanılan veri analizleri, görüntü işleme ve örüntüleri tanımlama konusunda en çok uygulanan yöntemlerden birisi olan kümeleme geçerlilik indekslerinin olduğunu söylemektedirler. Sonuçların değerlendirilmesi konusunda en önemli parametrelerden birisinin küme sayısı (c) olduğundan ve kalan parametrelerin sonuç bölüm üzerinde daha az etkisi olduğundan bahsetmektedirler. Analiz sonuçlarında en iyi sınıf sayısını belirlemek için kullanılan geçerlilik indekslerinden bahseden yazar bu çalışmasında yeni bir indeks olan Modified Partition Coefficient And Exponential Separation (MPCAES)'yi ele almaktadır. Bu indekste önerilen geçerlilik ölçüsü gürültülü bir ortamda farklı bir görünüm verebilir. Her sınıf için, potansiyel ölçümlerinin tanımlanabileceğini ifade

etmektedir. Önerilen “MPCAES” endeksinin etkinliği birkaç popüler geçerlik indeksi ile karşılaştırılmıştır. Belirlenen veri seti ve uygulanan indeksler ile elde edilen sonuçlar, önerilen yeni endeksinin (MPCAES) vakaların çoğunluğunu istenen sınıfların değerini sağladığını kanıtlamaktadır.

Rawashdeh ve Ralescu (2012), bu çalışmada bazı popüler bulanık küme geçerlilik indeksleriyle birlikte bulanık bölümlere genellemesine dayanan bir indeks olan Silhouettes (S)’ten bahsettikten sonra BCO kümeleme algoritması ile bağlantılı olarak karşılaştırmışlardır. İyi bir kümeleme çalışmasının dikkatli bir şekilde seçilmesi yani benzerlikleri-farklılıkları iyi tanımlaması gerektiğini düşünmektedirler. Farklı modellerden oluşan kümeleme algoritmaları, küme sayısı gibi değişken model parametreleri bazı yapılar üretirler. Geçerlik endeksinin görevi, çift yönlü benzerlikler-farklılıkları, diğer bir deyişle altta yatan yapıya en uygun kümelene ile en iyi desteklenen yapıyı bulmaktır. BCO, aslında aynı kümedeki puanların farklı üyelik değerlerine sahip olduğu noktalara atanan kısmi küme üyelikleri tarafından kolaylaştırılan, veri setlerinde çakışan kümeleri modellemek için kullanılır. Genelleştirilmiş S indeksi, veri kümesindeki en iyi yapıyı araştırmaya rehberlik etmek için yapı modellemesinin keskin ve bulanık iki yaklaşımı için de geçerlidir. Çalışmada mesafeye dayalı benzerlikler ve farklılıklar, mesafe büyüklüklerine değil, birbirleriyle mesafe değerlerinin karşılaştırılmasından anlaşılmaktadır. Dolayısıyla, altta yatan yapının gücü birbirine göre uzaklık değerleri ile belirlenir. Sonuç olarak, yapılan karşılaştırma ile uygulama üzerinde sonuçlar değerlendirilip grafik ve tablolar ile gösterilmektedir.

Joopudi ve Ark. (2013), bu çalışmada kümeleme algoritmalarının veri setleri üzerindeki başarımlarının küme sayısına bağlı olduğundan bahsetmektedirler. Kümeleme algoritmalarıyla ayrılan bölümler arasında uygun küme sayısını belirlemek için bugüne kadar birçok çalışma yapılmıştır ki bunlar arasında en önemlilerden birisi de kümeleme geçerlilik indeksleridir. Yazarlarda bu çalışmada bu konuyla ilgili “Graded Distance index” (GD_index) adında yeni bir indeks önermektedirler. GD_index, tüm veri noktalarının hem maksimum hem de

ikinci maksimum üyelik değerlerini kullanmaktadır. Önerilen indeks, tüm veri noktalarının kendi maksimum üyelik kümelerine doğru iletim gücünü en üst düzeye çıkarmakla birlikte, veri noktaları üzerindeki bulanık kümeler arasındaki çakışmayı en aza indirmektedir. GD_index, tüm veri noktalarının ilk maksimum üyeliği ile ikinci maksimum üyeliği arasındaki ortalama fark kullanılarak hesaplanır. İndeksin etkinliği, bilinen mevcut endekslerle karşılaştırılmakta ve çeşitli veri setleri üzerinde test edilmektedir. Test edilen indeksin, veri kümelerinin çoğunda optimal bölüm sayısını doğru bir şekilde hesaplayabileceği gözlemlenmektedir.

Ji ve Ark. (2013), bu çalışmada önerilen algoritma Aralık Değerli Olabilirlikli Bulanık C-Ortalamlar (AD-OBCO) teorik olarak açıklamakta ve daha sonrasında sıradan kümeleme algoritmalarından olan BCO, OCO, OBCO, Aralık Değerli Bulanık C-Ortalamlar (AD-BCO) ve Aralık Değerli Olabilirlikli C-Ortalamlar (AD-OCO) ile belirlenen veri seti üzerine uygulanıp karşılaştırmaktadırlar. AD-OBCO kümeleme algoritmasının sıradan algoritmalarından en büyük farkı, kümelerdeki belirsizlikleri, iki bulanıklaştırıcı (m_1, m_2) kullanarak genel BCO, OCO ve OBCO algoritmaları iki bulanıklaştırıcı (m_1, m_2) kullanarak geliştirilip formülize etmektir. Yapılan araştırmada önerilen algoritma ile karşılaştırılması istenilen algoritmalar iki boyutlu Gauss veri setleri, kıyaslamalı çoklu veri setleri, beyin Manyetik Rezonans (MR) görüntü setleri ve doğal görüntü setleri üzerinde uygulanmaktadır. Çıkan sonuçlarda önerilen AD-OBCO'nun karşılaştırılan diğer algoritmalara göre daha sağlam sonuç verdiği görülmekte ancak ek olarak veri setlerinin büyümesi halinde bu algoritmanın da hassaslaşacağı ve geliştirilmeye açık olduğu bilgisi de verilmektedir.

Pimentel ve Ark. (2013), temel olarak üyelerin daha doğal ve sağlam sonuçlar ortaya çıkaracağına yönelik çalışma yapılacağından bahsetmektedirler. Önerilen yaklaşım yapay ve gerçek veri setleri üzerine uygulanmaktadır. Modeldeki üyelerin özelliklerinin farklı dağılımlara sahip olabileceği ve BCO' nun başarımını etkileyeceği düşünülmektedir. Ancak modelin her bir üyenin özelliğini

belirleme konusunda daha doğal sonuçlar çıkaracağı, sonuçların da daha iyi olabileceği savunulmaktadır. Önerilen algoritma, üyelikleri dereceleri özellikleri bakımından birbirinden farklıdır. Bu çalışmanın uygulamasında 4 adet yapay 5 adet gerçek veri seti kullanılmaktadır. Veri setleri üzerine önerilen çok değişkenli BCO ve klasik BCO kümeleme algoritmaları uygulanmakta ve karşılaştırılmaktadır. Çalışmada önerilen ve karşılaştırılması istenilen algoritmalara ek olarak, kümelerin şekli ve büyüklüğünün tanımlanması açısından GK algoritması bu veri setleri üzerine uygulanmaktadır. Hesaplamalarda Mahalanobis uzantısı kullanılmaktadır. Karşılaştırmalar sonucunda üyelerin eşit derecede alınamayacağı ve farklı dağılımlar göstereceği düşünülmektedir.

Biju ve Mythili (2014), çalışmalarında Cdna mikrodizi görüntülemenin aynı anda genler ile ilgili binlerce yararlı bilgi sağlandığını gözlemlemektedirler. Bu sebepten de mikrodizi görüntüleme segmentasyonu çok önemli bir taslaktır. Bu çalışmada test edilen literatürler üzerinde cDNA mikrodizideki gürültülü verilere bulanık kümeleme algoritmaları uygulanmaktadır. Burada BCO, OCO, OBCO ve Bulanık Yerel Bilgi C-Ortalamlar (BYBCO) kümeleme algoritmaları uygulanmaktadır. Çalışmada kullanılacak yöntemlerde, teorik olarak ne tür veri tabanları kullanıldığı açıklanmaktadır. Uygulama sonucunda hem yapay hem de gerçek veriler üzerinde tablo-1'in ve diğer şekillerde görüldüğü gibi BYBCO'nun diğer algoritmalara göre daha iyi sonuç verdiği noktalar açıkça gözlenmiştir.

Xenaki ve Ark. (2014), bu çalışmada aralık uyumlu yeni bir Olabilirlikli kümeleme algoritmasından bahsedilmektedirler. Aykırı veri ve gürültülü noktalara karşı olan zafiyete göre bu durumun önlenmesi amaçlı geliştirilen diğer algoritmalarından olan Olabilirlikli kümeleme algoritmalarının daha iyi sonuçlar verdiği gözlenmektedir. Olabilirlikli kümelere aralıklı olarak bazı değişiklikler empoze edilerek kısıtlamalar oluşturulmaktadır. Oluşturulan kümeler iyileştirilerek yeniden kazanılmaktadır. Bu çalışmada Olabilirlikli kümeleme algoritması iki yönde geliştirilmektedir. İlk olarak, küme sayısı ve Olabilirlikli kümeleme algoritması içindeki n parametreleri olarak bilinmektedir. Kümelenmiş her bir veri

vektörü üzerindeki uygunluk dereceleri üzerine uygun aralıklar koyarak kısıtlamalar getirmektedir. Yapılan analizler sonucunda önerilen Aralık Uyumlu C-Ortalamalar (AU-OCO) kümeleme algoritmasının KO, BCO ve geleneksel OCO kümeleme algoritmalarına göre gittikçe daha iyi sonuç verdiği gözlenmektedir. Ayrıca birbirine yakın kümeler arası yoğunlukları iyileştirme konusunda da etkili bir algoritma olduğu gözlenmektedir.

Rosso (2014), kümeleme analizi üzerinde aykırı (gürültülü) değerlerin varlığını araştırmak için yapmış olduğu çalışmada 3 farklı kümeleme algoritmasından, bu algoritmaların aykırı değerler ile arasındaki ilişkilerden bahsetmektedir. İlk önce yapay olarak oluşturulan simetrik veri seti üzerine sert kümeleme algoritması uygulanıp değerlendirilmiştir. Sonraki işlemde veri setinin tam ortasına yeni bir veri eklenmiş ve tekrar hesaplamalar yapılmıştır. Burada ilk yapılan işleme göre farklı bir yapının ve doğruluktan uzak bir grafiğin ortaya çıktığı görülmüştür. Çıkan bu sonuçtan dolayı alternatif yöntem olarak küme sayısının artırılmasına karar verilmiş ve bir diğer alternatif olan bulanık kümeleme yöntemi uygulanmıştır. Uygulanan bulanık kümeleme yöntemi sonucunda üyelik derecelerinin farklı değerlere sahip olduğu ve farklı bir grafik yapısının ortaya çıktığı görülmektedir. Aynı zamanda küme merkezinin de değiştiği gözlenmektedir. Son olarak durumu daha genel ve sağlam hale getirmek, bulanık kümeleme yönteminde hesaplamada olduğu gibi bir diğer verinin eklenmesiyle oluşan grafikte de küme merkezi değişmektedir. Ayrıca hesaplama sonucunda üyelerin derecelerinin de değiştiği hatta başta bahsetmiş olduğumuz üyelik derecelerinin toplamının 1'e eşit olma kısıtının ortadan kalktığını da z12 noktasında görmekteyiz.

Adekunle ve Ark. (2014), kümeleri uygun bölümlere ayırma durumunu değerlendirme konusunda popüler geçerlilik indekslerinden farklı olarak “inter-

cluster ve intra-cluster separation” (IC2S) adında yeni indeksler önermektedirler. Kümeler arasında mevcut olan küme içi ve küme arası ayrımları birleştiren bir dengesizlik fonksiyonu öneren yazarlar, veri seti üzerinde önerilen indeksi ve daha önceden bilinen 5 geçerlilik indekslerini kullanarak yapmış oldukları uygulama çalışmasında alınan sonuçları karşılaştırıp değerlendirmişlerdir. Çalışma sonucu, önerilen “IC2S” indeksinin iyi bir küme sayısı tahmini üretme kabiliyetine sahip olduğunu ve iki veri seti (Fisher'ın IRIS ve Kelebek veri seti) üzerinden alınan sonuçların, IC2S’ de bulanık c-ortalamaları için diğer kümeleme geçerlilik indekslerinden daha iyi başarımlar elde edildiğini göstermiştir.

Cebeci ve Yıldız (2015), BCO kümeleme algoritmasının yapay olarak üretilen çeşitli boyutlarda veri üzerindeki başarımlarını çeşitli geçerlilik indeksleri ve işlem zamanı açısından karşılaştırmışlardır. Çalışmada farklı veri büyüklükleriyle 5’er kümelik 6 adet yapay veri seti R programı ile üretilmiş; üretilen veriler “fclust” paketindeki geçerlilik indeksleri ile değerlendirilmiştir. Çalışmada PC geçerlilik indeksinin uygun kümelemeyi ortaya koymak için yeterli olduğu belirtilmiştir.

Cebeci ve Ark. (2015), çalışmada ilk olarak Pham ve Ark. (2005) tarafından önerilen $f(K)$ fonksiyonunu tanımlamakta ve üretilen yapay veri setleri üzerinde bir kümeleme analizi çalışması yapmaktadırlar. Çalışmada K-Ortalamalar kümeleme algoritması ve “k-selection” paketi ile 4 farklı şekilde (kare, daire, elips, konkav) değişik yerleşimde kümeler içeren 12 tane yapay veri seti kullanılmaktadır. Elde edilen bulgulara göre k-selection algoritmasının küme sayısını tahmin etmede başarılı sonuçlar verdiği ortaya konmuştur.

Cebeci ve Yıldız (2015), farklı şekillere sahip yapay verileri ve gerçek veri setleri üzerinde BCO ve K-ortalamalar algoritmalarını bazı dış indekslere göre karşılaştırmışlar ve BCO algoritmasının diğer algoritmaya göre daha iyi sonuç verdiğini saptamışlardır.

Zhang ve Chen (2016), çalışmalarında daha önceden iyi bilinen BCO ve OCO genişletilmesiyle birlikte oluşan bir kümeleme algoritmasından

bahsetmektedirler. Orijinal Öklid uzaklığı için oluşturulmuş mesafe modelinin ikame edilmesi ile gerçekleştirilmiş olan bu algoritmalar, Kernel'in Bulanık C-Ortalaması (KBCO) ve Olabilirlikli C-Ortalamalar (KOCO) algoritmaları üzerinden test edilip değerlendirilmiştir. Yapılan ilk testte bir veya daha fazla aykırı değerler içeren veri setlerinin oluşturduğu kümelerde KBCO ve KOCO' nun BCO ve OCO' ya göre çok daha iyi ve sağlam sonuç verdiği gözlenmektedir. İkinci teste aykırı veri içeren 2 kümeye uygulanan bu algoritmalar arasında en kötü sonucu BCO' nun verdiği görülmektedir. OCO' nun da BCO gibi çok iyi bir sonuç vermediği görülmekle birlikte KBCO ve KOCO' nun da aykırı verilerden etkilendiği şekiller üzerinde görülmektedir. Üçüncü ve son örnekte ise BCO ve KBCO, veri kümesinin doğal yapısını temsil etmektedir. Ancak KBCO, BCO' dan biraz daha üstündür. OCO kullanıldığında, sonuçlar oldukça farklıdır. OCO tarafından elde edilen prototipler neredeyse aynı görülmektedir. Ancak, bu durum KOCO' da görünmemektedir.

Singhal ve Deepika (2016), bu çalışmada denetimsiz öğrenme için sunulan çeşitli kümeleme modelleri arasından 2005 yılında OCO ve BCO' nun karma bir modeli olarak önerilen OBCO' nun, esasen tipiklik değerleri, üyelik değerleri ve kümeleme merkezleri gibi üç değer ürettiğinden bahsetmektedirler. Metin dosyaları için kullanıldığında bu önerilen kümeleme, özel önişleme adımını gerektirmektedir. Bu adım doğru şekilde uygulandığında, OBCO kümeleme algoritmasının, sözcükleri bir metin dosyasında çok doğru bir şekilde topladığı saptanır. Genel olarak metin dosyalarını kümelemek için OBCO kümeleme algoritması önerilmektedir.

Singla ve Khehra (2016), görüntü tanımlama konusunda en etkili uygulamalardan birisinin hiç şüphesiz yönetimsiz kümeleme teknikleri olduğundan bahsetmektedirler. Bu makalede, Sert Kümeleme tabanlı standart KO ve BCO' nun yanı sıra OBCO gibi farklı kümeleme algoritmalarının başarımları incelenmektedir. Bu algoritmalar standart ve renkli görüntüleri bölümlenmek ve analiz etmek için kullanılır ancak bu araştırma, gürültülü gri tonlamalı görüntülerle ilgilidir. Ayrıca

uygulamada kullanılan program olan Matlab, gri tonlamalı görüntülerin görüntü kalitesini kontrol etmek için PSNR, MSE ve SSIM parametrelerinin işlevine de sahiptir. KO, FCM ve PFCM sonuçlarını karşılaştırmak için değerlendirme parametresi olarak PSNR, MSE ve SSIM kullanılmaktadır. Neticede, deneysel sonuçların OBCO' nun BCO ve KO kümeleme algoritmalarının üzerinde daha olumlu sonuçlar gösterdiğine ulaşılmakta ve OBCO kümeleme algoritmasının, gürültülü gri tonlamalı görüntülerin segmentasyonu için daha iyi olduğu sonucuna varılmaktadır.

Askari ve Ark. (2017), bu çalışmada gürültülü gerçek veri setleri üzerine Genelleştirilmiş Olabilirlikli Bulanık C-Ortamalar (GOBCO) kümeleme algoritmasının başarımını önerilen geçerlilik kümeleme indeksleriyle test etmektedirler. Verilerin gürültülü olduğu durumlarda bu algoritmanın BCO, OCO ve OBCO ile karşılaştırıldığında daha doğru olduğu gösterilmiştir. Kümeler farklı boyutlarda olduğunda veya kovaryans normu matrisi kullanıldığında BCO, OCO ve OBCO, temiz verilerde bile doğru prototipleri bulmakta başarısız olurken, GOBCO kovaryans norm matrisinin ise kümesi boyutuna ve türüne duyarsız olduğu görülmektedir. Veriler son derece gürültülü olduğunda GOBCO doğru küme merkezleri bulur ancak BCO, OCO ve OBCO algoritmaları bunun başarısız olduğunu gösterir. Gürültülü veri setlerinde kümelerdeki kompaktlığı, ayrımı ve son olarak her iki durum açısından küme sayılarını belirlemek için 3 farklı indeks tanımlanmaktadır. Bu endekslerin başarımının, çeşitli veri örnekleri kullanılarak tatmin edici olduğu görülmektedir.

3. MATERYAL VE METOT

3.1 Materyal

Bu tez çalışmasında 8 sentetik ve 4 gerçek veri seti olmak üzere toplam 12 veri seti kullanılmıştır. Bu veri setlerinin yapıları ve özellikleri aşağıda açıklanmaktadır.

3.1.1. Sentetik Veri Setleri

Sentetik veri setleri küme şekillerine bağlı olabilecek başarımların farklılıklarını incelemek için:

- a) Daire şekilli
- b) Elips şekilli
- c) Kare şekilli
- d) Konkav şekilli

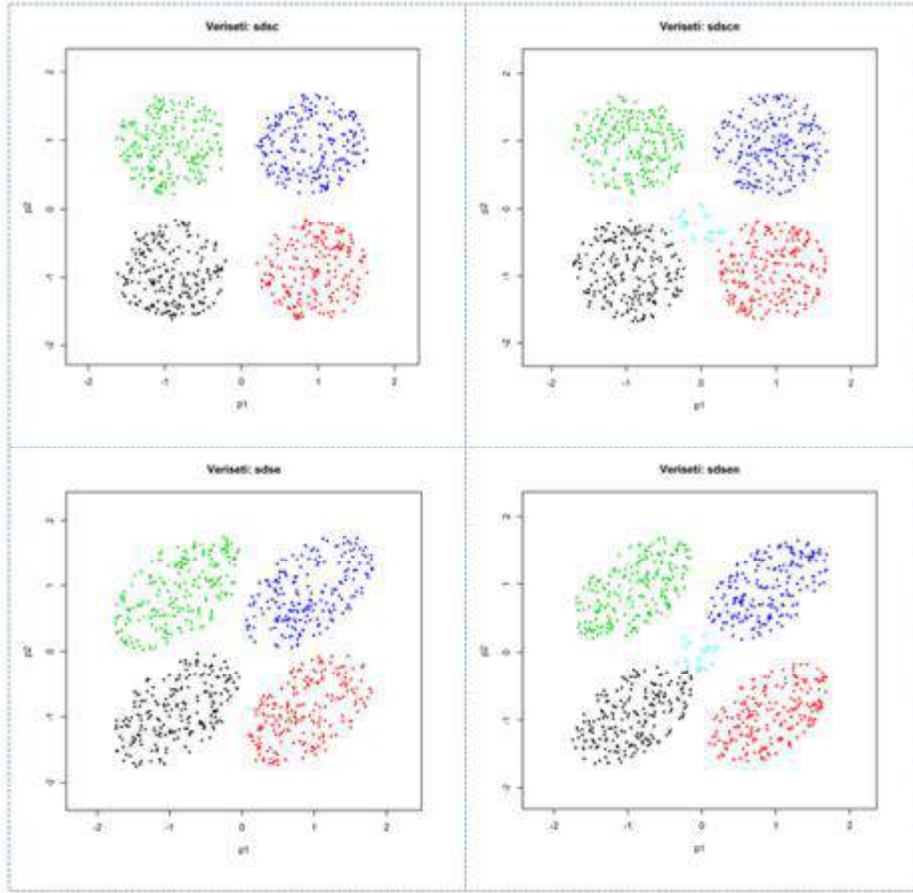
kümeleri içerecek şekilde üretilmesi amaçlanmıştır. Bu veri setlerinin her birine %1 düzeyinde gürültü eklenerek:

- a) Gürültüsüz
- b) Gürütlü

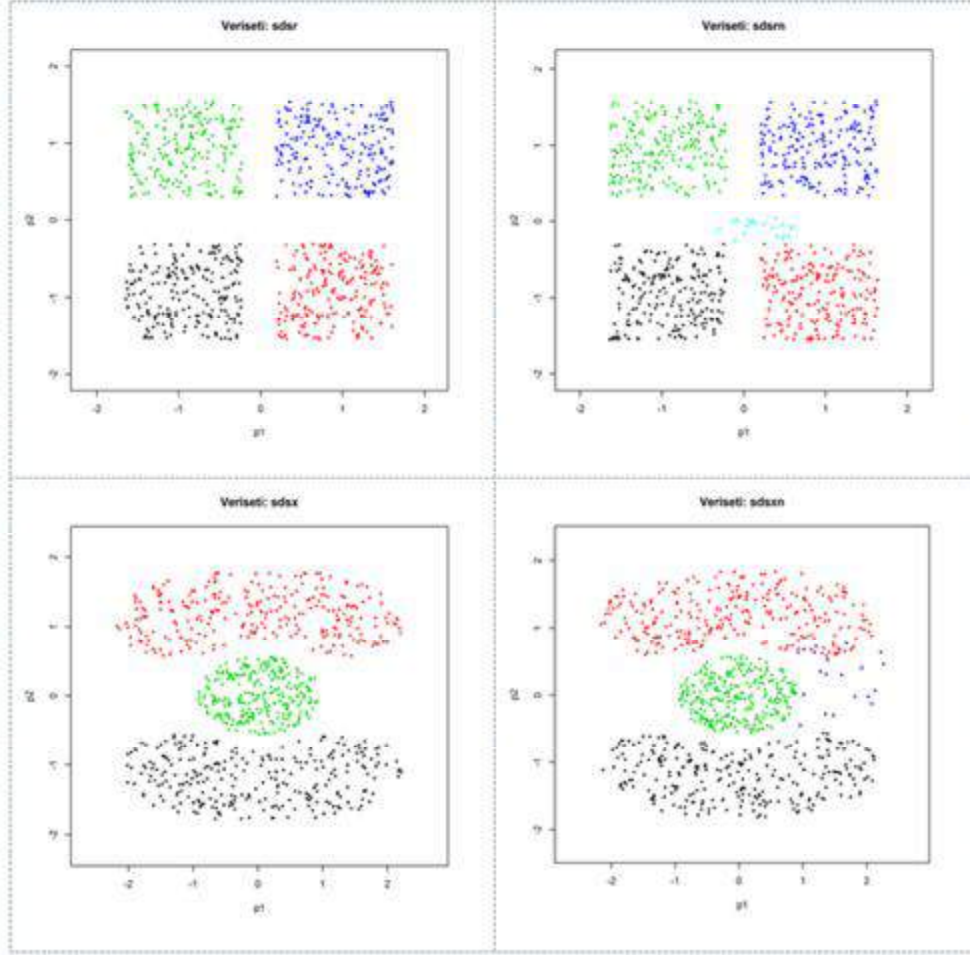
veri setleri oluşturulmuştur (Çizelge 3.1). Bu veri setlerinin her birinde mozaik (tiled) yerleşimle 200'er veri biriminden oluşan 4'er küme oluşturulmuş ve küme etiketleri eklenmiştir. Sentetik veri setlerindeki küme yapıları ve yerleşimleri Şekil 3.1 ve Şekil 3.2'de gösterilmiştir.

Çizelge 3 1:Sentetik Veri Setleri Ve Özellikleri

Veri Seti	Küme şekli - Gürültü varlığı	Nesne sayısı	Özellik sayısı	Küme sayısı
SDSC	Daire şekilli – Gürültüsüz	800	2	4
SDSCN	Daire şekilli – Gürültülü	820	2	4
SDSE	Elips şekilli – Gürültüsüz	800	2	4
SDSEN	Elips şekilli – Gürültülü	820	2	4
SDSR	Kare şekilli – Gürültüsüz	800	2	4
SDSRN	Kare şekilli – Gürültülü	820	2	4
SDSX	Konkav şekilli – Gürültüsüz	800	2	3
SDSXN	Konkav şekilli – Gürültülü	820	2	3



Şekil 3.1. Sentetik Veri Setlerinde İlk İki Özelliğin Mevcut Kümelere Göre Dağılımı



Şekil 3.2. Sentetik Veri Setlerinde İlk İki Özelliğin Mevcut Kümelere Göre Dağılımı

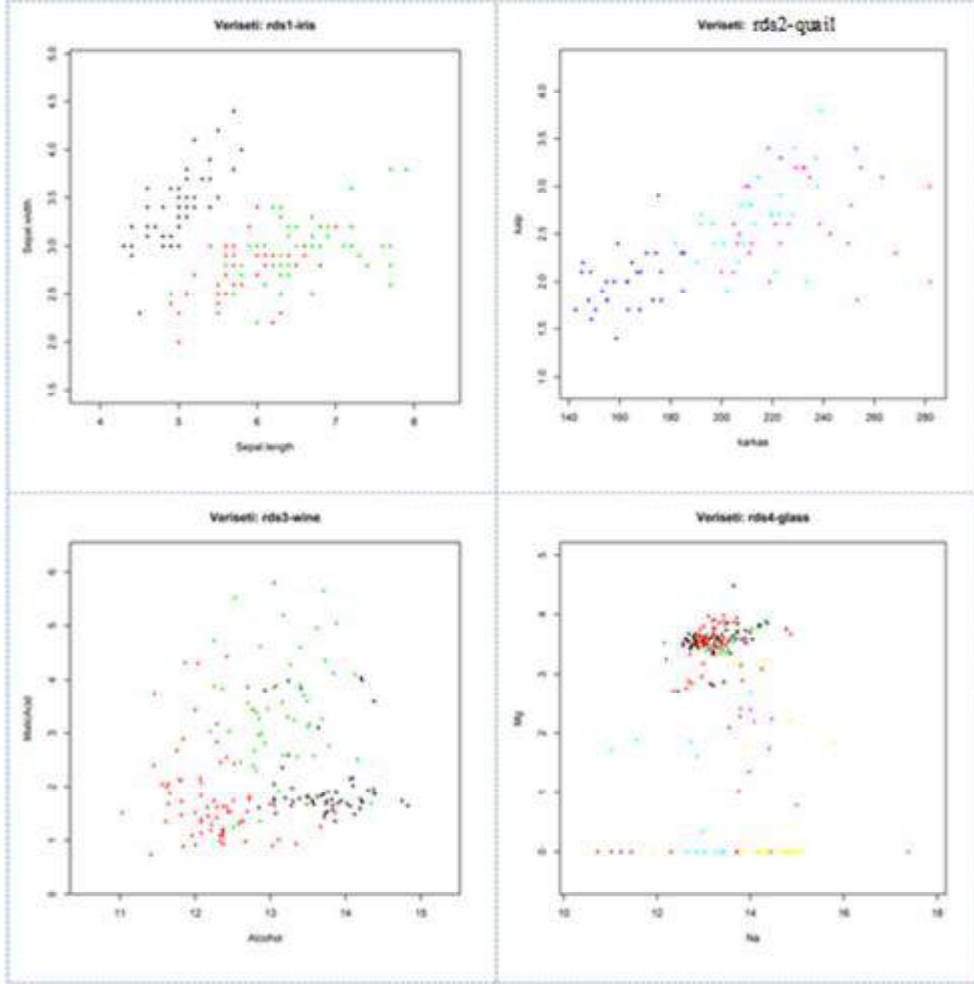
Sentetik veri setleri, R ortamında (R Core Team, 2017) `stats`, `ellipse` (Duncan ve Chow, 2013), `splancs` (Rowlingson ve Diggle, 2016) ve `mlbench` (Leisch ve Dimitriadou, 2010) gibi paketlerden çeşitli fonksiyonlar kullanılarak geliştirilen bir R programıyla 2 özellikli ($p=2$) olarak üretilmiştir. Veri setlerindeki iki özellik (değişken veya öznitelik olarak da adlandırılmaktadır) kullanılmasında, kümeleme sonucunda elde edilen kümelerin iki boyutlu uzayda serpilme diyagramları ile kolaylıkla İzlenmesi ve tartışılması amaçlanmıştır.

3.1.2. Gerçek Veri Setleri

Araştırmada test edilen algoritmaların başarımlarını test etmek için üçü UC Irvine Machine Learning Repository (<http://archive.ics.uci.edu/ml/>) deposundan, biri Ç.Ü.Z.F’de araştırma çalışmalarından seçilen ve Çizelge 3.2’de listelenen dört gerçek veri seti kullanılmıştır.

Çizelge 3 2. Gerçek Veri Setleri Ve Özellikleri

Veri seti	Nesne sayısı	Özellik sayısı	Küme sayısı
RDS1-Iris	150	4	2, 3
RDS2-Quail	90	4	3
RDS3-Wine	143	13	3
RDS4-Glass	191	10	6



Şekil 3.3. Gerçek Veri Setlerinde İlk İki Özelliğin Mevcut

Anderson (1935)'in Iris veri seti, veri madenciliği, yapay zekâ ile yönetimli ve yönetimsiz kümeleme çalışmalarının çoğunda test amacıyla kullanılan tanınmış bir veri setidir. Bu veri setinde 3 süsen türünün her biri için 4 özelliğe ait 50 adet gözlem yer almaktadır. Iris veri setinde (RDS1-Iris) son sütunda gözlemlerin ait olduğu süsen türlerini gösteren sınıf (küme) etiketleri (*I. setosa*, *I. versicolor* ve *I. virginica*) bulunmaktadır.

Quail veri seti (RDS2-Quail), Ç.Ü. Ziraat Fakültesi Araştırma ve Uygulama Çiftliği'nde 2015 yılında yetiştirilen 30 bıldırcına ait karkas ağırlığı, karaciğer, kalp ve taşlık gibi yenilen iç organ ağırlıkları olmak üzere 4 özelliğe ait 3, 4 ve 5. Haftalarda ölçülmüş değerlerini içermektedir. Veri setinde 90 kayıt ve haftaları gösteren 3 sınıf yer almaktadır.

Wine veri seti (RDS3-Wine), İtalya'da aynı bölgede yetiştirilen 3 farklı üzüm çeşidinden üretilen şaraplarda yapılan kimyasal analiz sonuçlarını içermektedir. Veri setinde kimyasal analiz 143 şarap örneğinde alkol, malik asit, kül, magnezyum vb. şarap kalitesini tanımlayıcı 13 özelliğe ait değerler ve bu değerlere bağlı olarak belirlenmiş sınıf etiketleri (3 sınıf) bulunmaktadır. Veri setine <http://archive.ics.uci.edu/ml/datasets/Wine> adresinden erişilebilmektedir.

ABD Adli Bilimler Servisi kayıtlarından sağlanan 191 adet bardak örneğinde 10 yapısal bileşene (Na, Fe, K vb) ait değerleri içeren Glass veri setine ve [tanımlayıcı bilgilerine](https://archive.ics.uci.edu/ml/datasets/Glass+Identification) [adresinden](https://archive.ics.uci.edu/ml/datasets/Glass+Identification) erişilebilmektedir. Glass veri setinde (RDS4-Glass) 6 bardak sınıfı yer almakta dolayısıyla test amaçlı çalışmalarda referans kümeler veya sınıflar olarak kullanılmaktadır.

3.2 Metot

3.2.1. Bölümleyici Kümeleme

3.2.1.1. Temel Kavramlar Ve Gösterimler

Kümeleme işlemi yapılacak veri setinde her bir gözlem p -boyutlu bir özellikler vektörü içinde yer alan p sayıda özellekle karakterize edilir. Bu özellikler vektörü k . özellik için:

$$\mathbf{x}_k = \{x_{1k}, \dots, x_{pk}\}^T, \mathbf{x}_k \in \mathbb{R}^p \quad (3.1)$$

olarak yazılabilir. Bu durumda n gözlemlik sonlu bir veri seti:

$$\mathbf{X} = \{\mathbf{x}_k | k = 1, 2, \dots, n\} \quad (3.2)$$

ile gösterilir ve $n \times p$ boyutlu veri matrisi olarak aşağıdaki gibi yazılabilir (di Martino ve Sessa, 2009):

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1p} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{np} \end{bmatrix} \quad (3.3)$$

Bölümleyici kümelemede amaç, p -özellikli n vektörden oluşan \mathbf{X} veri setinin (3.3) belli bir uzaklık normu, küme prototipleri (ya da küme merkezleri) ve diğer birincil-derece koşullarla tanımlanmış sert ve yumuşak bölümeleme algoritmalarına ait amaç fonksiyonların (J) minimize edilerek bölümlenmesidir.

3.2.1.2. Sert Ve Yumuşak Kümeleme

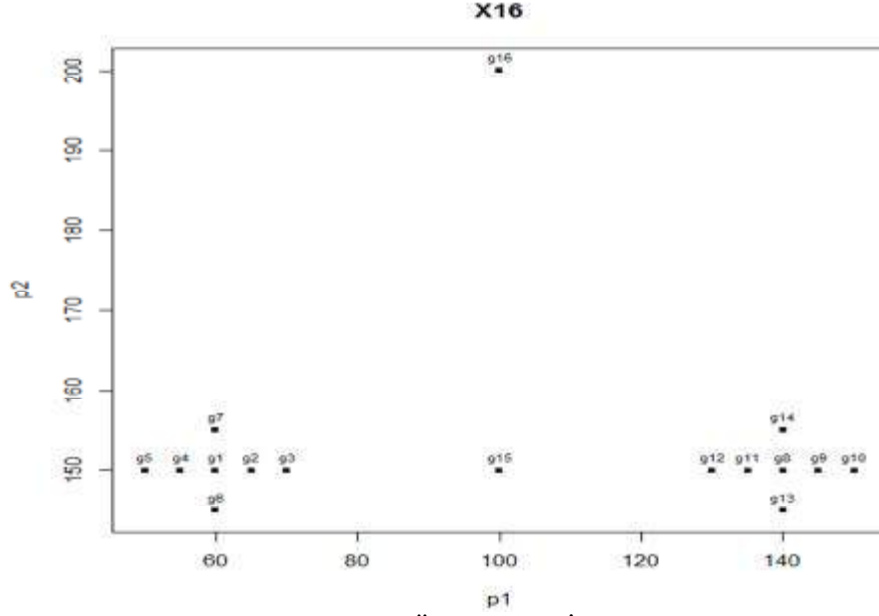
Bölümleyici kümeleme algoritmaları sert (hard) ve yumuşak (soft) kümeleme algoritmaları olarak iki grupta incelenmektedir. Sert kümelemede \mathbf{X} veri setindeki nesnelere her biri yalnız bir kümenin üyesi olabilmektedir. Yumuşak kümelemede ise nesnelere yalnız belli bir kümenin üyesi değil fakat aynı zamanda farklı üyelik dereceleriyle birden fazla kümenin üyesidirler. Bir başka deyişle nesnelere belli bir kümeye üye olmaya zorlanmamakta, aksine 0 ve 1 arasında değişen derecelerle birçok kümeye üye olmaktadır. Bu şekilde, \mathbf{X} veri setindeki kümelerin sınırlarında bulunan veya herhangi iki kümeye eşit uzaklıkta bulunan nesnelere küme üyelikleri ile sert kümelemede karşılaşılan sorunlar çözüme kavuşturulmaktadır.

Çizelge 3.3'de verilen 16 elemanlı sentetik veri setinde (X16), 16 nesnenin (g_1, \dots, g_{16}) iki özelliğine (p_1, p_2) ait gözlem değerleri yer almaktadır. X16 veri

setindeki gözlemlerin Şekil 3.4'deki serpilme diyagramı izlendiğinde, nesnelerin biri solda diğeri sağda olmak üzere iki farklı bölgede toplandıkları, dolayısıyla iki kümenin mevcut olabileceği anlaşılmaktadır. Ancak grafikte *g15* gözleminin bu kümelerin tam ortasında bulunması nedeniyle soldaki ve sağdaki kümelerden hangisine ait olacağı problemi söz konusudur. Aynı sorun, diğelerine göre daha uzakta yer alması nedeniyle bir aykırı değer olarak yorumlanabilecek *g16* nesnesi için de geçerlidir.

Çizelge 3.3. İki özellikli (p_1 ve p_2) örnek bir veri setinde (X_{16}) nesnelere ait gözlem değerleri

Nesne	p_1	p_2
<i>g1</i>	60	150
<i>g2</i>	65	150
<i>g3</i>	70	150
<i>g4</i>	55	150
<i>g5</i>	50	150
<i>g6</i>	60	145
<i>g7</i>	60	155
<i>g8</i>	140	150
<i>g9</i>	145	150
<i>g10</i>	150	150
<i>g11</i>	135	150
<i>g12</i>	130	150
<i>g13</i>	140	145
<i>g14</i>	140	155
<i>g15</i>	100	150
<i>g16</i>	100	200



Şekil 3.4. X16 Veri Seti P1 Ve P2 Özellik Çifti İçin Serpilme Diyagramı

Şekil 3.4'e bakıldığında bu nesnelerin sol ve sağdaki toplanma (kümeleme) merkezlerine göre aynı uzaklıkta olmalarına karşın sert kümelemede seçilen başlangıç küme merkezlerine göre ya soldaki ya da sağdaki kümenin üyesi olmaya zorlanmakta ve bu durum yanlış kümeleme yorumlarına yol açabilmektedir. Örneğin X16 veri seti, K-ortalamlar kümeleme algoritması ile farklı 4 küme sayısı ($c=2, \dots, 5$) için analiz edildiğinde üyelik matrisleri Çizelge 3.4.-3.7'de gösterildiği gibi elde edilmektedir

Çizelge 3.4. K-Ortalamlar Kümelemesine Göre C=2 Küme Sayısında Üyelik Dereceleri Matrisi

Küme	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	g13	g14	g15	g16
C1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Çizelge 3.5. K-Ortalamlar Kümelemesine Göre C=3 Küme Sayısında Üyelik Dereceleri Matrisi

Küme	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	g13	g14	g15	g16
C1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0
C3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Çizelge 3.6. K-Ortalamlar Kümelemesine Göre C=4 Küme Sayısında Üyelik Dereceleri Matrisi

Küme	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	g13	g14	g15	g16
C1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0
C3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
C4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Çizelge 3.7. K-Ortalamlar Kümelemesine Göre C=5 Küme Sayısında Üyelik Dereceleri Matrisi

Küme	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12	g13	g14	g15	g16
C1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
C2	0	0	0	0	0	0	0	1	0	0	1	1	1	1	0	0
C3	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
C4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
C5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Çizelge 3.4. incelendiğinde $g15$ ve $g16$ 'nın tam üyelik derecesi (1) ile ikinci kümeye (C2) ait oldukları görülmektedir. Buna göre K-ortalamlar kümeleme algoritması, iki kümeye de (C1 ve C2'ye) eşit uzaklıkta olmalarına karşın $g15$ ve $g16$ 'yı kesin ya da sert bir yaklaşımla C2'nin üyesi olarak atamaktadır. K-ortalamlar gibi sert kümeleme (hard clustering) veya katı kümeleme (crisp clustering) algoritmaları nesnelere üyelik dereceleri ya 0 ya da 1 olarak belirlemektedir. Herhangi bir nesnenin bir kümeye üyeliği 1 (%100) ise diğer kümelere üyeliği 0 (%0) olmaktadır. K-ortalamlar algoritmasıyla X16 veri seti için küme sayısı 3 olarak verildiğinde ise (Çizelge 3.5.), $g15$ ve $g16$ nesnelere farklı bir kümede (C3) yer almışlardır. Daha öncekine göre ($c=2$

kümelemesi) bu iki nesnenin diğer nesnelere ayrı kümelenebileceği iyi olsa da nesnelere küme merkezine göre uzaklıkları itibarıyla sorun hala devam etmektedir. Küme sayısı 4 olduğunda (Çizelge 3.6.), g_{15} ve g_{16} 'nın her biri ayrı bir küme olarak (C3 ve C4) değerlendirilmektedir. Bu daha önceki kümelemelere göre daha kabul edilebilir bir sonuç olmakla birlikte bu kümelere (C3 ve C4) tek nesnenin yer alması nedeniyle aykırı değer veya gürültü probleminden kaynaklanıp kaynaklanmadığı konusunda bilgi edinilememektedir. Küme sayısı 5 olduğunda (Çizelge 3.7.), Şekil 3.4'de sağ taraftaki g_9 ve g_{10} nesnelere ayrı bir küme daha oluşturulmuş ve doğal kümeleme yapısı bozulmaya başlamıştır.

Yukarıdaki açıklamalara göre, K-ortalama algoritması gibi sert kümeleme algoritmaları yukarıda açıklanan sorunlar söz konusu olduğundan kümeleme başarımı azalmaktadır. Bu soruna çözüm olarak yumuşak kümeleme algoritmalarının kullanılması yoluna gidilmektedir. Yumuşak kümeleme (soft clustering) nesnelere her bir küme için farklı üyelik dereceleriyle üye olabilmektedir. Yumuşak kümeleme olasılıklı, olabilirlikli veya bunların çeşitli varyantları şeklinde yaklaşımlar söz konusudur İlk geliştirilen ve en tanınmış algoritmalar olan Bulanık C-ortalama kümeleme (Fuzzy c-means) algoritması ile X16 veri seti farklı 4 küme sayısı ($c=2, \dots, 5$) için analiz edildiğinde üyelik matrisleri Çizelge 3.8-3.11'de gösterildiği gibi elde edilmektedir.

Çizelge 3.8. BCO'na Göre C=2 Küme Sayısında Üyelik Dereceleri Matrisi

Nesne	C1	C2
g1	0.001749	0.998251
g2	0.001519	0.998481
g3	0.012101	0.987899
g4	0.009219	0.990781
g5	0.021269	0.978731
g6	0.008649	0.991351
g7	0.003079	0.996921
g8	0.998251	0.001749
g9	0.990781	0.009219
g10	0.978731	0.021269
g11	0.998481	0.001519
g12	0.987899	0.012101
g13	0.991351	0.008649
g14	0.996921	0.003079
g15	0.500000	0.500000
g16	0.500000	0.500000

Çizelge 3.8'de iki küme için ($c=2$), BCO algoritması ile yapılan kümeleme analizinden elde edilen üyelik matrisi incelendiğinde nesnelerin kümelerden herhangi birine tam üye olmadığı ancak farklı üyelik dereceleriyle üye oldukları görülmektedir. Örneğin ilk nesne ($g1$ nesnesi) C1 kümesine 0.001749, C2 kümesine ise 0.998251 üyelik derecesi ile üye durumdadır. Sert kümelemede kümelerden herhangi birine tam üye olmaya zorlanan $g15$ ve $g16$ nesnelere bakıldığında ise bunların C1 ve C2 kümelerine 0.50 üyelik derecesi ile üye oldukları görülmektedir. Yani bir yumuşak kümeleme algoritması olarak BCO, C1 ve C2 kümelerine eşit uzaklıkta olan bu iki nesnenin kümelere üyelik derecesini 0.50 olarak belirlemiş ve kümelerden herhangi birine tam üye olmaya

zorlamamıştır. Bu tür algoritmalara bu nedenle yumuşak kümeleme algoritmaları denilmektedir.

Çizelge 3.9. BCO'na Göre C=3 Küme Sayısında Üyelik Dereceleri Matrisi

Nesne	C1	C2	C3
g1	0.000165	0.999735	0.000099
g2	0.005168	0.991640	0.003192
g3	0.026381	0.956677	0.016942
g4	0.007879	0.987453	0.004668
g5	0.023950	0.961978	0.014072
g6	0.005939	0.990037	0.004024
g7	0.007602	0.988374	0.004024
g8	0.000165	0.000099	0.999735
g9	0.007879	0.004668	0.987453
g10	0.023950	0.014072	0.961978
g11	0.005168	0.003192	0.991640
g12	0.026381	0.016942	0.956677
g13	0.005939	0.004024	0.990037
g14	0.007602	0.004024	0.988374
g15	0.261038	0.369481	0.369481
g16	0.994469	0.002766	0.002766

X16 veri seti için küme sayısı 3 ($c=3$) olarak alındığında, elde edilen üyelik dereceleri matrisi Çizelge 3.9'da görülmektedir. Burada $g16$ gözleminin tek elemanlı C1 kümesinde ayrı bir küme elemanı olduğu ancak $g15$ 'in 0.369481 üyelik dereceleriyle C2 ve C3 kümelerine üye olduğu görülmektedir.

Çizelge 3.10. BCO'na Göre C=4 Küme Sayısında Üyelik Dereceleri Matrisi

Nesne	C1	C2	C3	C4
g1	0.000036	0.999941	0.000014	0.000009
g2	0.021659	0.966510	0.007145	0.004687
g3	0.099680	0.855660	0.026476	0.018184
g4	0.010980	0.981034	0.004926	0.003060
g5	0.035651	0.935536	0.017867	0.010945
g6	0.015035	0.975888	0.005297	0.003780
g7	0.015039	0.974419	0.006761	0.003781
g8	0.000036	0.000009	0.000014	0.999941
g9	0.010980	0.003060	0.004926	0.981034
g10	0.035651	0.010945	0.017867	0.935536
g11	0.021659	0.004687	0.007145	0.966510
g12	0.099680	0.018184	0.026476	0.855660
g13	0.015035	0.003780	0.005297	0.975888
g14	0.015039	0.003781	0.006761	0.974419
g15	1.000000	0.000000	0.000000	0.000000
g16	0.000005	0.000003	0.999988	0.000003

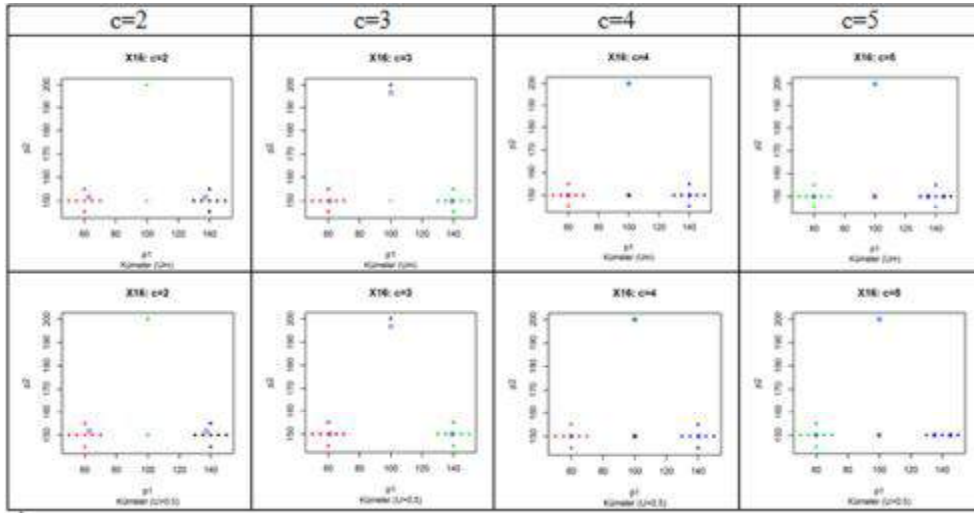
Küme sayısının 4 ($c=4$) olması durumunda, $g15$ ve $g16$ ayrı kümelerde yer almışlar ve diğer nesnelere de C2 ve C4 kümelerine yüksek üyelik dereceleri ile üye olmuşlardır. Bu durum, X16 veri setinde gürültü olmadığını varsaydığımızda 4 doğal kümenin bulunduğu işaret etmektedir. Nitekim Çizelge 3.12'de görülen indeks değerlerine bakıldığında, diğer indekslere göre 3 olarak önerilmesine karşın PE ve MPC indekslerine göre veri setindeki küme sayısının 4 olduğu ($c=4$) görülmektedir.

Çizelge 3.11. BCO'na Göre C=5 Küme Sayısında Üyelik Dereceleri Matrisi

Nesne	C1	C2	C3	C4	C5
g1	0.000010	0.000047	0.999912	0.000013	0.000018
g2	0.004156	0.022256	0.960959	0.005444	0.007185
g3	0.015631	0.100520	0.836938	0.020876	0.026035
g4	0.002691	0.010976	0.978076	0.003414	0.004843
g5	0.009670	0.035538	0.925137	0.012109	0.017547
g6	0.003346	0.015252	0.971822	0.004305	0.005275
g7	0.003346	0.015257	0.970357	0.004306	0.006734
g8	0.474638	0.007752	0.001960	0.512566	0.003083
g9	0.999726	0.000012	0.000003	0.000253	0.000006
g10	0.891664	0.008241	0.002565	0.093347	0.004183
g11	0.000013	0.000001	0.000000	0.999986	0.000000
g12	0.095942	0.023914	0.004464	0.869188	0.006492
g13	0.478458	0.014924	0.003816	0.497459	0.005342
g14	0.477770	0.014910	0.003813	0.496698	0.006809
g15	0.000064	0.999695	0.000081	0.000107	0.000053
g16	0.000001	0.000002	0.000001	0.000001	0.999994

X16 veri setindeki doğal küme sayısından daha yüksek bir küme sayısı için yapılan, örneğin 5 küme için yapılan ($c=5$) kümelemesinden elde edilen üyelik matrisinin gösterildiği Çizelge 3.11 incelendiğinde $g9$ ve $g10$ nesnelерinin ayrı bir küme (C1) kümesi içinde yer aldıkları anlaşılmaktadır. Yine aynı Çizelgede $g13$ ve $g14$ için üyelik derecelerine bakıldığında tüm üyelik derecelerinin 0.50 altında oldukları görülmektedir. Bu durumda, örneğin $g13$ 'ün en yüksek üyeliği C4 kümesine olduğundan bu küme içinde yer alabilir. Ancak üyelik dereceleri için belli bir eşik değeri belirlendiğinde bu nesne belli bir küme içinde değerlendirilemeyebilir. Şekil 3.5'de X16 veri seti için farklı küme sayılarına göre BCO algoritması ile elde edilen kümeler görülmektedir. Buradaki grafiklerin ilk sırasında maksimum üyelik derecelerine göre çizilen kümeler, ikinci sırasında ise

üyelik derecelerinin 0.50 üstünde olmasına göre çizilen kümeler bulunmaktadır. Şekilde $c=2$ için üyelik derecesinin 0.50 üstünde bulunması çizilen kümelere bakıldığında $g15$ ve $g16$ 'nın hiçbir kümede bulunmadıkları ve aykırı değer olarak yorumlanmaları gerekmektedir. Keza $c=5$ için çizilen grafiklere bakıldığında $g13$ ve $g14$ gözlemlerinin $C4$ 'e üyeliklerinin 0.50 altında olması nedeniyle grafikte gösterilmediği ve bunların bu kümeleme için atıpk oldukları söylenebilir.



Şekil 3.5. X16 Veri Setinde Farklı Küme Sayılarına Göre BCO Algoritması Kümeleme Sonuçları

Çizelge 3.12. X16 veri setinde farklı küme sayılarına göre BCO kümelemesine göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{PC}	V_{MPC}	V_{Fsil}
$c=2$	0.037	0.843	1.093	0.102	0.923	0.847	0.802
$c=3$	0.020	0.721	0.924	0.078	0.928	0.892	0.831
$c=4$	0.021	1.130	1.392	0.077	0.927	0.903	0.814
$c=5$	0.231	16.519	19.688	0.202	0.834	0.793	0.635

Yukarıda yapılan açıklamalara göre nesnelerin belli bir kümeye ait olup olmayacaklarına karar verebilmek için serpilme diyagramları yol gösterici olsa da

bu değerlendirme sezgisel olduğu kadar zaman alıcı da bir işlemdir. Dahası pratikte kullanılan veri setleri, X16 veri setinde olduğu gibi yalnız iki özellikli olmayıp çok sayıda özellik içeren çok boyutlu veri setleridir. Böylesi çok boyutluluk durumunda küme serpilme diyagramları üzerinde değerlendirme yapmak yerine nesnelerin kümelere üyeliği konusunda daha bilgi verici olan, bir başka deyişle bir nesnenin herhangi bir kümeye üye olabilirliliğini gösteren başka yöntem ve tekniklere ihtiyaç duyulmaktadır. Krishnapuram ve Keller (1993) tarafından açıklanan Olabilirlikli c-ortalamlar (OCO) kümelemesi bu soruya yanıt vermektedir. Çizelge 3.13'de X16 veri setinde iki küme için (c=2) yapılan OCO kümeleme algoritması sonuçları görülmektedir.

Çizelge 3.13. OCO Kümelemesine Göre C=2 Küme Sayısında Tipiklik Dereceleri Matrisi

Nesne	C1	C2
g1	0.033977	0.999233
g2	0.038504	0.913686
g3	0.043984	0.707915
g4	0.030197	0.883785
g5	0.027009	0.672650
g6	0.033846	0.897414
g7	0.033850	0.899561
g8	0.999233	0.033977
g9	0.883785	0.030197
g10	0.672650	0.027009
g11	0.913686	0.038504
g12	0.707915	0.043984
g13	0.897414	0.033846
g14	0.899561	0.033850
g15	0.124464	0.124464
g16	0.051968	0.051968

Çizelge 3.13’de verilen değerler, BCO algoritması ile elde edilen üyelik dereceleri matrisi değil, tipiklik (typicality) dereceleri matrisidir. Tipiklik dereceleri bir nesnenin belli bir kümede bulunma olasılığını; ya da aidiyet derecesini gösteren değerlerdir. Çizelge 3.13’den *g15* ve *g16* nesnelere C1 ve C2’ye tipiklik derecelerinin sırasıyla 0.124464 ve 0.051968 oldukları görülmektedir. Oysa bu nesnelere BCO algoritması ile bu kümelere üyelik dereceleri 0.50 ve 0.50 olarak birbirine eşit bulunmuşlardı. OCO algoritmasından elde edilen tipiklik derecelerine göre *g15* nesnesi *g16*’ya göre söz konusu küme merkezlerine daha yakın olduğu için daha yüksek tipiklik derecesine sahip olmuştur. Ayrıca her iki nesnenin tipiklik dereceleri diğerlerine göre çok düşük bulunması bu değerlerin 2 küme için yapılan analizde birer aykırı değer veya gürültü olarak yorumlanması gerektiğine işaret etmektedir. Bu durumda olasılıklı yumuşak kümelemede tipiklik dereceleri BCO algoritması ile karşılaşılan gürültü ve aykırı değer sorunlarına çözüm sağlamaktadır.

Çizelge 3.14. OCO Kümelemesine Göre C=3 Küme Sayısında Tipiklik Dereceleri Matrisi

Nesne	C1	C2	C3
g1	0.038885	0.999863	0.011225
g2	0.042763	0.751119	0.012754
g3	0.046810	0.425088	0.014616
g4	0.035260	0.735929	0.009955
g5	0.031934	0.415384	0.008888
g6	0.034599	0.742972	0.011182
g7	0.043779	0.743922	0.011182
g8	0.038885	0.011225	0.999863
g9	0.035260	0.009955	0.735929
g10	0.031934	0.008888	0.415384
g11	0.042763	0.012754	0.751119
g12	0.046810	0.014616	0.425088
g13	0.034599	0.011182	0.742972
g14	0.043779	0.011182	0.743922
g15	0.063431	0.043541	0.043541
g16	0.989953	0.017405	0.017405

Çizelge 3.14'de X16 veri setinde OCO ile 3 küme ($c=3$) için yapılan kümelemeden elde edilen tipiklik dereceleri incelendiğinde $g16$ 'nın yüksek bir tipiklikle C1 kümesini oluşturduğu, ancak $g15$ 'in herhangi bir kümeye ait olmadığı görülmektedir.

Çizelge 3.15. OCO kümelemesine göre $c=4$ küme sayısında tipiklik dereceleri matrisi

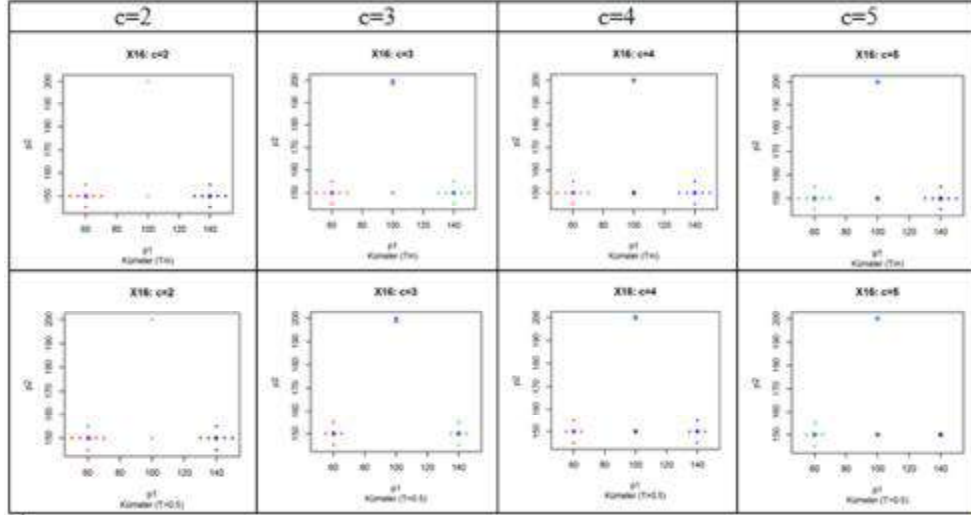
Nesne	C1	C2	C3	C4
g1	0.016409	0.999942	0.002219	0.006268
g2	0.021326	0.621838	0.002441	0.007126
g3	0.028805	0.289327	0.002674	0.008173
g4	0.013010	0.612721	0.002011	0.005556
g5	0.010564	0.285376	0.001820	0.004958
g6	0.016161	0.616952	0.001967	0.006244
g7	0.016162	0.617541	0.002509	0.006244
g8	0.016409	0.006268	0.002219	0.999942
g9	0.013010	0.005556	0.002011	0.612721
g10	0.010564	0.004958	0.001820	0.285376
g11	0.021326	0.007126	0.002441	0.621838
g12	0.028805	0.008173	0.002674	0.289327
g13	0.016161	0.006244	0.001967	0.616952
g14	0.016162	0.006244	0.002509	0.617541
g15	0.999999	0.024638	0.003634	0.024638
g16	0.010567	0.009748	0.999998	0.009748

Çizelge 3.15'de 4 küme için analiz yapıldığında $g15$ için C1 ve $g16$ için C3 kümeleri oluşturulduğu görülmektedir. Keza Çizelge 3.17'deki genelleştirilmiş ve uzatılmış indeks değerlerinden bazılarına göre X16 veri setinde 4 doğal küme olduğu tahmin edilmektedir.

Çizelge 3.16. OCO kümelemesine göre $c=5$ küme sayısında tipiklik dereceleri matrisi

Nesne	C1	C2	C3	C4	C5
g1	0.003260	0.009151	0.983025	0.003368	0.001196
g2	0.005912	0.019004	0.966878	0.006107	0.002099
g3	0.013882	0.052707	0.914366	0.014340	0.004705
g4	0.004722	0.011841	0.976788	0.004878	0.001771
g5	0.008933	0.020365	0.958075	0.009228	0.003399
g6	0.005252	0.014573	0.973035	0.005425	0.001715
g7	0.005244	0.014553	0.972600	0.005418	0.002184
g8	0.495868	0.004616	0.003045	0.495868	0.000603
g9	0.488327	0.007856	0.005802	0.496839	0.001175
g10	0.477421	0.016310	0.013253	0.490294	0.002723
g11	0.484951	0.012735	0.007333	0.493575	0.001406
g12	0.459886	0.042926	0.020917	0.472439	0.003832
g13	0.487048	0.009711	0.006480	0.495618	0.001143
g14	0.486895	0.009703	0.006474	0.495472	0.001456
g15	0.012481	0.949806	0.022932	0.012889	0.001894
g16	0.005036	0.005826	0.009307	0.005201	0.974629

Çizelge 3.16'da küme sayısı 5 olarak ($c=5$) alındığında $g15$ ve $g16$ 'nın sırasıyla C2 ve C5 kümelerinde yer aldıkları; $g8-g14$ nesnelere için tipiklik derecelerinin azaldığı ve 0.50'nin altına düştükleri görülmektedir. Bu kümeleme sonucunda elde edilen ve Şekil 3.6'da verilen küme grafikleri incelendiğinde maksimum tipiklik derecelerine göre kümelerin elde edildiği ancak 0.50 tipiklik eşik değeri olarak belirlendiğinde C1 ve C4 kümelerinde üye bulunmadığı görülmektedir. Bu durum, kümeleme kalitesinde düşmeye işaret etmekte olup nitekim Çizelge 3.17'den uzatılmış indeks değerleri incelendiğinde tüm indeksler itibarıyla $c=5$ kümelemesinin önerilmediği açıkça görülmektedir.



Şekil 3.6. X16 veri seti için küme sayısına göre OCO algoritması ile kümeleme sonuçları

Çizelge 3.17. X16 veri seti için küme sayısına göre OCO algoritmasından elde edilen genelleştirilmiş ve uzatılmış indeks değerleri

	V_{GXB}	V_{GKwon}	V_{GPE}	V_{EXB}	V_{EKwon}	V_{EPE}	V_{EPESIM}
c=2	0.037	0.843	0.102	0.039	0.879	0.473	0.187
c=3	0.020	0.721	0.078	0.023	0.773	0.448	0.145
c=4	0.021	1.130	0.077	0.026	1.218	0.324	0.100
c=5	0.231	16.519	0.202	0.329e8	0.162e10	0.370	0.115

3.2.2. Bulanık Ve Olabilirlikli Kümeleme Algoritmaları

3.2.2.1. Bulanık C-Ortalamlar (BCO) Algoritması

Bulanık kümeleme 1960'lı yıllarda Zadeh'in bulanık küme ile ilgili çalışmalarına dayanmaktadır (Zadeh, 1965). Döring ve ark. (2006) bulanıklık ve olabilirlik kavramlarının geniş ve anlaşılabilir şekilde derlemesini yapmışlardır. Kümeleme analizinde bulanıklık kavramı Dunn (1973) ve Bezdek (1981)'in çalışmaları ile başlamıştır. Bu nedenle en yaygın olarak kullanılan bölümleyici bulanık kümeleme algoritması Bezdek (1981)'nin Bulanık C-Ortalamlar (BCO) algoritması olup olasılıklı yaklaşıma dayanan bir algoritmadır. Tartılı grup içi hata

kareler toplamının en küçüklenmesine dayanan BCO algoritmasının amaç fonksiyonu Eşitlik 3.4'deki gibi yazılabilir:

$$J_{BCO}(X; U, V) = \sum_{k=1}^n \sum_{i=1}^c u_{ik}^m d^2(x_k, v_i) \quad (3.4)$$

Burada:

x_k : k . p -boyutlu veri vektörünü

v_i : i . kümenin merkez prototipini

u_{ik} x_k 'nin i . kümeye üyelik derecesini

m : bulanıklaştırma parametresini

$d(x_k, v_i)$, x_k vektörü ile v_i küme merkezi arasındaki uzaklığı

n , veri nesnelere sayısını

c , küme sayısını göstermektedir.

BCO amaç fonksiyonunda (Eşitlik 3.4) m , üyeliklerin bulanık kümeler arasında paylaşımını kontrol etmek için kullanılan bir bulanıklaştırma parametresidir. BCO algoritması, sert K-ortalama algoritmasından türetildiğinden üyelikler m gibi bir üsle tartılmak suretiyle bulanıklaştırılmaktadır. $m \in [1, \infty]$ olup hali hazırda bu parametrenin optimal seçimi için teorik bir dayanak önerilmemiş olduğundan çalışmaların çoğunda 2 olarak seçilmektedir (Menárd ve ark., 2003).

Küme merkezleri ile veri noktaları arasındaki uzaklıklar genellikle Öklid normunda hesaplanmakta olup aşağıdaki gibi formüle edilebilir.

$$d^2(x_k, v_i) = \|x_k - v_i\|^2 = (x_k - v_i)^T (x_k - v_i)$$

BCO amaç fonksiyonu v_i küme merkezleri (3.5) ve u_{ik} üyelik derecelerinin (3.6) iteratif işlemlerle güncellenmesi ile minimize edilmektedir.

$$v_i = \frac{\sum_{k=1}^n u_{ik}^m x_k}{\sum_{k=1}^n u_{ik}^m} ; 1 \leq i \leq c; 1 \leq k \leq n \quad (3.5)$$

$$u_{ik} = \frac{1}{\sum_{j=1}^c \left(\frac{d_{ik}}{d_{ij}} \right)^{\frac{2}{m-1}}} ; 1 \leq i \leq c \quad (3.6)$$

Burada üyelik dereceleri (u_{ik}) için aşağıdaki kısıtlar söz konusudur:

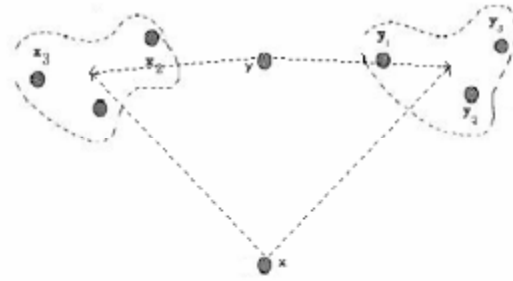
$$0 < \sum_{k=1}^n u_{ik} < n, \quad u_{ik} \in [0, 1] \quad \forall i, k, \quad \sum_{i=1}^c u_{ik} = 1, \quad \forall k = 1, \dots, n \quad (3.7)$$

En küçük kareler yaklaşımlarına dayanan diğer algoritmalar gibi BCO ve türevi algoritmalar gürültüye ve aykırı değerlere karşı duyarlıdırlar. Gürültü noktalarıyla ortaya çıkan düşük üyelik dereceleri gürültü kirliliğini ifade ediyor olsa da (3.6)'dan görüleceği üzere (3.7)'deki üçüncü kısıt nedeniyle üyelik dereceleri olasılık gösteren relatif sayılardır (Dave ve Krishnapuram, 1997). Bu, gürültü noktaları ve aykırı değerlerin en azından kalan kümelerin tümünün $1/c$ 'ne denk bir değer olacağı anlamına gelir. Bir noktanın bir kümeye üyelik değerinde herhangi bir artış olduğunda, bu noktanın başka herhangi bir kümeye ait olma derecesinde bir azalma olacaktır. Bu durumda, gürültülü noktaların yüksek üyelik derecelerine sahip olabileceği ve bu nedenle prototip parametre tahminlerini etkileyebileceğini göstermektedir.

Krishnapuram ve Keller (1993) tarafından açıklandığı gibi BCO'nun ikinci önemli sorunu, (3.7)'deki son kısıt nedeniyledir. Bu, kümelerin üyelik paylaşma dereceleri olarak yorumlanır, ancak bir noktanın bir sınıfa ait olma olabilirliğinin derecesi değildir. Böylesi durumlarda tipiklik dereceleri (typicality) veya kümelere aidiyet olabilirliği ile çalışmak klasik bulanık kümelemeye karşı daha uygun olabilir.

3.2.2.2. Olabilirlikli C-Ortalamlar (OCO) Algoritması

Yukarıdaki (3.7) nolu kısıt üyelik derecelerinin olasılıklı değerler olmasını zorladığından üyeliklerin tipiklik veya uyumlulukların temsili için uygun değildir. Şekil 3.7’de x noktası bir aykırı değeri ve y noktası ise gürültüyü temsil etmektedir (Wachs ve ark., 2004). Bu örnekte hem x hem de y kümelerin her birine eşit uzaklıkta olmasına karşın y noktası x ’e göre küme merkezlerinden daha uzakta bulunmaktadır.



Şekil 3.7. Gürültü ve aykırı değerlerin temsili gösterimi (Wachs ve ark.,2004)

Sezgisel olarak böyle bir durumda x ve y noktasının kümelerden herhangi birine üyelik derecesinin diğer kümeye üyeliğinden yüksek olmaması gerekir. Ancak y noktasına göre daha uzakta olan x noktasının her iki kümeye üyelik derecesinin daha küçük olması beklenir. Buna karşın BCO kümelemesi ile x ve y noktalarının her ikisinin de kümelere üyelik dereceleri 0.5 olarak hesaplanmaktadır (Krishnapuram ve Keller, 1996). Bu sonuca göre, BCO ile elde edilecek üyelik derecelerinin kümelere “ait olma” veya “aidiyet” derecesini temsil edemedikleri; yani aykırı değer ve gürültü arasında bir ayırım yapamadıkları anlaşılmaktadır.

Olabilirlikli C-Ortalamlar (OCO) algoritması ile BCO’nun amaç fonksiyonunda (3.4) üyelik dereceleri toplamının 1’e eşit olma kısıtı (3.7) yumuşatılarak bu soruna çözüm sağlanmaktadır. OCO’da küme merkezine yakın veri noktaları “tipik”, ancak uzak veri noktaları “atipik” olarak görülerek veri

noktalarının kümelere üyelikleri $[0,1]$ aralığında “tipiklik” dereceleri ile belirlenir. Sıfır veya sıfıra yakın tipiklik değerleri ise “gürültü” olarak değerlendirilir.

OCO'nun amaç fonksiyonuna tüm i 'ler için $u_{ik} = \mathbf{0}$ şeklinde önemi olmayan bir çözümden kaçınmak üzere u_{ik} 'leri mümkün olduğunca büyük olmaya zorlayan bir cezalandırma terimi eklenmektedir. Böylece (3.4)'deki BCO amaç fonksiyonu, OCO amaç fonksiyonuna (3.9) dönüştürülmektedir.

$$J_{OCO}(X; T, V) = \sum_{k=1}^n \sum_{i=1}^c t_{ik}^m d^2(x_k, v_i) + \sum_{i=1}^c \eta_i \sum_{k=1}^n (1 - t_{ik})^m \quad (3.8)$$

Burada:

t_{ik}, x_k 'nin i . küme için tipiklik derecesi

$\eta_i > \mathbf{0}; 1 \leq i \leq c$

$t_{ik} \in [0, 1]$

kısıtları söz konusu olup tipiklik ve küme prototipleri güncelleme eşitlikleri:

$$t_{ik} = \frac{1}{1 + \left[\frac{d^2(x_k, v_i)}{\eta_i} \right]^{1/(m-1)}} ; 1 \leq i \leq c; 1 \leq k \leq n \quad (3.9.1)$$

$$v_i = \frac{\sum_{k=1}^n t_{ik}^m x_k}{\sum_{k=1}^n t_{ik}^m} ; 1 \leq i \leq c; 1 \leq k \leq n \quad (3.9.2)$$

şeklinde oluşturulmaktadır.

Burada t_{ik} değerleri, veri noktasının kümelere ait olabilme derecelerini yani tipiklik değerlerini göstermektedir. (Krishnapuram ve Keller, 1996) kümeleme için iyi bir başlangıç sağlamak üzere η_i 'leri hesaplamak için BCO'dan elde edilen

üyelik derecelerinin kullanılmasını önermişlerdir. η_i parametresi her bir küme için ayrı olarak değerlendirilmekte olup üyelik derecesi 0.5'teki uzaklığı belirtmektedir.

$$\eta_i = K \sum_{k=1}^n t_{ik}^m d^2(x_k, v_i) / \sum_{k=1}^n t_{ik}^m ; K > 0 \quad (3.10)$$

Eşitlik (3.10)'da K sıfırdan büyük bir sayı olup genellikle 1 olarak seçilmektedir. $\eta = \{\eta_1, \dots, \eta_c\}$ olup η_i i . küme için cezalandırma terimi olup t_{ik} 'yi 1'e getirmeye çalışmaktadır. $\eta_i = 0$ olduğunda OCO'nun BCO ile aynı olduğu görülecektir.

OCO algoritması gürültü ve aykırı değerler söz konusu olduğunda küme merkezlerini bulmada BCO'ya göre daha sağlam ve etkin bir algoritmadır. Bununla birlikte BCO gibi başlangıç küme merkezleri seçimine karşı duyarlıdır. Bu algoritmanın en önemli sorunu (3.8)'deki amaç fonksiyon gerçekten minimize edildiğinde ortaya çıkmaktadır. Tipiklik matrisinin (T) satır ve sütunları birbirinden bağımsız olduğundan çakışan kümeler (coincident clusters) oluşabilmektedir (Grover, 2014). Eğer her bir satır başlatma sırasında yeterince farklı değilse bu sorun ortaya çıkmaktadır. Bu sorun tipiklik derecelerinde (3.9.1) sadece nokta ile küme merkezi arasındaki uzaklığa bağlı olunması, yani diğer kümelere göreceli uzaklıkların dikkate alınmamasından kaynaklanmaktadır. Algoritma, (3.8)'deki OCO amaç fonksiyonu bir lokal minimuma yaklaştırdığından normal olarak merkezlerin sadece bir kısmı çakışmakla birlikte hala arzu edilmeyen yanları bulunmaktadır (Barni ve ark, 1996).

3.2.2.3. Karma C-Ortalamlar Algoritmaları (Hibrit Algoritmalar)

BCO ve OCO'nun önerilmesinden sonra her iki algoritmadaki sorunlara çözüm bulabilmek için bu algoritmaların çeşitli şekillerde birleştirilmesi ile karma algoritmalar (hibrit algoritmalar) geliştirilmiştir.

3.2.2.3.(1). Bulanık Olabilirlikli C-Ortalamalar (BOCO) Algoritması

Pal ve ark. (1997) olabilirlikli ve olasılıklı yaklaşımlarını birleştiren ve Bulanık Olabilirlikli C-Ortalamalar (BOCO) olarak adlandırılan karma bir algoritma önermişlerdir. BOCO, bir alternatif optimizasyon işlemi vasıtasıyla hem üyeliklerin hem de tipikliklerin elde edilmesini ve birlikte kullanılmasını sağlamaktadır. BOCO algoritması ile BCO'da karşılaşılan gürültü sorununa çözüm sağlanırken OCO'daki çakışan kümeler sorunu da ortadan kaldırılmaktadır. Eşitlik (3.11)'deki BOCO amaç fonksiyonu, tüm c küme merkezleri yerine n veri noktalarının tümüne dayanarak, (3.4)'deki BCO amaç fonksiyonundaki üyelik derecesine bir olabilirlik terimi eklenerek elde edilmektedir.

$$J_{BOCO}(\mathbf{X}; \mathbf{U}, \mathbf{T}, \mathbf{V}) = \sum_{k=1}^n \sum_{i=1}^c (u_{ik}^m + t_{ik}^\eta) d^2(x_k, v_i) \quad (3.11)$$

Burada:

$$m > 0 \text{ ve } \eta > 0$$

olup üyelik dereceleriyle ilgili kısıtlar (3.6)'da ve (3.7)'deki ile aynı olup (3.14)'deki kısıtın sağlanması da gereklidir. BOCO'da hem üyelik hem de tipiklik değerleri hesaplanmakta olup sırasıyla (3.12.1) ve (3.12.2)'deki eşitliklerle güncellenmektedir.

$$u_{ik} = \left(\sum_{k=1}^c \left(\frac{d(x_k, v_i)}{d(x_k, v_j)} \right)^{2/(m-1)} \right)^{-1} \quad \forall i, k \quad (3.12.1)$$

$$t_{ik} = \left(\sum_{k=1}^n \left(\frac{d^2(x_k, v_i)}{d^2(x_k, v_j)} \right)^{2/(\eta-1)} \right)^{-1} \quad \forall i, k \quad (3.12.2)$$

$$v_i = \frac{\sum_{k=1}^n (u_{ik}^m + t_{ik}^\eta) x_k}{\sum_{k=1}^n (u_{ik}^m + t_{ik}^\eta)}, \quad \forall i \quad (3.13)$$

$$\sum_{k=1}^n t_{ik} = 1, \quad \forall i, k \quad (3.14)$$

Küme prototipleri hem üyelik hem de tipiklik değerleri kullanılarak (3.13)'deki gibi hesaplanmaktadır. BOCO algoritmasında karşılaşılan sorunlardan biri m ve η parametreleri için geçerli optimal bir aralık tanımlı olmamasıdır.

3.2.2.3.(2). Olabilirlikli Bulanık C-Ortalamlar (OBCO) Algoritması

BOCO algoritmasında satır toplamlarından kaynaklanan sorunlar söz konusudur. Tüm veri noktalarının bir kümeye tipiklik değerlerinin toplamının 1'e eşit olması kısıtı (3.14) özellikle büyük veri setlerinde sorunlara yol açabilmektedir. Pal ve ark. (2005) bu kısıtı yumuşatarak BCO'nın amaç fonksiyonu (3.4) ile OCO'nın amaç fonksiyonunun (3.8) bir karması olan Olabilirlikli Bulanık C-Ortalamlar (OBCO) algoritmasını önermişlerdir.

$$J_{OBCO}(\mathbf{X}; \mathbf{U}, \mathbf{T}, \mathbf{V}) = \sum_{k=1}^n \sum_{i=1}^c (a u_{ik}^m + b t_{ik}^\eta) d^2(\mathbf{x}_k, \mathbf{v}_i) + \sum_{i=1}^c \gamma_i \sum_{k=1}^n (1 - t_{ik})^\eta \quad (3.15)$$

Eşitlik (3.15)'deki amaç fonksiyon için kısıtlamalar aşağıdaki gibidir.

$$\sum_{i=1}^c u_{ik} = 1, \quad \forall k; \quad 0 \leq u_{ik}, t_{ik} \leq 1; \quad a > 0; \quad b > 0; \quad m > 1; \quad \eta > 1$$

Eşitlik (3.15)'de u_{ik} BCO'daki bulanık üyelikleri ve t_{ik} OCO'daki tipiklik değerlerini gösterdiğinden a ve b parametreleri sırasıyla bulanık üyelik dereceleri ve tipiklik değerlerinin göreceli önemliliğini ifade eden katsayılardır. Bu durumda a artarken b aynı miktarda azalacak demektir. b 'nin büyüklüğü optimal tipiklik değerine bağlı olduğundan $a + b = 1$ kısıtlaması yapıldığında modelin esnekliği

kaybolacaktır. OBCO algoritması ile her bir iteratif adımda küme prototipleri, üyelik ve tipiklik değerleri (3.16) ve (3.17)'deki gibi güncellenerek amaç fonksiyon minimize edilmektedir.

$$u_{ik} = \left(\sum_{i=1}^c \left(\frac{d(x_k, v_i)}{d(x_k, v_j)} \right)^{\frac{2}{m-1}} \right)^{-1} ; 1 \leq i \leq c ; 1 \leq k \leq n \quad (3.16)$$

$$t_{ik} = \frac{1}{1 + \left(\frac{b}{v_i} d^2(x_k, v_i) \right)^{\frac{1}{\eta-1}}} ; 1 \leq i \leq c ; 1 \leq k \leq n \quad (3.17)$$

$$v_i = \frac{\sum_{k=1}^n ((a u_{ik}^m) + (b t_{ik}^\eta)) x_k}{\sum_{k=1}^n (a u_{ik}^m) + (b t_{ik}^\eta)} ; 1 \leq i \leq c \quad (3.18)$$

olarak tanımlanmıştır.

Eşitlikte (3.20) görüldüğü üzere a ve b 'yi karşılaştırırken a için büyük bir değer alındığında üyelik değerleri tipiklik değerlerine göre daha çok etkilenmiş olacaktır. Diğer yandan a için büyük bir değer kullanıldığında merkezler üye değerlerden çok etkilenmiş olacaktır. Böylece aykırı verileri azaltmak için b değeri a 'dan daha büyük olmak zorundadır. η 'nin kontrollü seçimi ile birbirine yakın etkiler elde edilebilir. Örneğin η için geniş ve daha küçük değerler kullanılırsa merkezler üzerinde aykırı gözlemlerin etkisi azaltılmış olacaktır. Büyük değerlerin etkisi üye merkezleri üzerinde azaldığında OBCO daha çok OCO gibi davranmış olur (Kumari ve ark, 2012).

3.2.2.4. Olabilirlikli C-Ortalamalar Kümeleme Algoritmasında İyileştirmeler

3.2.2.4.(1). İtmeli Olabilirlikli C-Ortalamalar (iOCO) Algoritması

BCO ve OCO'nun eksikliklerine çözüm olarak geliştirilen algoritmalarından bir diğeri İtmeli (repulsion) Olabilirlikli Kümeleme algoritmasıdır. Bir noktanın

kümeye tipikliği yalnız nesne ile küme merkezi arasındaki uzaklıkla belirlendiğinden OCO'nun (3.8)'deki amaç fonksiyonu ile izah edilen istenmeyen davranışı sadece eğer tüm merkezler çakışık ise doğru biçimde minimize edilebilirler. Bu yöntem zımnen (3.7)'deki son kısıtı kullanmaksızın (3.8)'e bir küme itme terimi ekleyerek kümeler arası uzaklıkları maksimize etmeye çalışırken kümeler-içi uzaklıkları minimize etmeyi amaçlar (Bezdek ve Pal, 1998).

$$J_{iOCO}(\mathbf{X}; \mathbf{T}, \mathbf{V}) =$$

$$\sum_{k=1}^n \sum_{i=1}^c t_{ik}^m d^2(x_k, v_i) + \sum_{i=1}^c \eta_i \sum_{k=1}^n (1 - t_{ik})^m + \gamma \sum_{i=1}^c \sum_{k=1, k \neq i}^c \frac{1}{d^2(v_i, v_k)} \quad (3.19)$$

Burada:

γ : Bir tartı (ağırlık) faktörü olup,

$$t_{ik} \in [0, 1], \quad \forall i \quad (3.20)$$

kısıtı sözkonusudur.

İtme terimi eğer kümeler yeterince yakın ise uygundur. Uzaklık artarken kümelerin çekimi ile telafi edildiklerinden etkileri daha az olmaktadır. Diğer yandan eğer kümeler yeterince ayrık iseler ve kümeler arası uzaklıklar yüksek değilse (ilk terim nedeniyle), küme çekimi sadece itme terimi ile telafi edilebilmektedir. Eşitlik (3.19)'un küme prototiplerine ilişkin minimizasyonu:

$$v_i = \frac{\sum_{k=1}^n t_{ik} x_k - \gamma \sum_{k=1, j \neq i}^c v_k \frac{1}{d^2(v_j, v_i)}}{\sum_{k=1}^n k - \gamma \sum_{k=1, k \neq i}^c \frac{1}{d^2(v_j, v_i)}} \quad \forall i \quad (3.21)$$

ile elde edilmektedir.

Herhangi bir iterasyonda bir veya daha fazla uzaklık sıfır olduğunda, $d^2(v_k, v_i) = 0$, tekillik sorunu ortaya çıkar. Böyle bir durumda (3.21)

hesaplanamaz. Bu olursa, T matrisinde tekil olmayan her bir sınıfa 0 (i hariç tüm sınıflar) ve i . sınıfa 1 atanır. OCO algoritması için benzer durum t_{ik} üyelik derecelerini güncellemek için Eşitlik 3.5'deki formül kullanarak elde edilir.

Eşitlik (3.19)'daki amaç fonksiyonu minimize etmek için tepkinim terimi olarak bir başka alternatif aynı araştırmacılar tarafından önerilmiştir.

$$J_{iOCO2}(\mathbf{X}; \mathbf{T}, \mathbf{V}) = \sum_{k=1}^n \sum_{i=1}^c t_{ik}^m d^2(x_k, v_i) + \sum_{i=1}^c \eta_i \sum_{k=1}^n (1 - t_{ik})^m + \gamma \sum_{i=1}^c \sum_{k=1, k \neq i}^c e^{-d^2(v_k, v_i)} \quad (3.22)$$

İki terim arasındaki başlıca fark kümeler arası uzaklıklar artarken tepki terimindeki azalmadır. Bu, olabilirlikli kümelemede görülen problemleri kaldıran bir tepki terimi ekleme stratejinden oluşur. γ tartı faktörü örneğin kümeler arası uzaklıkları maksimize ve kümeler içi uzaklıkları minimize ederek çekme ve itme güçlerini dengelemek için kullanılır. Daha öncekinde olduğu gibi, bu algoritmanın temel sorunu bir çözüm parametresi (bu durumda γ tartı faktörü) gerektirmesi olup gürültüden kaçınmak ve daha iyi bölümlenmeler/kümeler elde etmek üzere kullanılacak parametrenin doğru aralığı hakkında herhangi bir ipucu söz konusu değildir (Dave ve Krishnapuram, 1997).

3.2.2.4.(2). Yönetimsiz Olabilirlikli Bulanık C-Ortalamlar (uOBCO) Algoritması

Wu ve ark (2010) hem üyelik hem de tipiklik derecelerini eş zamanlı olarak üreten ve Yönetimsiz Olabilirlikli Bulanık C-Ortalamlar (Unsupervised Fuzzy and Possibilistic Clustering Algorithm) olarak adlandırdıkları bir karma kümeleme algoritması önermişlerdir. Yang ve Wu (2006)'nın Olabilirlikli Kümeleme Algoritması (OKA) ile karşılaşılan çakışan kümeler sorununa çözüm olarak önerilen Yönetimsiz Olabilirlikli Bulanık C-Ortalamlar (uOBCO); BCO'daki gürültüye duyarlılık ve OKA'daki çakışan kümeler sorununa çözüm

sağlayarak BCO ve OKA'nın avantajlarını birleştirmektedir. Bu algoritma, aynı amaçla geliştirilen OBCO'dan farklı olarak, BCO'nun çalıştırması sonucu hesaplanan *eta* vektörüne (η_i) hesaplanmasına ihtiyaç duymamaktadır. Bu nedenle hesaplama süresinin kısalması avantajı da sağlamaktadır.

$$J_{uOBCO}(\mathbf{X}; \mathbf{U}, \mathbf{V}) = \sum_{k=1}^n \sum_{i=1}^c (a u_{ik,BCO}^m + b u_{ik,OKA}^\eta) d^2(x_k, v_i) + \frac{\beta}{n^2 \sqrt{c}} \sum_{k=1}^n \sum_{i=1}^c (u_{ik,OKA}^\eta \log u_{ik,OKA}^\eta - u_{ik,OKA}^\eta) \quad (3.23)$$

Eşitlik (3.23)'deki amaç fonksiyondaki kısıtlamalar:

$$\sum_{i=1}^c u_{ik,BCO} = 1, \forall k ; 0 \leq u_{ik,BCO} \leq 1 ; a > 0 ; b > 0 ; m > 1 ; \eta > 1$$

olup burada $u_{ik,BCO}$, x_k 'nin i . kümeye bulanık üyelik derecesini ve $u_{ik,OKA}$ ise x_k 'nin i . kümeye tipiklik derecesini göstermektedir. m ve η değerleri diğer olabirlikli algoritmalarda olduğu üzere sırasıyla bulanıklık ve tipiklik için kullanılan katsayılar olup genelde 2 olarak seçilmektedirler. Amaç fonksiyondaki a ve b sabitleri sırasıyla bulanıklık ve tipiklik derecelerine verilecek göreceli önemi gösteren ağırlık katsayılarıdır. Eşitlik (3.23)'deki amaç fonksiyonda $b=0$ ise uOBCO amaç fonksiyonu BCO amaç fonksiyonuna dönüşmektedir. Uygulamada genellikle her iki katsayı birbirine eşit ve 1 ($a=b=1$) olarak kullanılmaktadır.

uOBCO algoritmasında amaç fonksiyonun minimize edilmesi için yürütülen iterasyonlarda üyelik dereceleri ve küme merkezleri aşağıda verilen eşitlikler (3.24.1, 3.24.2 ve 3.25) güncelleştirilir.

$$u_{ik,BCO} = \left(\sum_{j=1}^c \left(\frac{d(x_k, v_i)}{d(x_k, v_j)} \right)^{2/(m-1)} \right)^{-1} \quad \forall i, k \quad (3.24.1)$$

$$u_{ik,OKA} = \exp\left(\frac{b n \sqrt{c} d^2(x_k, v_i)}{\beta}\right) \quad \forall i, k \quad (3.24.2)$$

Yang ve Wu (2006) tarafından örneklem kovaryansı olarak adlandırılan (3.24.2)'deki β pozitif bir parametre olup (3.24.3)'deki gibi veri noktalarının özellik ortalamalarından sapmalarından hesaplanan varyansı göstermektedir.

$$\beta = \frac{1}{n} \sum_{k=1}^n d^2(x_k, \bar{x}) ; \bar{x} = \frac{1}{n} \sum_{k=1}^n x_k \quad (3.24.3)$$

Her iterasyon adımında küme merkezleri hem BCO hem de OKA dereceleri kullanılarak (3.25)'deki gibi güncellenmektedirler.

$$v_i = \frac{\sum_{k=1}^n (a u_{ik,BCO}^m + b u_{ik,OKA}^\eta) x_k}{\sum_{k=1}^n (a u_{ik,BCO}^m + b u_{ik,OKA}^\eta)}, \quad \forall i \quad (3.25)$$

3.3 Kümeleme Geçerlilik İndeksleri

Bir kümeleme algoritmasının ürettiği sonuçların doğruluğu ya da geçerliliği geçerlilik indeksleri (validity indices) ile değerlendirilir. Eğer herhangi bir veri setindeki kümelerin sayısı önceden biliniyorsa herhangi bir indeksin önermiş olduğu küme sayısı ile karşılaştırılır. Eğer algoritma analiz edilen veri setindeki küme sayısı ile aynı veya çok yakın küme değerleri öneriyorsa incelenen algoritmanın başarılı olduğu sonucuna varılır.

Kümeleme analizinde elde edilen kümeleme sonuçlarının geçerliliğini test etmek üzere çok sayıda indeks geliştirilmiştir. Kümeleme geçerlilik indeksleri temel olarak dış ve iç indeksler olarak gruplandırılmaktadırlar. Dış indeksler (external indices), kümeleme geçerliliğini sınamak için veri setlerinde daha önceden bilinen küme etiketlerini kullanır. Şöyle ki \mathbf{X} veri setindeki her bir x_k 'nin ait olduğu küme veya sınıf etiketi mevcut ise çalışılan algoritmanın önerdiği küme etiketi ile karşılaştırılır. Bu işlem veri setindeki tüm veri noktaları için

yapıldığında doğru ve yanlış kümelerde bulunan veri noktaları sayıları kullanılarak küme geçerliliğini ortaya koyan ölçütler elde edilebilir. Rand indeksi, Kulczynski indeksi, McNemar indeksi, Rogers-Tanimoto indeksi, Russel-Rao indeksi ve Phi indeksi gibi çok sayıda dış indeks geliştirilmiştir. Bu indekslerden çoğunu kapsayan yazılım paketleri de geliştirilmiştir. Örneğin R ortamındaki clusterCrit paketi bu indeksler yanında iç indekslerden bazılarını da içeren bir pakettir (Desgraupes, 2016).

Dış indekslerin dayandığı mantık yerinde olmakla birlikte uygulamada veri noktalarının ait olduğu sınıf bilgileri çoğu kez mevcut değildir. Bu durumda, geçerlilik analizlerinin kümeleme sonucunda elde bilgilerden yani kümeleme içi değerlerden belirlenmesi gerekir. Bu şekilde çalışan indeksler iç indeksler (internal indices) olarak adlandırılmakta olup bu amaçla çok sayıda indeks geliştirilmiştir. Ancak, iç indekslerden çoğu hiyerarşik kümeleme ve bölümleyici sert kümeleme analizlerinin geçerliliğini araştırmak üzere geliştirilmiştir. Az sayıda olsa da bulanık kümeleme sonucunda elde edilen üyelik derecelerini değerlendirmeye özel bir dizi bulanık kümeleme geçerlilik indeksi de bulunmaktadır (Halkidi ve ark., 2001; Xu, 2005; Wang ve Zhanga, 2007)

Kümelemede, birbirinden iyice ayrılmış; ancak kendi içindeki veri noktaları birbirine oldukça yaklaşmış kümeler elde edilmesi amaçlanmaktadır. Bunlardan ilkinde ayrışma (separation), ikincisine ise kompaktlık (compactness) denilmektedir. Kompaktlık en basit şekilde küme içindeki elemanların küme merkezinden uzaklığına dayanan küme içi varyans (within-group variance) şeklinde hesaplanır. Grup-içi varyansın küçük olması küme içindeki elemanların birbirine yakın olduğunu dolayısıyla kümenin kompakt olduğunu ifade eder. Ayrışma, iki küme arasında uzaklığı ifade eder ve bu kümeleri temsil eden elemanlar arasındaki uzaklıktan hesaplanır. Bu ölçüt hesaplama etkinliği ve aşırı yuvarlak kümeler için etkili olması nedeniyle yaygın olarak kullanılmaktadır. İç geçerlilik indekslerin çoğu, kompaktlık ve ayrışma ölçütlerinin farklı ve bazı durumlar için özelleştirilmiş uyarlamalarına dayanmaktadır.

3.3.1. Bulanık Kümeleme Geçerlilik İndeksleri

Bulanık kümelemede kullanılan geçerlilik indeksleri bulanık kümeleme sonucunda elde edilen sonuçları kullanmaktadırlar. Bulanık kümeleme geçerlilik indeksleri genel olarak 3 ayrı grupta incelenmektedir. İlk grupta yer alan indeksler yalnızca üyelik derecelerini kullanmakta ve iyi bir bulanık kümelemenin sert kümeleme sonucuna daha yakın sonuçlar vermesi gerektiği varsayımında bulunmaktadırlar. İkinci grubu oluşturan indeksler yalnızca üyelik derecelerini değil aynı zamanda veri setini de kullanarak verilerin geometrik özelliklerini de dikkate almaktadırlar. Üçüncü grupta yer alan indeksler daha çok kümeleme algoritmalarının durağanlığı (stabilitesi) üzerinde duran indekslerdir.

3.3.1.1. Üyelik Derecesi Kullanan İndeksler Bölümleme Katsayısı

Bulanık kümeleme analizinde ilk kullanılan geçerlilik indeksi olarak Bölümleme Katsayısı (PC: Partition Coefficient) bulanık kümeleme algoritmasını da geliştiren Bezdek tarafından önerilmiştir (Bezdek, 1974b). Aşağıdaki gibi formüle edilen PC yalnızca bulanık üyelik değerlerinden hesaplandığı için basit ve hesaplama maliyeti düşük bir indekstir. Ayrıca yuvarlak ya da küresel şekilli kümeler içeren veri setlerinde etkinliğinin aşağıda açıklanan PE ve XB indeksleriyle aynı olduğu; hatta küme sayısının gerçekten daha büyük seçilmesi durumunda XB'den daha iyi olduğu bildirilmektedir (Cebeci ve Yıldız, 2015).

$$V_{PC}(\mathbf{U}) = \frac{1}{n} \left(\sum_{i=1}^c \sum_{k=1}^n u_{ik}^m \right) \quad (3.26)$$

Eşitlik (3.26)'daki formül V_{PC} değerlerinin $[1/c, 1]$ aralığında olmasını sağlamaktadır. Değerler 1'e yaklaşırken sert kümeler elde edilirken alt sınıra yakın indeks değerleri kümelerin bulanıklaştığını göstermektedir. Eğer $1/c$ gibi bir değer söz konusu ise kümenin tüm üyelerinin üyelik dereceleri birbirine eşit ($u_{ik} = 1/c$) olur ki veri setinde herhangi bir kümeleme eğilimi bulunmadığını ya da kullanılan

algoritmanın başarısız olduğunu ifade eder (Halkidi ve ark., 2001). Kümelemede amaç birbirinden kesin olarak ayrılmış kümelemeler yapmak olduğundan $\max(V_{PC})$ değeri en iyi kümelemeyi verir.

Değiştirilmiş Bölümleme Katsayısı: PC indeksi yukarıda da açıklandığı üzere c 'ye bağlı olarak monotonik azlama eğilimi gösterdiğinden bunu düzeltmek üzere monotonik eğilimi gideren Değiştirilmiş Bölümleme Katsayısı (MPC; Modified Partition Coefficient) kullanılması önerilmiştir (Dave, 1996).

$$V_{MPC}(\mathbf{U}) = 1 - \frac{1}{c-1} (1 - V_{PC}(\mathbf{U})) \quad (3.27)$$

V_{MPC} indeks değerleri $[0,1]$ aralığında elde edilir ve $\max(V_{MPC})$ değeri en iyi kümelemeyi veren indeks değeri olarak değerlendirilir. MPC indeksi c 'ye bağımlı olarak oluşan monotonik eğilimi düşürmekle birlikte sadece bulanık üyelik değerlerini kullanmakta ve veri setindeki kümelerin geometrik yapısı ile ilgilenmemektedir.

Bölümleme Entropisi: Bezdek (1974a) tarafından önerilen diğer bir bulanık geçerlilik indeksi olan Bölümleme Entropisi (PE; Partition Entropy) (3.28)'deki gibi formülize edilmiştir.

$$V_{PE}(\mathbf{U}) = \frac{1}{n} (\sum_{i=1}^c \sum_{k=1}^n u_{ik} \log_a(u_{ik})) \quad (3.28)$$

Burada a , bir logaritma tabanını göstermektedir. V_{PE} değerleri $[0, \log_a c]$ aralığında elde edilir. PC indeks değerlerinin aksine PE değerleri küçük olduğunda yani 0 'a doğru yaklaştığında birbirinden iyi ayrılmış kümeler elde edilirken üst sınıra doğru veri yaklaşırken küme yapıları bulanık hale gelmeye başlar. $\log_a c$ 'ye eşit bir V_{PE} indeks değeri veri yapısında kümelenme eğilimi olmadığını ya da kullanılan algoritmanın tam başarısız olduğunu gösterir. Bu nedenle, $\min(V_{PE})$ değeri en iyi kümelemeyi veren indeks değeridir.

PC ve PE indeksleri küme sayısına (c) monotonik bağımlılık göstermektedirler. Küme sayısına göre PC grafiğinde bir tepe noktası veya küme sayısına göre PE grafiğinde bir çukur noktası aranır. Her iki indeks de m bulanıklaştırma parametresine karşı hassastır. Öyle ki as $m \rightarrow 1$ giderken indisler tüm c 'ler için aynı değerleri verirler. Ancak $m \rightarrow \infty$ giderken her iki indeks $c = 2$ 'de çok belirgin bir tepe/çukur gösterirler. Her iki indeksin bir başka dezavantajı veri setinde küme yapıları ve şekillerine ait bilgileri kullanmaması yani veri setinin (\mathbf{X}) kendisinden yararlanmamasıdır.

3.3.1.2. Veri Seti, Üyelik Dereceleri Ve Küme Merkezlerini Kullanan İndeksler

Xie-Beni İndeksi: Xie ve Beni (1991), payda küme kompaktlığını ve paydada küme ayrışmasını dikkate alan Xie-Beni İndeksini (XB; Xie - Beni Index) geliştirmişlerdir.

$$V_{XB}(\mathbf{X}, \mathbf{U}, \mathbf{V}) = \frac{\sum_{i=1}^c \sum_{k=1}^n u_{ik}^m \|x_k - v_i\|^2}{n(\min_{i \neq j} \{\|v_j - v_i\|^2\})} \quad (3.29)$$

Eşitlik (29)'da pay terimi bir kümedeki birimlerin kendi küme merkezlerinden uzaklıklarını dikkate alarak bulanık kümelemenin kompaktlığını ölçerken payda terimi küme merkezleri arasındaki uzaklıklar ile kümelerin ayrışma gücünü dikkate almaktadır. Bu nedenle küçük XB indeksi değerleri kompakt ve iyi ayrılmış kümeler elde edildiğini göstermektedir. Ancak c , n 'ye doğru yaklaştıkça XB monotonik şekilde azalış gösterebilmektedir. XB indeks değerlerindeki bu azalma eğilimini elimine etmek için monotonik davranışın başlangıç noktası olarak c_{max} saptanıp XB'nin minimum değerleri $\min(V_{XB})$, $[2, c_{max}]$ aralığında araştırılır. XB indeksinin bir diğer dezavantajı ise $m \rightarrow \infty$ giderken $V_{XB} \rightarrow \infty$ de sonsuza gitmesidir.

Kwon İndeksi: Kwon (1998), Xie-Beni İndeksi'nde küme sayısının artışıyla ortaya çıkan monotonik azalma eğilimini elimine eden bir geçerlilik fonksiyonu önermiştir. Bunun için XB indeksinin payına (3.30)'da gösterilen bir cezalandırma terimi eklenmiştir.

$$V_{Kwon}(\mathbf{X}, \mathbf{U}, \mathbf{V}) = \frac{\sum_{i=1}^c \sum_{k=1}^n u_{ik}^m \|x_j - v_i\|^2 + \frac{1}{c} \sum_{i=1}^c \|v_i - \bar{v}\|^2}{\min_{i \neq j} \{\|v_i - v_j\|^2\}} \quad (3.30)$$

Burada:

$$\bar{v} = \frac{1}{n} \sum_{k=1}^n x_k$$

Olup Kwon indeksi ile en iyi kümeleme $\min(V_{Kwon})$ ile araştırılır.

Tang-Sun İndeksi: Tang ve Sun (2005), Kwon İndeksi'ni Kwon indeksine bir cezalandırma terimi ekleyerek iyileştirmişlerdir.

$$V_{TS}(\mathbf{U}, \mathbf{V}, \mathbf{X}) = \frac{\sum_{i=1}^c \sum_{k=1}^n u_{ik}^m \|x_k - v_i\|^2 + \frac{1}{c(c-1)} \sum_{i=1}^c \|v_i - \bar{v}\|^2}{\min_{i \neq j} \{\|v_i - v_j\|^2\} + 1/c} \quad (3.31)$$

Eşitlik (3.31)'de pay ve paydadaki ikinci terim bir *ad hoc* cezalandırma (küme merkezleri arasındaki ortalama uzaklık) olup $c \rightarrow n$ giderken azalma eğilimini ortadan kaldırmak için uygulanmıştır.

Bulanık Siluet İndeksi

$$V_{FSil}(\mathbf{U}, \mathbf{V}, \mathbf{X}) = \frac{\sum_{k=1}^n (u_{ik} - u_{ik'})^\alpha s_i(k)}{\sum_{k=1}^n (u_{ik} - u_{ik'})^\alpha} \quad (3.32)$$

Burada:

$$s_i(k) = \frac{b_i - a_i}{\max(b_i, a_i)} \quad (3.33)$$

olup i . veri birimi için siluet indeksidir.

a_i bir veri noktası ile aynı kümeye ait tüm veri birimleri arasındaki ortalama benzeşmezliktir. b_i bir veri noktasının üye olmadığı diğer kümelerin herhangi biri için en küçük ortalama benzeşmezliktir. u_{ik} ve u_{ik}' sırasıyla üyelik matrisi U 'nın i . sırasındaki ilk ve ikinci en büyük elemanlardır. α tartı katsayısı olup genelde 1'e eşit olarak kullanılır.

3.3.1.3. Diğer Bulanık Geçerlilik İndeksleri

Yu ve Li (2006), BCO için durağanlık indeksi (stability index for FCM) olarak adlandırılan ve optimallik testine dayanan bir indeks geliştirmişlerdir. Bu tür indekslerin diğerlerine göre avantajları bulunmakla birlikte hesaplama maliyetinin yüksekliği nedeniyle bu tez çalışmasında kümeleme algoritmalarının durağanlığı üzerine odaklanan algoritmalar üzerinde çalışılmamıştır.

3.3.2. Genelleştirilmiş Ve Uzatılmış Geçerlilik İndeksleri

Yukarıdaki bölümlerde açıklanan indeksler BCO algoritmasından elde edilen üyelik dereceleri matrisini (U) kullanmaktadırlar. Buna karşın OCO algoritması üyelik dereceleri yerine tipiklik dereceleri matrisini (T) üretmektedir. Öte yandan OBCO, BOCO, iOBCO ve uOBCO gibi algoritmalar ise hem üyelik dereceleri hem de tipiklik dereceleri matrisini üretmektedirler. Bu durumda bulanık geçerlilik indeksleri kullanışsız hale gelmekte ya da yetersiz kalmaktadırlar.

Bu soruna çözüm bulmak üzere yapılan çalışmalarda genelde iki tür yaklaşım yapıldığı görülmektedir. Bu yaklaşımlardan ilki T matrisini U için verilen kısıtları sağlayacak şekilde dönüştürmek, ikincisi ise U ve T matrislerini kombine ederek (birleştirerek) mevcut bulanıklık indeksleri ile kullanmak fikrine dayanmaktadırlar.

3.3.2.1. Genelleştirilmiş Bulanık İndeksler

Yang ve Wu (2006), bulanık üyelik derecelerinin kullanan yukarıdaki indekslerin olabilirlikli algoritmalarından hesaplanan \mathbf{T} değerleri için çalışmayacağını; örneğin c artarken PC ve PE değerlerinin de artacağını, diğer indeksler için de arzulanmayan sonuçlara yol açacağını ifade etmiştir. Araştırmacılar bu sorunu çözmek için mevcut geçerlilik indekslerini genelleştirilmiş olarak kullanmak üzere bir normalleştirme tekniği önermişlerdir. Daha önce açıklandığı üzere, olabilirlikli kümelemede (3.7)'deki $\sum_{i=1}^c u_{ik} = 1$ kısıtı gevşetilmektedir. Bu ise bulanık geçerlilik indekslerinin etkinliğini kaybetmesine yol açmaktadır. Ancak, \mathbf{T} değerleri $[0,1]$ aralığında olacak şekilde normalleştirildiğinde yukarıdaki kısıt sağlandığından bulanık indeksler tipiklik dereceleri için de uygulanabilir duruma gelmektedir.

$$u'_{ik} = \frac{t_{ik}}{\sum_{i=1}^c t_{ik}} \quad (3.34)$$

Eşitlik (34)'deki gibi normalleştirilmiş tipiklik dereceleri (u'_{ik})'nin aslında her bir tipiklik değerinin bulunduğu satırın toplamına bölünmesi ile hesaplandığı görülmektedir. Böylece bulanık geçerlilik indekslerinde üyelik dereceleri (u_{ik}) yerine konularak indeks değerleri hesaplanabilmekte ve böylece bulanık indeksler olabilirlikli algoritmalarla kullanılabilir hale gelmektedir. Yang ve Wu (2006) bu normalleştirilmiş değerleri kullanan indeksleri *genelleştirilmiş indeksler* (*generalized indexes*) olarak adlandırmışlardır. Anlatılan teknik (3.35)'de sadece PE için gösterilmiş olup diğer indekslerde de benzer şekilde uygulanmaktadır.

$$V_{GPE}(\mathbf{U}') = \frac{1}{n} (\sum_{i=1}^c \sum_{k=1}^n u'_{ik} \log_a(u'_{ik})) \quad (3.35)$$

3.3.2.2. Uzatılmış Bulanık İndeksler

Uzatılmış Bulanık İndeksler (Extended Fuzzy Indices), üyelik ve tipiklik derecelerinin kombine edilmesine dayanmaktadır. Aslında bunlar özgün birer indeks değil, ancak **T** ve **U**'nun kombine edilmiş değerlerinin daha önce açıklanan bulanık indekslerle kullanılmasına dayanmaktadır. Eşitlik (3.36), (3.37) ve (3.38)'de XB, Kwon ve Tang-Sun indeksleri için uzatılmış indeks değeri hesaplama formülleri gösterilmektedir. Bu formüllerin tümünde kombine değerlerin toplam şeklinde olduğu yani **U+T** şeklinde uygunlandığı görülmektedir.

Uzatılmış Xie-Beni İndeksi

$$EV_{XB}(\mathbf{X}, \mathbf{U}, \mathbf{T}, \mathbf{V}) = \frac{\sum_{i=1}^c \sum_{k=1}^n (u_{ik}^m + t_{ik}^n) \|x_k - v_i\|^2}{n(\min_{i \neq j} \{\|v_j - v_i\|^2\})} \quad (3.36)$$

Uzatılmış Kwon İndeksi

$$EV_{Kwon}(\mathbf{X}, \mathbf{U}, \mathbf{T}, \mathbf{V}) = \frac{\sum_{i=1}^c \sum_{k=1}^n (u_{ik}^m + t_{ik}^n) \|x_k - v_i\|^2 + \frac{1}{c} \sum_{i=1}^c \|v_i - \bar{v}\|^2}{\min_{i \neq j} \{\|v_i - v_j\|^2\}} \quad (3.37)$$

Uzatılmış Tang-Sun İndeksi

$$EV_{TS}(\mathbf{U}, \mathbf{V}, \mathbf{X}) = \frac{\sum_{i=1}^c \sum_{k=1}^n (u_{ik}^m + t_{ik}^n) \|x_k - v_i\|^2 + \frac{1}{c(c-1)} \sum_{i=1}^c \|v_i - \bar{v}\|^2}{\min_{i \neq j} \{\|v_i - v_j\|^2\} + 1/c} \quad (3.38)$$

Uzatılmış indeks geliştirme ile ilgili çalışmalar çok azdır. Bu çalışmalardan birinde Zang ve ark. (2008) PE için Uzatılmış Bölümleme Entropisi (EPE) ve kümeler arası benzerlik derecesinin farklı ağırlıklarda kombine edilmesine dayanan EPESIM adında yeni bir indeks önermişlerdir.

Uzatılmış Bölümleme Entropisi: Zang ve ark. (2008) tarafından önerilen bu indeks, (3.39) ve (3.40)'daki formüllerle **U** ve **T** değerleri kullanılarak

hesaplanan ara indekslerin belli ağırlıklarla (3.41)'deki gibi birleştirilmesine dayanmaktadır.

$$V_{PEU}(\mathbf{U}) = \frac{1}{n \log_a(c)} (\sum_{i=1}^c \sum_{k=1}^n u_{ik} \log_a(u_{ik})) \quad (3.39)$$

$$V_{PET}(\mathbf{T}) = \frac{1}{c \log_a(n)} (\sum_{i=1}^c \sum_{k=1}^n t_{ik} \log_a(t_{ik})) \quad (3.40)$$

$$V_{EPE}(\mathbf{U}, \mathbf{T}) = \alpha_1 V_{PEU}(\mathbf{U}) + \beta_1 V_{PET}(\mathbf{T}) \quad (3.41)$$

Burada, α_1 ve β_1 üyelik ve tipiklik derecelerine verilecek ağırlık yüzdesini gösteren katsayılar olup $\alpha_1 + \beta_1 = 1$ koşulu sağlanmalıdır. Zhang ve ark. (2008) eşitlik ağırlıkla çalışmış ve $\alpha_1 = 0.5$ ve $\beta_1 = 0.5$ olarak seçmişlerdir.

Kümeler-arası Benzerlik Derecesi: Kümeleme algoritmasının sonucunda elde edilen her bir bulanık küme, bir bulanık set ve veri matrisinin tamamı (\mathbf{X}) ise bunların ait olduğu evrendir. Dolayısıyla kümelerin ayrışması bu bulanık setler arasındaki benzerlikle modellenebilir (Zang ve ark., 2008). Az benzerlik çok ayrışma anlamına gelir ki, kümelemenin iyi çalıştığını ifade eder. \mathbf{A} ve \mathbf{B} gibi iki küme arasındaki benzerlik:

$$S(A, B) = \frac{\|A \cap B\|}{\|A \cup B\|} \quad (3.42)$$

ile hesaplanır. Burada $\|.\|$, \mathbf{A} ve \mathbf{B} bulanık setlerinin kardinalitesidir. Bulanık setlerin kardinalitesini hesaplamak için çeşitli yöntemler bulunmakla birlikte Eşitlik (3.39)'daki sigma sayılarının kullanılması klasik yollar yöntemlerden biridir. Birleşik bir benzerlik değeri üretmek için üyelik ve tipiklik dereceleri için ayrı ayrı benzerlikler benzerlikler hesaplanır Eşitlik (3.42)'deki gibi belli ağırlıklar kullanılarak toplanır.

Üyelik derecelerine göre ortalama kümeler-arası benzerlik matrisi:

$$\begin{aligned}\bar{S}(\mathbf{U}) &= \frac{1}{c(c-1)/2} \sum_{i \neq j} S(U(i), U(j)) \\ &= \frac{1}{c(c-1)/2} \sum_{i \neq j} \frac{\sum_{k=1}^n \min(u_{ik}, u_{jk})}{\sum_{k=1}^n \max(u_{ik}, u_{jk})}\end{aligned}\quad (3.43)$$

Burada $U(i)$ ve $U(j)$ üyelik matrisinin i . ve j . sıralarını gösteren vektörlerdir. $\bar{S}(\mathbf{U})$ benzerlik ölçüsü olduğundan $[0,1]$ aralığında bir değer olarak hesaplanmaktadır. Küçük benzerlik değerleri kümelerin birbirinden daha çok ayrık olduğunu ve dolayısıyla bölümlene algoritmasının iyi çalıştığını gösterir.

Benzer şekilde tipiklik derecelerine göre ortalama kümeler-arası benzerlik matrisi:

$$\begin{aligned}\bar{S}(\mathbf{T}) &= \frac{1}{c(c-1)/2} \sum_{i \neq j} S(T(i), T(j)) \\ &= \frac{1}{c(c-1)/2} \sum_{i \neq j} \frac{\sum_{k=1}^n \min(t_{ik}, t_{jk})}{\sum_{k=1}^n \max(t_{ik}, t_{jk})}\end{aligned}\quad (3.44)$$

olarak hesaplanıp α_2 ve β_2 tartı sabitleri kullanılarak birleştirilip U ve T'ye bağlı olarak ortalama kümeler-arası benzerlik indeksi bulunur.

$$V_{ICSIM}(\mathbf{U}, \mathbf{T}) = \alpha_2 \bar{S}(\mathbf{U}) + \beta_2 \bar{S}(\mathbf{T}) \quad (3.45)$$

Burada, α_2 ve β_2 üyelik ve tipikliğe verilecek ağırlık yüzdesini gösteren katsayılar olup $\alpha_2 + \beta_2 = 1$ koşulu sağlanmalıdır. Buna göre eşitlik ağırlık durumunda $\alpha_2 = 0.5$ ve $\beta_2 = 0.5$ olarak seçilmelidir.

Epesim İndeksi: Zang ve ark. (2008), Eşitlik (3.38)'deki V_{EPE} ve Eşitlik (42)'deki V_{ICSIM} değerlerini sırasıyla α_3 ve β_3 oranlarında kombine eden ve EPESIM adını verdikleri uzatılmış bir kümeleme geçerlilik indeksi önermişlerdir.

$$V_{EPESIM}(\mathbf{U}, \mathbf{T}) = \alpha_3 V_{EPE}(\mathbf{U}, \mathbf{T}) + \beta_3 V_{ICSIM}(\mathbf{U}, \mathbf{T}) \quad (3.46)$$

Burada $\alpha_3 + \beta_3 = 1$ olup EPE ve ICSIM indekslerine verilen ağırlıkları ifade etmektedir. Araştırmacılar tarafından yapılan 2 yapay veri seti ve Iris veri seti üzerinde yapılan testlere göre indeksin diğer uzatılmış indekslere göre daha az işlem zamanı gerektirmesi nedeniyle avantajlı olduğu bildirilmiştir.

3.4 Veri Setlerinin Oluşturulması Ve Analizler

3.4.1 Veri Setleri Ve Kümeleme Sonuçlarının İncelenmesi

Araştırmada karşılaştırılacak algoritmalar için R ortamında geliştirilmiş olan çeşitli özel paketler kullanılmıştır. Karşılaştırma sonucunda elde edilen sonuçların niteleyici ve niceleyici değerlendirmeleri için R ortamında çalışacak bir R programı geliştirilmiştir.

3.4.2 Algoritma Parametrelerinin Seçilmesi

Küme sayısı (c) parametresi: Kümeleme analizinde kullanılan algoritmalar için yapılan testlerde minimum küme sayısı 2 ve maksimum küme sayısı ise veri büyüklüğünü dikkate alan (3.47)'deki formül kullanılmıştır.

$$c_{maz} = \log_e(n) + 2 \quad (3.47)$$

Eşitlik (3.47)'deki veri büyüklüğü (n) çok küçük olduğunda küme sayısını en az 2 yapmak üzere tasarlanmıştır. Bu basit formül çoğu veri setleri için pratik olarak kullanılabilir durumda olup bu tez çalışmasında maksimum veri büyüklüğü 820 olduğundan maksimum küme sayısı $c_{max} = 9$ olarak hesaplanmıştır.

Bulanıklaştırma katsayısı: Bölümleyici algoritmalarla ilgili çalışmaların çoğunda bulanıklaştırma katsayısı (fuzzyfier) 2 olarak belirlendiğinden bu çalışmada m parametresi 2 olarak alınmıştır ($m=2$).

Tipikleştirme katsayısı: Tipikleştirme katsayısı ya da tipikleştirici (η) olarak konu üzerinde yapılan diğer çalışmalarda olduğu gibi $\eta = 2$ seçilmiştir.

Eta (η) vektörü: Olabilirlikli kümelemede kullanılan *eta* vektörünün elemanları (n_i), (3.10)'da verilen formülle analiz esnasında her bir veri seti için dinamik olarak hesaplanmış ve kullanılmıştır.

Yaklaşma (ϵ) değeri: Algoritmalarda mevcut iterasyonla bir önceki iterasyon arasındaki farklılığın ölçüsü olarak yaklaşma değeri (convergence value) $1e-03$ ile $1e-09$ arasında seçilmektedir. Büyük değerler seçildiğinde iterasyonlar arası yaklaşma daha az sayıda iterasyonla sağlanabileceğinden hassas sonuçlar elde edilemeyebilir. Buna karşılık çok küçük değerlerin seçilmesi halinde iterasyon sayısı artacağından hesaplama maliyeti yükselecektir. Deneysel çalışmaların birçoğunda bu değer $1e-05$ veya $1e-06$ olarak alındığından bu çalışmada $\epsilon = 1e - 05$ yaklaşma değeri kullanılmıştır.

Maksimum iterasyon sayısı: Bazı durumlarda yaklaşma değerine ulaşmak çok uzun sürebilmekte ve hesaplama zamanı çok uzayabilmektedir. Böylesi durumların önüne geçmek için kümeleme algoritmalarında maksimum iterasyon sayısı sınırlaması yapılabilir. Bu çalışmada maksimum iterasyon sayısı 5000 (maxiter=5000) olarak tanımlanmıştır.

a ve b ağırlık katsayıları: OBCO algoritmasında kullanılan *a* değeri bulanık üyelik derecelerine; *b* ise tipiklik derecelerine verilen ağırlığı ifade etmektedirler. Göreli olarak *a* değeri büyük tutulduğunda sonuçlar BCO algoritması sonuçlarına yaklaşmakta, aksine *b* değeri büyük tutulduğunda ise OCO algoritmasına yakınlaşmaktadır. Bu tez çalışmasında üyelik ve tipiklik eşit ağırlıkla ele alınmış ve $a=b=1$ olarak belirlenmiştir.

γ değeri: İtmeli olabilirlikli C-ortalamalar algoritmasında kullanılan itme teriminin tartı (ağırlık) faktörü olan gamma (γ) değeri pozitif bir reel sayı olarak belirlenmektedir. Bu faktörün seçimi veri yapısına bağlı olmakla birlikte (Vanisri

ve Loganathan, 2011) tarafından yapılan çalışmada 15-20 arasında bir değerin kullanılması önerildiğinden bu çalışmada $\gamma = 20$ olarak kullanılmıştır.

β değeri: Yönetimsiz Olabilirlikli Bulanık C-ortalamalar (uOBCO) algoritmasında kullanılan değeri β (3.24.3)'deki formülle analiz sırasında kullanılan veri setlerinden dinamik olarak hesaplanmıştır.

Uzatılmış indeks ağırlık katsayıları: Uzatılmış indeks değerlerinin hesaplanmasında kullanılan bulanık üyelik dereceleri (**U**) ve tipiklik dereceleri (**T**) için eşit ağırlıklarla çalışılmış ve $\alpha_1 = 0.5, \beta_1 = 0.5; \alpha_2 = 0.5, \beta_2 = 0.5$ ve $\alpha_3 = 0.5, \beta_3 = 0.5$ olarak kullanılmıştır.

3.4.3 Kümeleme Başarımlarının Karşılaştırılması

Çalışmada her bir algoritma, minimum iki küme sayısından ($c=2$) başlamak üzere veri setindeki nesne sayısında bağlı olarak (3.42)'de belirlenen maksimum küme sayısına ($c=c_{max}$) kadar farklı küme sayıları için çalıştırılmıştır. Farklı algoritmalarda üyelik dereceleri ve küme prototiplerinin başlangıç matrislerinin üretilmesi sırasında aynı matrislere ulaşmak için R'de şans sayısı üretici için tohum (seed) değeri olarak 3 atanmıştır (`set.seed(3)`). Her bir küme sayısı için yapılan analizlerde üyelik dereceleri (**U**) ve tipiklik dereceleri (**T**) matrisleri hesaplanmıştır.

Algoritmaların kümeleme başarımlarının karşılaştırılmasında yaygın olarak kullanılan ve Bölüm 3.3'de tanıtılan bulanık iç indeksler (internal fuzzy indices) kullanılmıştır. Olabilirlikli kümeleme algoritmalarının kümeleme başarımlarında ise uzatılmış bulanık iç indeksler (extended internal fuzzy indices) kullanılmıştır. Tüm algoritmalarda her bir küme sayısı için bu indeks değerleri için karşılaştırma tabloları üretilmiş ve ayrıca grafikler oluşturulmuştur.

Algoritma başarımlarının karşılaştırılmasında ihtiyaç duyulan iterasyon sayıları ve hesaplama süreleri de hesaplanmış ve grafikleri çizilmiştir. Bunların

dışında her bir algoritma kendi içinde farklı küme sayıları ve amaç fonksiyon değerleri bakımından da karşılaştırılmıştır.

4. ARAŞTIRMA BULGULARI VE TARTIŞMA

4.1. BCO Algoritması

4.1.1. Sentetik Veri Setleri

4.1.1.1. Sentetik Veri Seti 1 (SDSC)

Çizelge 4.1’de daire şekilli 4 adet küme içeren ilk sentetik veri seti (SDSC) için BCO algoritmasından hesaplanan indeks değerleri görülmektedir. Elde edilen sonuçlara göre V_{XB} , V_{Kwon} , V_{TS} , V_{MPC} ve V_{Fsil} indeksleri veri setinde bulunan küme sayısını ($c=4$) isabetli şekilde saptamıştır. Küme şekillerini dikkate almayan V_{PE} indeksi ise küme sayısını 2 ($c=2$) olarak tahmin etmiştir. Buna karşın Şekil 4.1’deki PE grafiği incelendiğinde bu değerden sonraki ilk bükümün 4 küme ($c=4$) için olduğu görülmektedir. Bu sonuçlara göre genel bir değerlendirme yapılırsa BCO algoritmasının daire şekilli kümeleri saptamada başarılı olduğu görülmektedir.

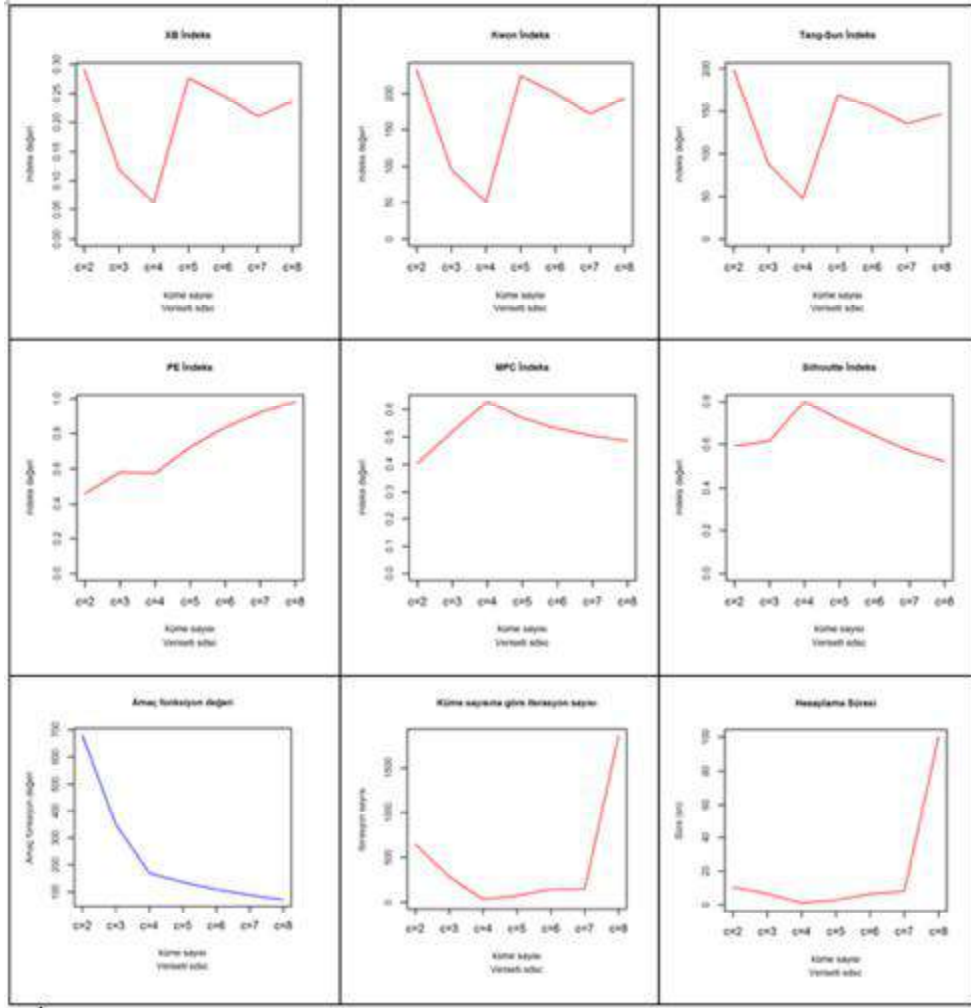
Çizelge 4.1. SDSC veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
$c=2$	0.291	232.704	198.973	0.462	0.405	0.592
$c=3$	0.119	95.343	87.588	0.583	0.518	0.615
$c=4$	0.063	51.129	47.739	0.574	0.629	0.799
$c=5$	0.276	224.004	169.549	0.723	0.567	0.716
$c=6$	0.247	200.909	155.112	0.833	0.530	0.640
$c=7$	0.210	171.440	135.324	0.918	0.504	0.571
$c=8$	0.236	193.432	146.215	0.983	0.486	0.522

Çizelge 4.2. SDSC veri seti için BCO algoritması başarımı

C	İterasyon sayısı	Hesaplama süresi (s)	Fonksiyon değeri
$c=2$	647	10.480	680.568
$c=3$	279	6.560	349.577
$c=4$	32	1.000	169.345
$c=5$	75	2.850	135.969
$c=6$	141	6.500	110.149
$c=7$	150	8.100	88.522
$c=8$	1855	100.480	71.911

BCO algoritmasının SDSC veri seti üzerinde hesaplama süresi ve iterasyon başarımları Çizelge 4.2’de gösterilmiştir. Çizelge 4.2 ve Şekil 4.1 incelendiğinde hesaplama süresinin veri setindeki küme sayısına yaklaşırken düştüğü; küme sayısından uzaklaşırken artmaya başladığı anlaşılmaktadır.



Şekil 4.1. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımları (SDSC)

Şekil 4.1'e göre hesaplama süresi ve iterasyon sayısı arasında güçlü bir ilişki olduğu görülmektedir. Her iki başarımlı ölçüsü veri setindeki gerçek küme sayısında ($c=4$) minimuma ulaşmakta ve bu noktadan sonra yükselmeye başlamaktadır. Bu sonuç, hesaplama süresi ve iterasyon sayısındaki en alçak büküm noktasının veri setindeki küme sayısının bir göstergesi olarak kullanılabilceğini ifade etmektedir. Cebeci ve Yıldız (2015)'in BCO algoritması için yapmış oldukları çalışmada da benzer sonuçlar alındığı bildirilmiştir. Şekil 4.1 ve Çizelge 4.2'de amaç fonksiyon değerleri incelendiğinde amaç fonksiyon değerinin küme sayısının artışıyla kuadratik bir azalma eğilimi gösterdiği ve gerçek küme sayısında belirgin olmayan bir bükülme noktasına sahip olduğu anlaşılmaktadır.

4.1.1.2. Sentetik Veri Seti 2 (SDSCN)

SDSCN veri seti daire şekilli 4 adet küme içeren sentetik veri seti olup Şekil 31a'dan görüleceği üzere alttaki iki kümenin üst kısmına gürültüler eklenmiştir. BCO algoritmasının gürültü ve aykırı değerlere karşın sorunlu olması nedeniyle kümeleme başarımında sorunlar olmaktadır. Çizelge 4.3'de BCO algoritmasından hesaplanan indeks değerlerine göre V_{XB} , V_{Kwon} , V_{TS} , V_{MPC} ve V_{Fsil} indeksleri SDSCN veri setinde bulunan küme sayısını ($c=4$) isabetli şekilde saptamıştır. V_{PE} indeksi ise küme sayısını gürültüsüz SDSC veri setinde olduğu gibi 2 ($c=2$) olarak önermektedir. Buna karşın Şekil 4.2'deki PE grafiği incelendiğinde bu değerden sonraki ilk bükümün 4 küme ($c=4$) için olduğu görülmektedir. Bu sonuçlara göre BCO algoritmasının daire şekilli ve gürültülü kümeleri saptamada başarılı olduğu görülmektedir. Ancak Ek 1.1 ve Ek 1.2'de SDSC ve SDSCN için kümelere üyelik dereceleri için çizilen grafik incelendiğinde küme sayısının başarılı olarak bulunduğu ($c=4$) küme grafiklerinin farklı olduğu anlaşılmaktadır. Buna göre, eğer daha fazla gürültülü ve karmaşık veri setleri ile çalışıldığında BCO'nun sorunlu olabileceği matematiksel ve deneysel olarak ortaya konulmuştur.

Çizelge 4.3. SDSCN veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.308	252.616	214.593	0.469	0.388	0.556
c=3	0.126	103.535	94.955	0.606	0.496	0.596
c=4	0.065	53.522	50.084	0.598	0.611	0.778
c=5	0.276	228.587	176.151	0.778	0.531	0.684
c=6	0.277	230.652	175.678	0.888	0.494	0.602
c=7	0.268	223.870	169.547	0.955	0.484	0.562
c=8	0.230	193.277	149.063	1.000	0.473	0.519

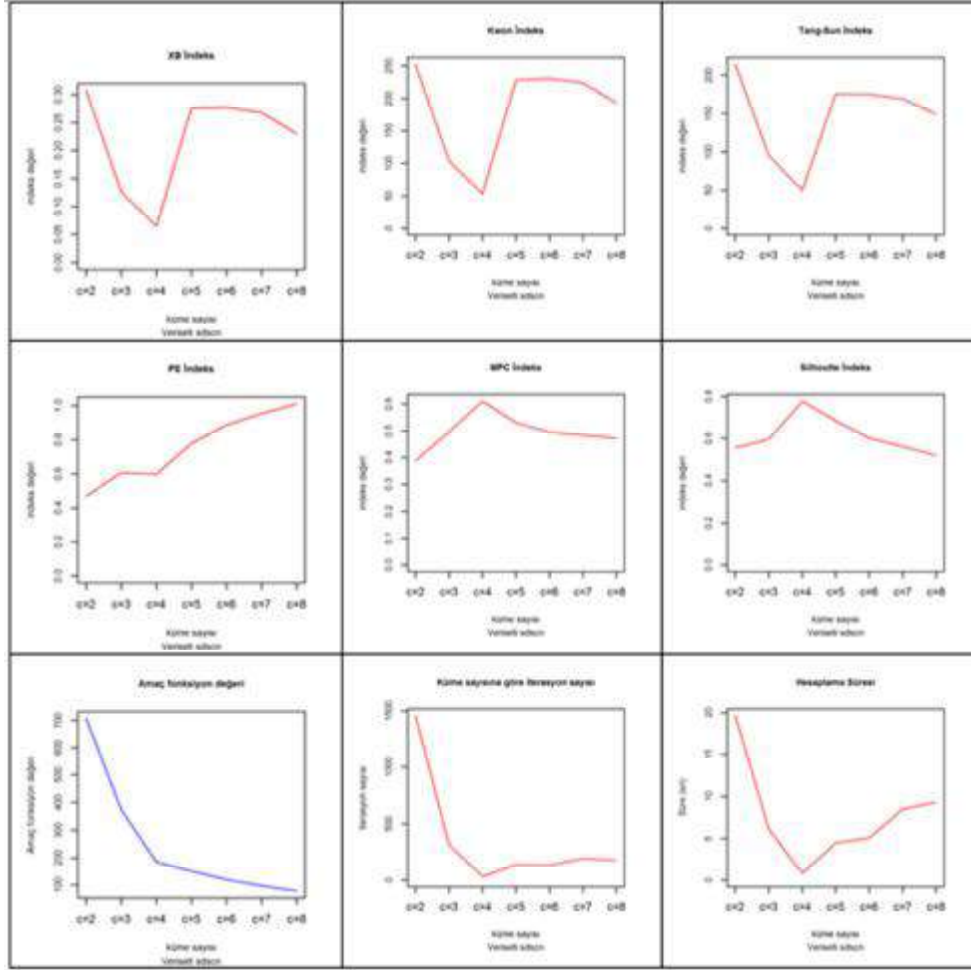
Çizelge 4.4. SDSCN veri seti için BCO algoritması başarımı

C	İterasyon sayısı	Hesaplama süresi (s)	Fonksiyon değeri
c=2	14593	19.670	707.492
c=3	314	6.150	373.049
c=4	32	0.810	184.042
c=5	136	4.460	150.099
c=6	132	5.050	119.620
c=7	193	8.470	96.781
c=8	180	9.260	78.416

Çizelge 4.4'de BCO algoritmasının SDSCN veri setindeki başarımları sonuçları görülmektedir. Hesaplama süresinin SDSC veri seti için yapılan değerlendirmeye benzer eğilimde olduğu; veri setindeki küme sayısına yaklaşılan küme sayısından uzaklaşırken artmaya başladığı anlaşılmaktadır. Şekil 4.3'e göre de hesaplama süresi ve iterasyon sayısı arasında pozitif ve yüksek bir ilişki olduğu görülmektedir. Her iki başarımların ölçüsü veri setindeki gerçek küme sayısında (c=4) minimuma ulaşmakta ve bu noktadan sonra yükselmeye başlamaktadır.

Şekil 4.2 ve Çizelge 4.2'de amaç fonksiyon değerleri incelendiğinde amaç fonksiyon değerinin küme sayısının artışıyla kuadratik bir azalma eğilimi gösterdiği ve gerçek küme sayısında belirgin olmayan bir bükülme noktasına sahip

olduğu görülmektedir. Elde edilen bu sonuç SDSC veri seti ile elde edilenlerle büyük benzerlik içindedir.



Şekil 4.2. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSCN)

4.1.1.3. Sentetik Veri Seti 3 (SDSE)

Eliş şekilli gürültüsüz kümeler içeren SDSE veri setinde BCO algoritması sonuçlarına göre hesaplanan indeks değerleri incelendiğinde (Çizelge 4.5 ve Şekil

4.3), V_{XB} , V_{Kwon} , V_{TS} , V_{MPC} ve V_{Fsil} indekslerine göre SDSCN veri setinde bulunan küme sayısı ($c=4$); V_{PE} indeksine göre ($c=2$) olarak bulunmuştur.

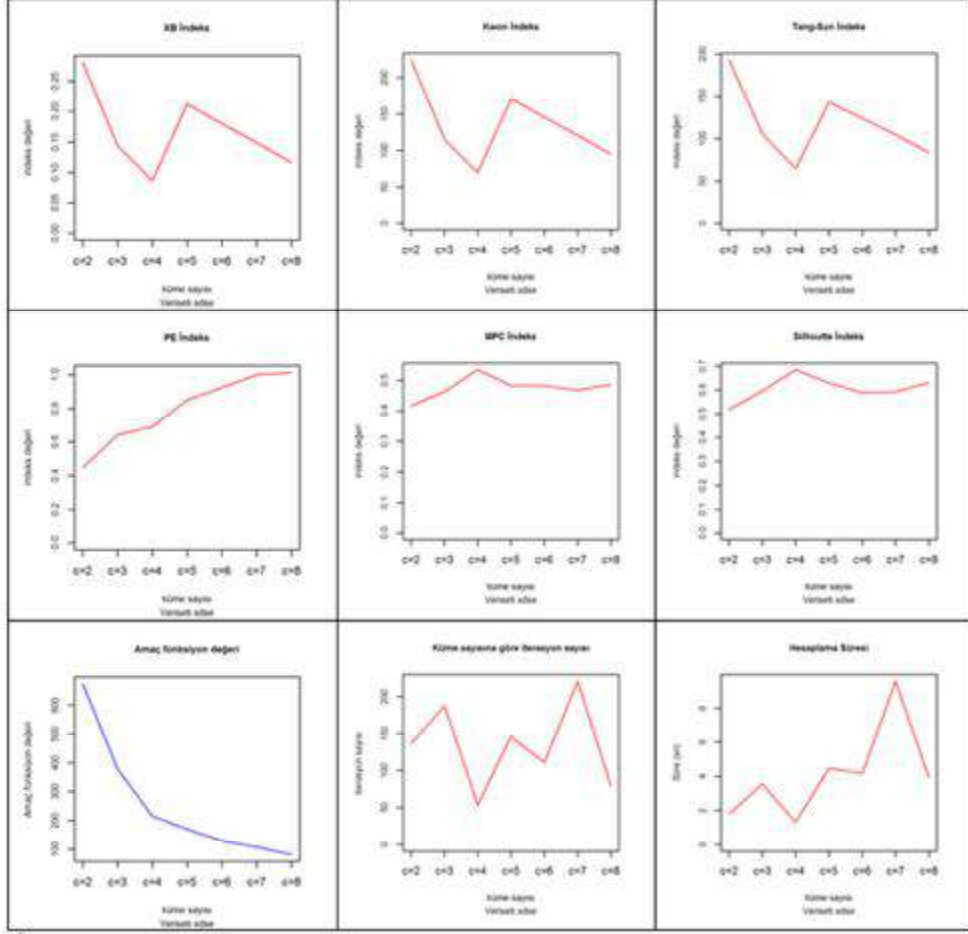
Çizelge 4.5. SDSE veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
$c=2$	0.281	224.818	192.931	0.448	0.416	0.519
$c=3$	0.144	115.797	105.288	0.641	0.464	0.597
$c=4$	0.086	69.706	64.702	0.692	0.535	0.685
$c=5$	0.212	171.274	143.191	0.852	0.482	0.629
$c=6$	0.180	145.732	123.408	0.923	0.481	0.589
$c=7$	0.148	120.813	104.613	1.000	0.468	0.592
$c=8$	0.116	95.105	83.663	1.000	0.486	0.631

Çizelge 4.6. Çizelge 4.6. SDSE veri seti için BCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	136	1.820	674.084
$c=3$	188	3.560	378.868
$c=4$	54	1.330	216.078
$c=5$	146	4.480	170.612
$c=6$	111	4.170	130.171
$c=7$	221	9.590	107.323
$c=8$	79	3.930	82.647

Çizelge 4.5 ve Şekil 4.3'den BCO algoritmasının SDSE veri setindeki başarımları sonuçları incelendiğinde daha önce SDSC ve SDSCN veri setindeki sonuçlarla benzerlik gösterdiği anlaşılmaktadır.



Şekil 4.3. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSE)

4.1.1.4. Sentetik Veri Seti 4 (SDSEN)

Çizelge 4.7 ve Şekil 4.4 incelendiğinde SDSEN veri setine uygulanan BCO algoritması sonuçlarına göre hesaplanan indeks değerleri, V_{XB} , V_{Kwon} , V_{TS} , V_{MPC} ve V_{Fsil} indeks değerlerinin gerçek küme sayısını tahmin ettikleri gözlenmektedir. Elde edilen analiz sonuçlarına göre V_{XB} , V_{Kwon} , V_{TS} , V_{MPC} ve V_{Fsil} veri setinde bulunan küme sayısı ($c=4$); V_{PE} indeksine göre ($c=2$) olarak bulunmaktadır.

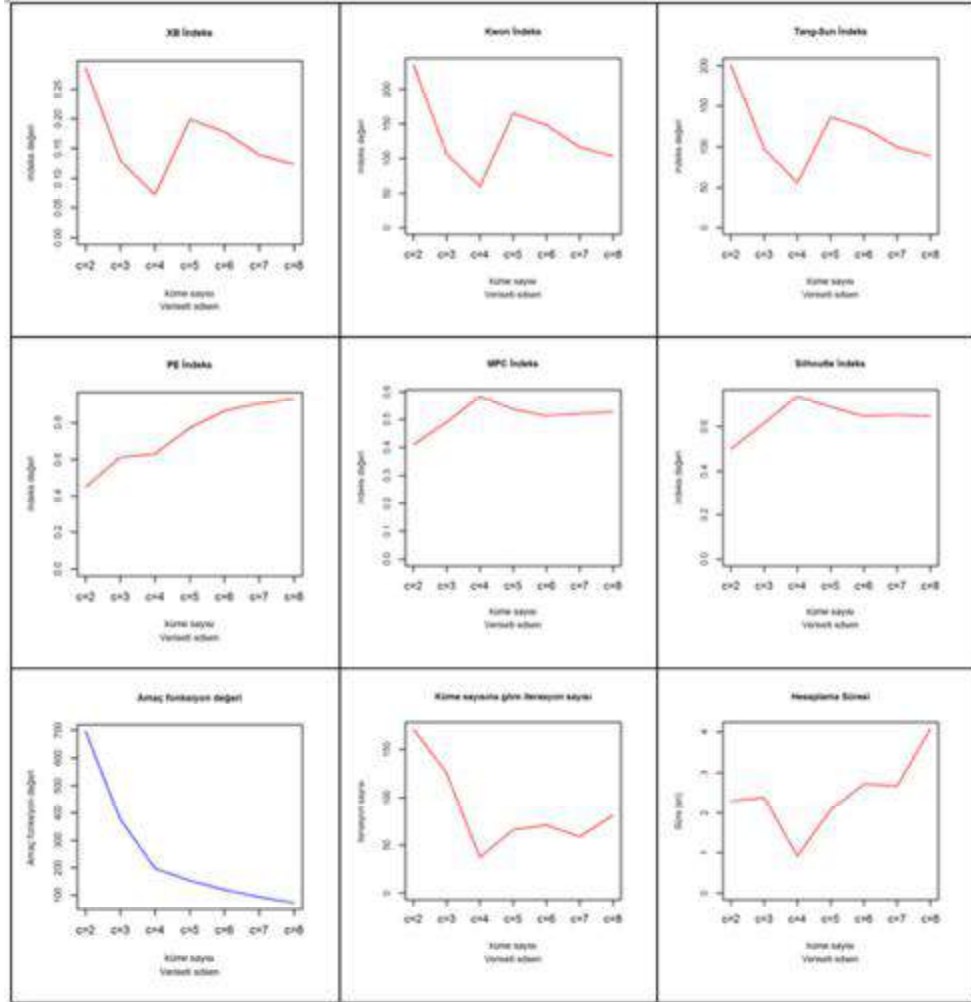
Çizelge 4.7. SDSEN veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.286	234.602	200.959	0.451	0.409	0.499
c=3	0.130	106.715	97.680	0.613	0.491	0.618
c=4	0.073	60.332	56.245	0.632	0.584	0.740
c=5	0.199	164.699	136.281	0.772	0.537	0.691
c=6	0.178	147.852	123.407	0.867	0.513	0.648
c=7	0.139	116.220	99.369	0.906	0.522	0.653
c=8	0.124	104.019	88.934	0.933	0.527	0.648

Çizelge 4.8. SDSEN veri seti için BCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	171	2.300	694.749
c=3	125	2.380	375.744
c=4	37	0.920	197.260
c=5	66	2.080	153.702
c=6	71	2.720	120.466
c=7	59	2.670	93.942
c=8	81	4.080	72.810

BCO algoritmasının SDSEN veri setindeki başarımlarını Çizelge 4.7’de görülmektedir. Buradaki görüntü itibariyle veri setinde küme sayısına yaklaşırken hesaplama süresinin düştüğü, küme sayısının artışıyla birlikte uzaklaştığı gözlenmektedir. Maksimum hesaplama süresi ise 8 (c=8)’de görülmektedir. Amaç fonksiyonunun küme sayısının artmasıyla belli oranlarda azaldığı hem Şekil 4.4. hem de Çizelge 4.8’de açıkça görülmektedir.



Şekil 4.4. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSEN)

4.1.1.5. Sentetik Veri Seti 5 (SDSR)

Kare şekilli gürültüsüz sentetik veri seti (SDSR) için BCO algoritması kümelemesi sonuçlarından hesaplanan indeks değerleri Çizelge 4.9’ da görülmektedir. Elde edilen sonuçlara göre V_{XB} , V_{Kwon} , V_T , V_{MPC} ve indeksleri veri setinde bulunan küme sayısını ($c=4$) isabetli şekilde saptamıştır. V_{PE} indeksi ise

daha önceki farklı şekilli veri setlerinde olduğu gibi küme sayısını 2 ($c=2$) olarak saptamıştır.

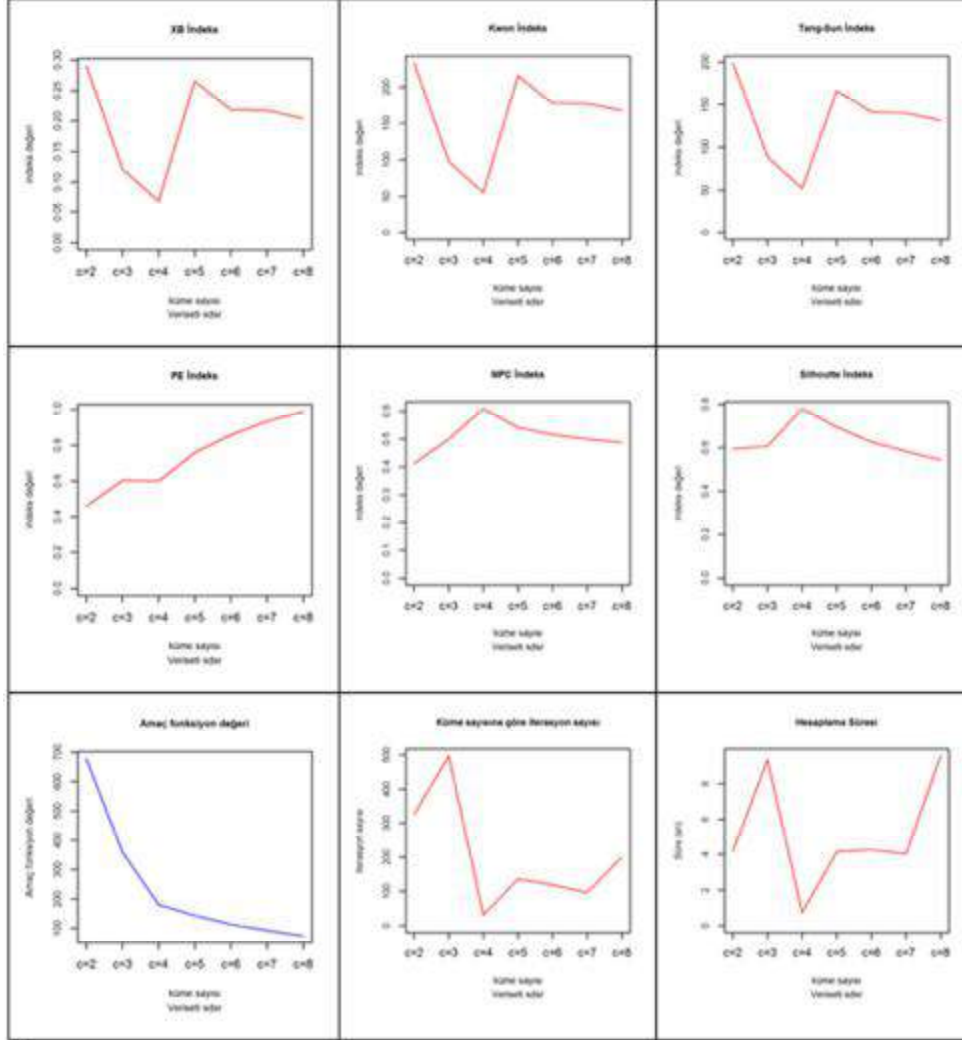
Çizelge 4.9. SDSR veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
$c=2$	0.291	232.777	198.949	0.459	0.412	0.594
$c=3$	0.121	97.251	89.379	0.603	0.499	0.606
$c=4$	0.069	55.820	52.004	0.599	0.609	0.780
$c=5$	0.266	215.164	166.639	0.760	0.542	0.695
$c=6$	0.218	177.225	141.322	0.856	0.516	0.626
$c=7$	0.217	176.989	139.983	0.932	0.499	0.583
$c=8$	0.204	167.323	131.851	0.989	0.487	0.543

Çizelge 4.10. SDSR veri seti için BCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	326	4.180	678.743
$c=3$	497	9.360	358.706
$c=4$	31	0.750	179.921
$c=5$	136	4.140	144.198
$c=6$	119	4.260	112.714
$c=7$	96	4.030	92.427
$c=8$	200	9.570	74.604

Çizelge 4.10'da BCO algoritmasının SDSR veri setindeki başarımları sonuçları görülmektedir. Küme sayısının artmasıyla amaç fonksiyon değerinin azaldığı gözlenmektedir. Şekil 4.5 incelendiğinde, hesaplama süresi ile iterasyon sayılarının artış ve azalışların benzer yönelimler gösterdiği ve amaç fonksiyon değerlerinin küme sayısının artmasıyla azaldığı anlaşılmaktadır.



Şekil 4.5. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR)

4.1.1.6. Sentetik Veri Seti 6 (SDSRN)

Çizelge 4.11’de kare şekilli-gürültülü sentetik veri seti (SDSRN) için BCO algoritmasından hesaplanan indeks değerleri görülmektedir. Elde edilen sonuçlara göre indeksler SDSR ile aynı küme sayılarını (Çizelge 4.9) tahmin etmiştir. İndekslerin tahmin ettikleri küme değerleri birebir SDSR ile aynı olmasa da

birbirlerine çok yakın değerler göstermektedir. Uygulanan algoritmanın bu veri seti üzerinde de kümelerin tespitinde iyi bir başarımlar gösterdiği anlaşılmaktadır.

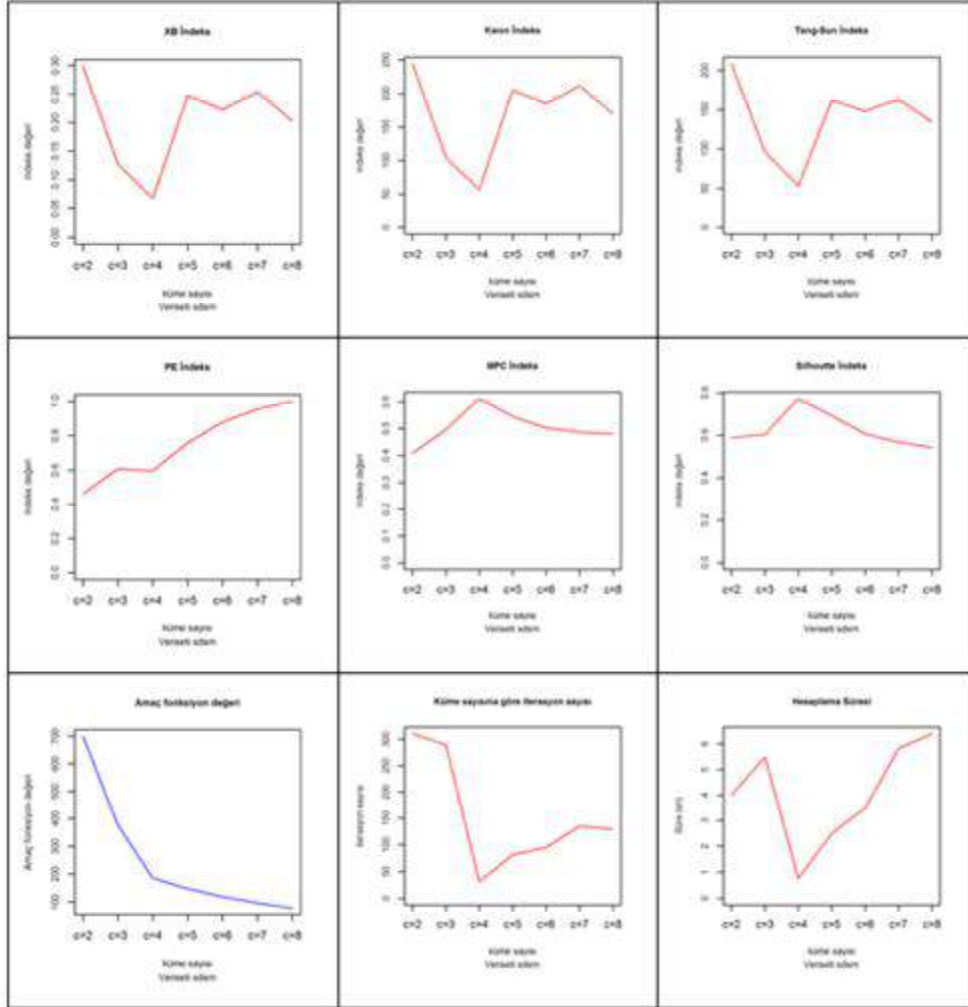
Çizelge 4.11. SDSRN veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.298	244.725	208.528	0.462	0.408	0.589
c=3	0.127	104.798	96.064	0.608	0.495	0.602
c=4	0.068	56.469	52.672	0.596	0.612	0.773
c=5	0.248	205.361	161.670	0.759	0.544	0.690
c=6	0.222	184.835	147.689	0.877	0.502	0.605
c=7	0.254	212.178	162.855	0.952	0.486	0.567
c=8	0.202	169.696	134.561	1.000	0.479	0.542

Çizelge 4.12. SDSRN veri seti için BCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	310	4.000	699.354
c=3	289	5.480	375.305
c=4	31	0.770	185.403
c=5	81	2.490	148.274
c=6	95	3.490	119.041
c=7	135	5.820	97.003
c=8	129	6.390	78.083

Çizelge 4.12 ve Şekil 4.6'da BCO algoritmasının SDSRN veri setindeki başarımlar sonuçları görülmektedir. Burada hesaplama süresi ve iterasyon sayıları üzerinden bir yorum yapmak gerekirse veri setindeki gerçek küme sayısında en düşük iterasyon sayısı elde edilmekte ve hesaplama süresi de azalmaktadır.



Şekil 4.6. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN)

4.1.1.7. Sentetik Veri Seti 7 (SDSX)

SDSX veri seti konkav şekilli-gürültüsüz kümeler içeren sentetik veri setidir. Çizelge 4.13’de BCO algoritması diğer veri setlerinde gösterdikleri başarımı bu veri setinde gösterememiştir. Küme sayısı (c=3) olan veri setinde elde edilen küme tahminlerin 5 (c=5) olarak saptanması BCO algoritmasının bu veri seti üzerinde kümeleri saptamada iyi sonuç vermediğini ortaya koymaktadır.

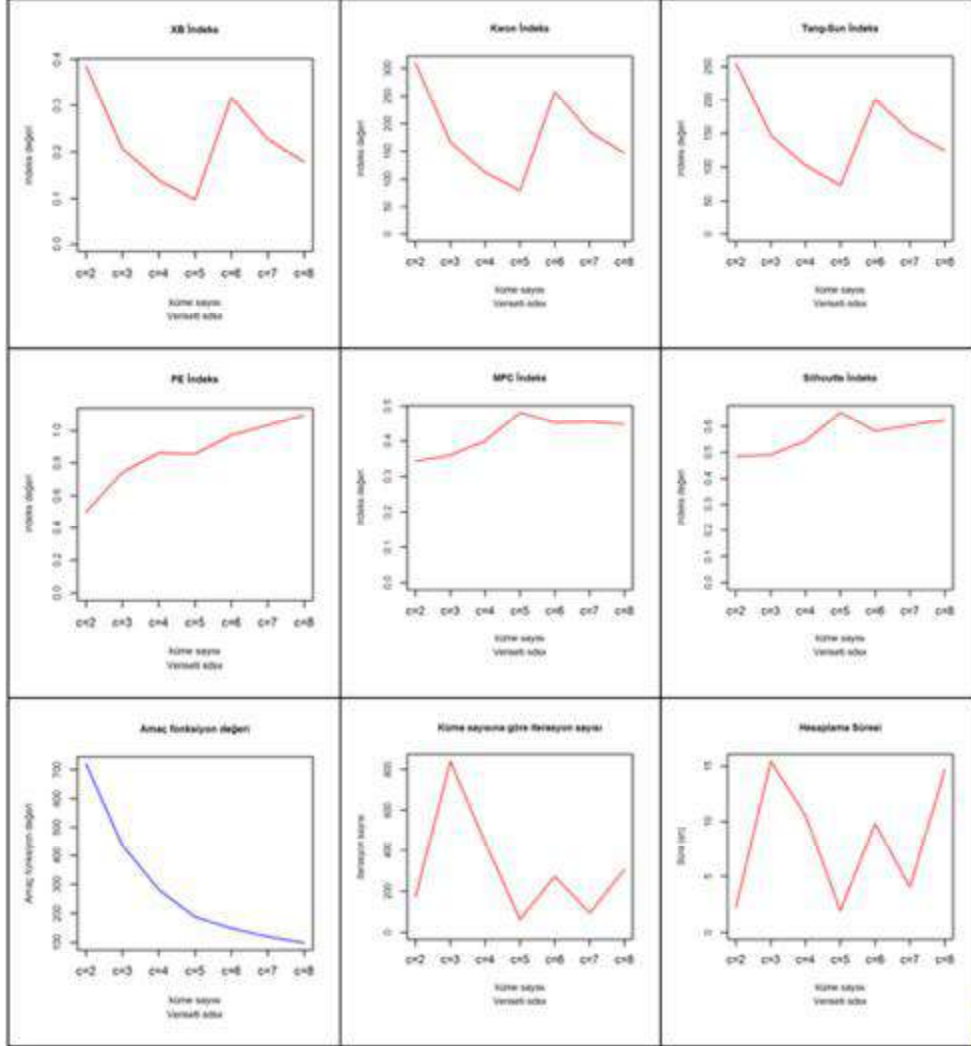
Çizelge 4.13. SDSX veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.386	309.824	255.274	0.499	0.342	0.484
c=3	0.207	166.198	147.687	0.744	0.360	0.490
c=4	0.140	112.545	102.579	0.863	0.400	0.544
c=5	0.098	79.506	73.536	0.856	0.482	0.655
c=6	0.318	257.851	200.898	0.973	0.452	0.581
c=7	0.227	185.320	152.499	1.000	0.454	0.605
c=8	0.179	146.427	124.276	1.000	0.448	0.624

Çizelge 4.14. SDSX veri seti için BCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	173	2.240	721.040
c=3	841	15.470	436.418
c=4	438	10.520	282.147
c=5	63	1.910	187.340
c=6	273	9.780	148.058
c=7	93	4.050	119.262
c=8	304	14.700	98.770

Çizelge 4.14’de BCO algoritmasının SDSX veri setindeki başarımlarını sonuçları görülmektedir. Hesaplama süresini ve iterasyon sayısı; veri setindeki küme sayısına yaklaşırken azaldığı görülmektedir. Diğer veri setleri için yapılan analiz sonuçlarında olduğu gibi, Şekil 4.7 ve Çizelge 4.14’de amaç fonksiyon değerleri incelendiğinde amaç fonksiyon değerinin küme sayısının artmasıyla birlikte azalma eğiliminde olduğu görülmektedir.



Şekil 4.7. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX)

4.1.1.8. Sentetik Veri Seti 8 (SDSXN)

Şekil 3.1’de ve Çizelge 4.15’de görülen analiz sonuçlarına göre BCO algoritmasının gürültülü noktaları bulamadığı ve SDSX ile aynı sonuçları verdiği anlaşılmaktadır.

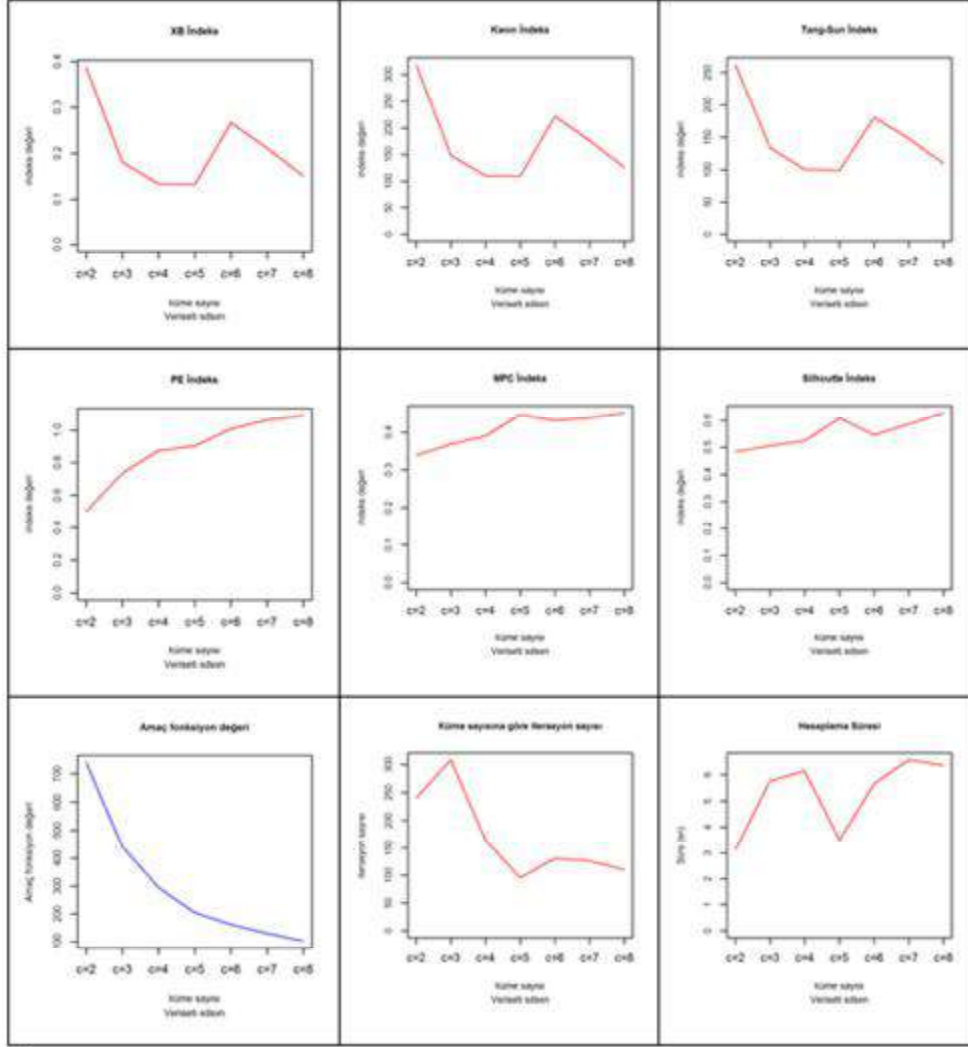
Çizelge 4.15. SDSXN veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	<i>V_{XB}</i>	<i>V_{Kwon}</i>	<i>V_{TS}</i>	<i>V_{PE}</i>	<i>V_{MPC}</i>	<i>V_{Fsil}</i>
c=2	0.388	318.831	262.581	0.502	0.339	0.485
c=3	0.180	148.529	133.716	0.736	0.369	0.505
c=4	0.134	110.308	101.054	0.874	0.390	0.525
c=5	0.132	109.428	99.145	0.906	0.447	0.608
c=6	0.266	221.44	181.065	1.000	0.432	0.547
c=7	0.211	175.943	148.013	1.000	0.439	0.586
c=8	0.151	126.236	110.096	1.000	0.453	0.629

Çizelge 4.16. SDSXN veri seti için BCO algoritması başarımı

<i>C</i>	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	240	3.130	740.293
c=3	309	5.780	440.711
c=4	164	6.140	293.640
c=5	96	3.510	204.225
c=6	131	5.660	161.757
c=7	126	6.580	129.249
c=8	111	6.360	103.301

Çizelge 4.16’de BCO algoritmasının SDSXN veri setindeki başarımlarını sonuçları görülmektedir. Hesaplama süresi ve iterasyon sayısı da gerçek kümeden farklı küme sayıları üzerinde tahmin yapmaktadır. Şekil 4.8 ve Çizelge 4.16’de amaç fonksiyon değerleri incelendiğinde amaç fonksiyon değerinin küme sayısının artmasıyla birlikte azalma eğiliminde olduğu görülmektedir.



Şekil 4.8. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSXN)

4.1.2. Gerçek Veri Setleri

4.1.2.1. Gerçek Veri Seti 1 (RDS1-Iris)

Çizelge 4.17’de RDS1-Iris gerçek veri seti için BCO algoritmasından hesaplanan indeks değerleri görülmektedir. Elde edilen sonuçlara göre V_{XB} , V_{Kwon} , V_{TS} , V_{MPC} , V_{PE} ve V_{Fsil} indeksleri veri setinde bulunan küme sayısını ($c=2$) tahmin etmiştir. Gerçek küme sayısı 3 ($c=3$) olarak değerlendirildiğinde algoritmamızın veri seti üzerinde iyi bir başarıyı gösterdiğinden bahsedemeyiz. Ancak indekslerin tahmin ettiği küme sayısından ($c=2$) sonra tahmin edilen küme sayısının 3 ($c=3$)’ü göstermesi BCO algoritmasının çok da kötü sonuç verdiği anlamına gelmemektedir. Ancak gerçek küme sayısını 2 ($c=2$) olarak değerlendirdiğimizde, tahmin edilen sonuçlara göre BCO algoritmasının RDS1-Iris veri seti üzerinde kümeleri saptamada başarılı olduğu görülmektedir.

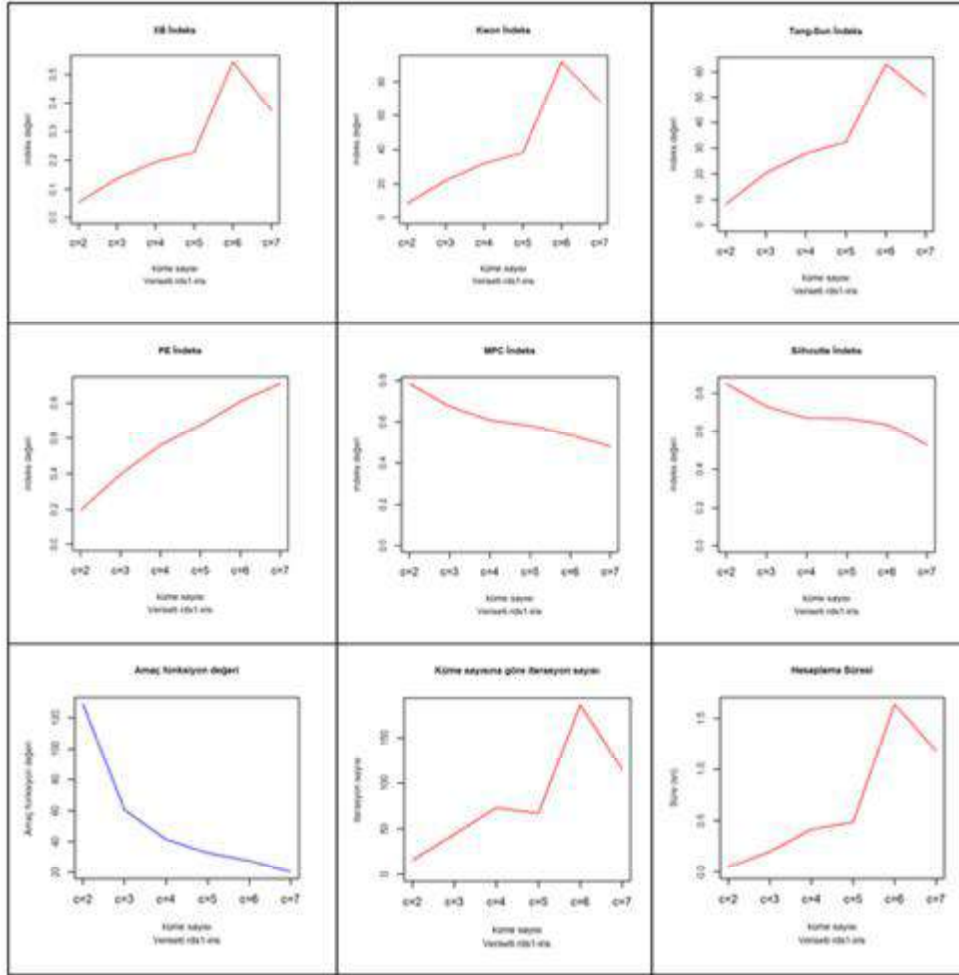
Çizelge 4.17. RDS1-Iris veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.054	8.376	8.363	0.196	0.784	0.850
c=3	0.137	21.955	20.360	0.395	0.675	0.732
c=4	0.195	31.978	27.951	0.561	0.609	0.670
c=5	0.228	38.238	32.482	0.675	0.582	0.668
c=6	0.544	91.773	63.038	0.809	0.536	0.637
c=7	0.374	68.525	50.670	0.911	0.480	0.530

Çizelge 4.18. RDS1-Iris veri seti için BCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	16	0.050	128.895
c=3	44	0.200	60.506
c=4	73	0.420	41.614
c=5	67	0.490	32.733
c=6	186	1.640	27.855
c=7	115	1.190	20.388

BCO algoritmasının RDS1-Iris gerçek veri seti üzerinde hesaplama süresi ve iterasyon başarımları Çizelge 4.18’de gösterilmiştir. Çizelge 4.18 ve Şekil 4.9 incelendiğinde hesaplama süresinin veri setindeki küme sayısına yaklaşırken düştüğü gözlemlenmektedir. Şekil 4.9’da amaç fonksiyon değerleri incelendiğinde amaç fonksiyon değerinin küme sayısının artışıyla bir azalma eğilimi (c=6 hariç) gösterdiği ve gerçek küme sayısında belirgin bir bükülme noktasına sahip olduğunu söylemek mümkündür.



Şekil 4.9. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS1-Iris)

4.1.2.2. Gerçek Veri Seti 2 (RDS2-Quail)

Çizelge 4.19’da RDS2-Quail gerçek veri seti için BCO algoritmasından hesaplanan indeks değerleri görülmektedir. Elde edilen sonuçlara göre indeks değerleri RDS1-Iris veri setinde olduğu gibi küme sayısı tahminini 2 ($c=2$) yönünde yapmaktadır. Burada da küme sayıları arasında indeks değerlerine bakıldığında aslında algoritmamızın çokta uzak bir tahminde bulunmadığını ancak sonuç itibarıyla veri seti üzerindeki kümelerin bulunması anlamında başarılı olduğunu da söylemek mümkün olmaz.

Çizelge 4.19. RDS2-Quail veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

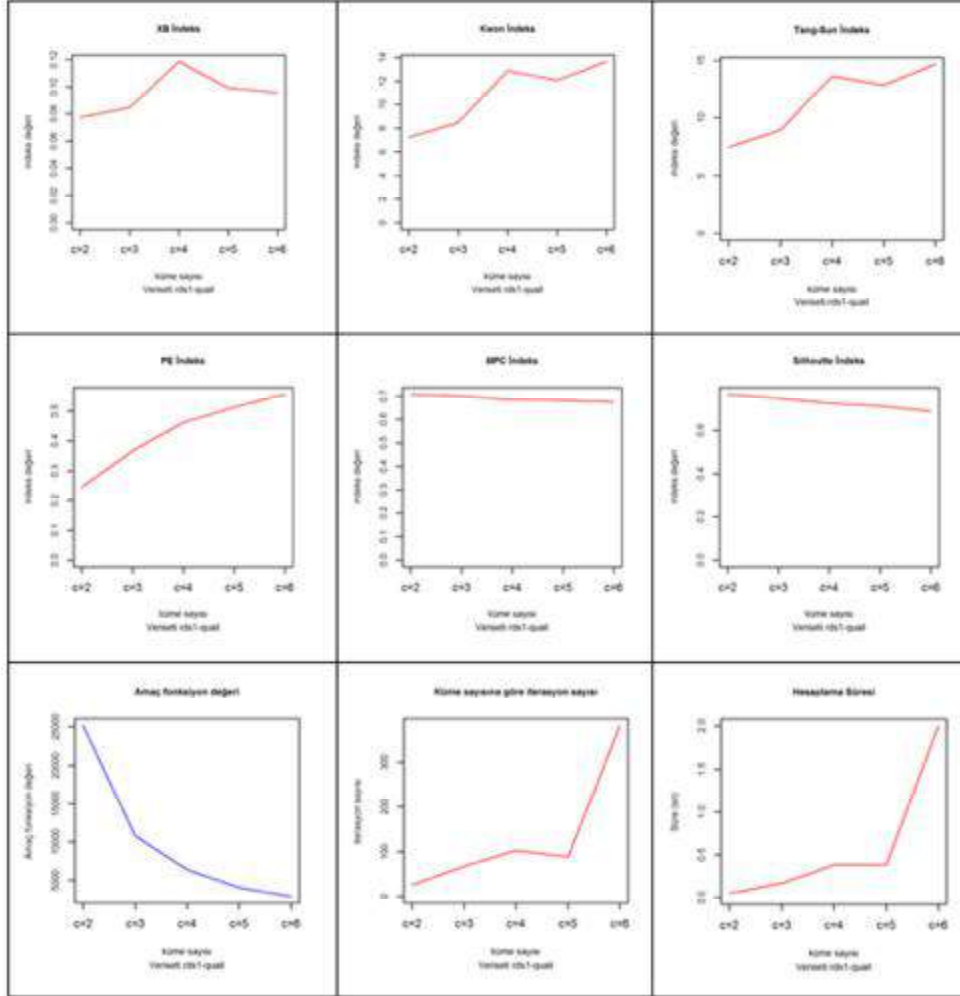
C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
$c=2$	0.077	7.209	7.458	0.246	0.709	0.769
$c=3$	0.084	8.494	8.942	0.368	0.700	0.749
$c=4$	0.119	12.909	13.648	0.462	0.684	0.726
$c=5$	0.099	12.082	12.866	0.512	0.681	0.713
$c=6$	0.096	13.679	14.688	0.558	0.674	0.688

Çizelge 4.20. RDS2-Quail veri seti için BCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	26	0.050	25160.258
$c=3$	67	0.170	10777.185
$c=4$	103	0.380	6401.727
$c=5$	88	0.380	4041.657
$c=6$	381	2.000	2896.889

BCO algoritmasının RDS2-Quail gerçek veri seti üzerinde hesaplama süresi ve iterasyon başarımları Çizelge 4.20’de gösterilmiştir. Çizelge 4.20 ve Şekil 4.10 incelendiğinde hesaplama süresi ve iterasyon sayısı değerlerine bakıldığında tahminlerini 2 ($c=2$) olarak göstermektedir.

Şekil 4.10. ve Çizelge 4.20’de amaç fonksiyon değerleri incelendiğinde amaç fonksiyon değerinin küme sayısının artışıyla yüksek oranda azalma eğilimi gösterdiği gözlemlenmektedir.



Şekil 4.10. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail)

4.1.2.3. Gerçek Veri Seti 3 (Wine)

Wine gerçek veri setinde BCO algoritması sonuçlarına göre hesaplanan indeks değerleri incelendiğinde (Çizelge 4.21 ve Şekil 4.11), optimal küme sayısını

2 ($c=2$) olarak saptamıştır. RDS1-Iris ve RDS2-Quail veri setleri üzerinde olduğu gibi bu veri setinde de BCO kümeleme algoritması küme bulmada istenilen başarıyı gösterememektedir. Ayrıca indeks değerleri arasında gerçek kümeden daha yakın başka küme tahminleri de göstermektedir ki bu durum BCO'nun başarısını RDS1-Iris ve RDS2-Quail'de yapılan uygulamadan daha başarısız olduğunu göstermektedir.

Çizelge 4.21. RDS3-Wine veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
$c=2$	0.058	10.644	10.894	0.216	0.752	0.819
$c=3$	0.126	23.598	24.213	0.380	0.686	0.728
$c=4$	0.094	18.785	19.455	0.418	0.711	0.718
$c=5$	0.099	20.379	21.076	0.510	0.680	0.701
$c=6$	0.139	30.462	31.592	0.522	0.696	0.712
$c=7$	0.085	22.932	24.233	0.505	0.717	0.738

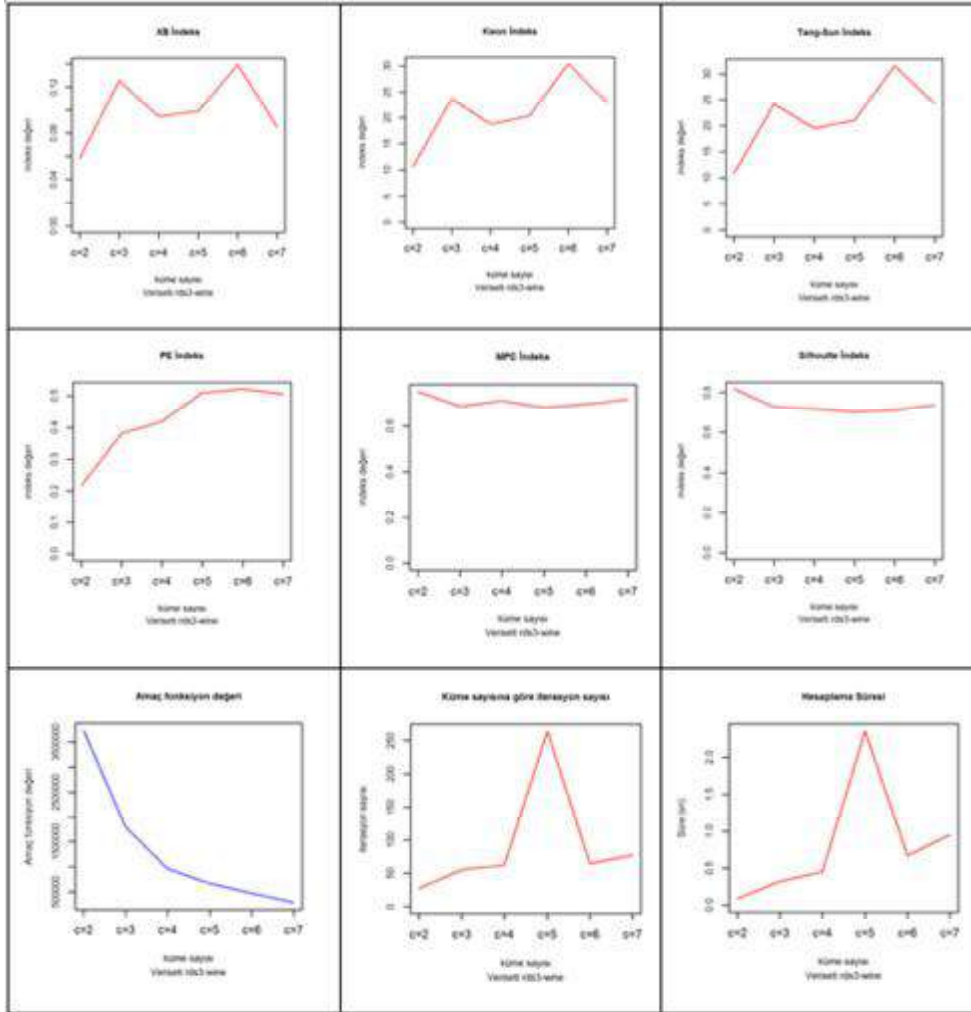
Çizelge 4.22. RDS3-Wine veri seti için BCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	27	0.090	3739139.975
$c=3$	55	0.320	1796082.760
$c=4$	62	0.450	959086.331
$c=5$	264	2.360	666042.748
$c=6$	64	0.670	475238.836
$c=7$	77	0.770	286134.365

BCO algoritmasının RDS3-Wine gerçek veri seti üzerinde hesaplama süresi ve iterasyon başarımları Çizelge 4.22'de gösterilmiştir. Çizelge 4.22'de elde edilen iterasyon sayısı ve hesaplama süreleri değerlerine göre güçlü bir ilişki olduğunu söylemek zordur. Ancak Şekil 4.11'de oluşan grafik görünümü bu durumun netliğini arttırmaktadır ki oluşan grafik görünümü nerdeyse aynıdır

diyebiliriz. Ancak iterasyon sayısı ve hesaplama süresi değerleri üzerinde oluşan büküm noktaları aynı küme sayısı ($c=5$) olması guclu bir ilişkiyi göstermektedir.

Şekil 4.11 ve Çizelge 4.22’de amaç fonksiyon değerleri incelendiğinde amaç fonksiyon değerinin küme sayısının artışıyla yüksek oranda bir azalma eğilimi gösterdiği saptanmaktadır.



Şekil 4.11. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS3-Wine)

4.1.2.4. Gerçek Veri Seti 4 (RDS4-Glass)

Çizelge 4.23’de RDS4- Glass gerçek veri seti için BCO algoritmasından hesaplanan indeks değerleri görülmektedir. Elde edilen sonuçlara göre indekslerin gerçek küme (c=6) gerçek küme sayısının altında sonuçlar üretmişlerdir. V_{XB} , V_{Kwon} , V_{TS} , V_{MPC} ve V_{Fsil} indeks değerleri küme sayısını 3 (c=3) olarak tahmin etmektedir. Yalnızca V_{PE} indeks değeri uyguladığımız bütün veri setinde olduğu gibi küme sayısını 2 (c=2) olarak vermiştir. Bu durumda BCO algoritması RDS4-Glass veri seti için küme sayısını doğru tahmin edememiştir.

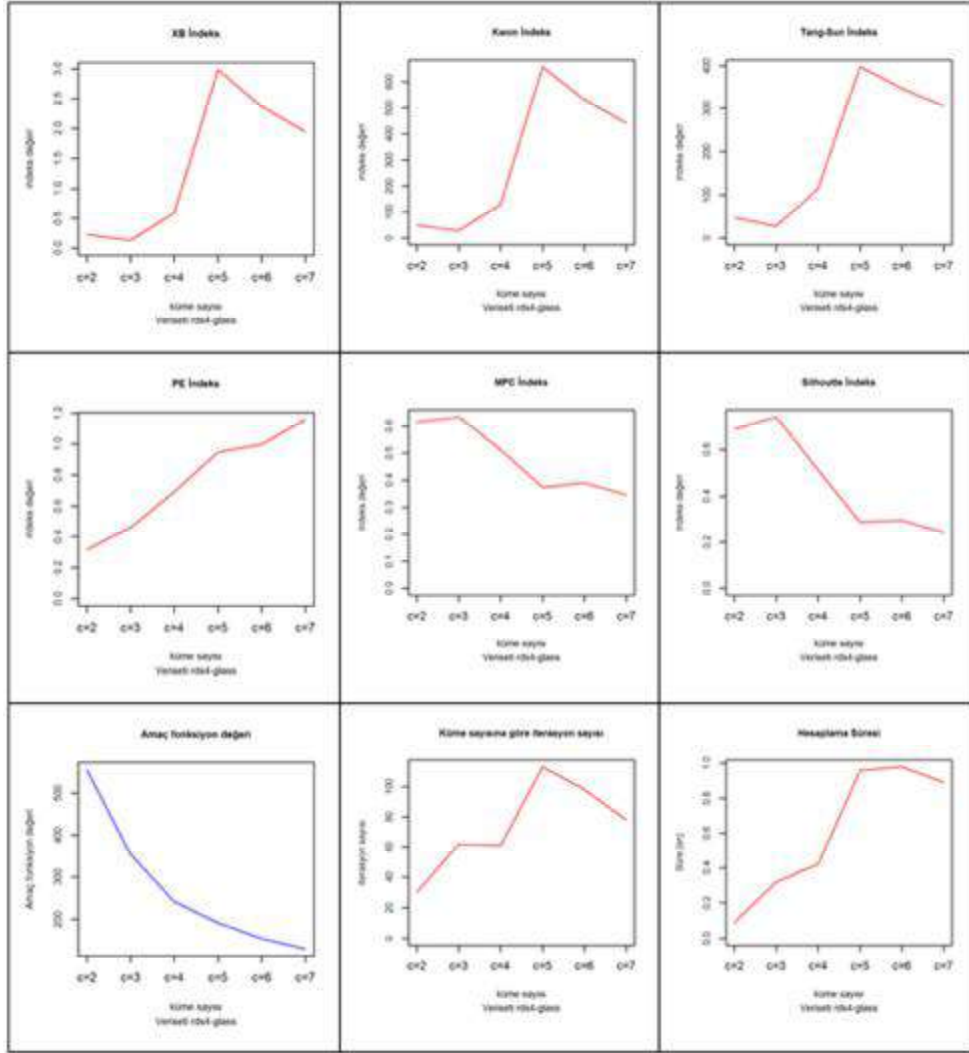
Çizelge 4.23. RDS4-Glass veri setinde BCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.225	48.376	46.610	0.317	0.615	0.688
c=3	0.129	27.920	27.403	0.461	0.635	0.743
c=4	0.591	129.532	115.471	0.690	0.513	0.512
c=5	2.989	659.322	397.608	0.948	0.374	0.289
c=6	2.358	531.306	347.217	0.996	0.392	0.296
c=7	1.945	441.391	305.940	1.000	0.346	0.242

Çizelge 4.24. RDS4-Glass veri seti için BCO algoritması başarıımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	30	0.090	556.384
c=3	62	0.320	353.112
c=4	61	0.420	241.784
c=5	113	0.960	190.751
c=6	98	0.980	154.146
c=7	78	0.890	130.261

BCO algoritmasının RDS4-Glass gerçek veri seti üzerinde hesaplama süresi ve iterasyon sayıları Çizelge 4.24'de gösterilmiştir. Sonuçlara bakıldığında en az hesaplama süresi ve iterasyonun küme sayısının 2 olması ($c=2$) durumunda elde edildiği görülmektedir.



Şekil 4.12. BCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS4-Glass)

4.2. OCO Algoritması

4.2.1. Sentetik Veri Setleri

4.2.1.1. Sentetik Veri Seti 1 (SDSC)

Çizelge 4.25 incelendiğinde T değerlerinin U değerlerine dönüştürülmesi ile elde edilen U' değerlerinden hesaplanan genelleştirilmiş bulanık indeks değerlerine göre optimal küme sayısının V_{PE} indeksi dışında 4 olarak önerildiği görülmektedir.

Çizelge 4.25. SDSC veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	676.008	5.40e+04	1727.883	0.693	0.000	0.356
c=3	0.304	243.636	205.382	0.887	0.222	0.614
c=4	0.140	112.543	99.304	0.930	0.355	0.799
c=5	1359.852	1.09e+05	855.950	1.000	0.291	0.498
c=6	1642.213	1.32e+05	849.420	1.000	0.253	0.260
c=7	8090.595	6.54e+05	832.768	1.000	0.220	0.017
c=8	3452.190	2.80e+05	782.055	1.000	0.192	-0.247

Çizelge 4.26'da hem U ve hem de T değerlerini kullanarak hesaplanan uzatılmış indeks değerlerine göre optimal küme sayısının 4 olduğu anlaşılmaktadır. Ancak yeni bir indeks olarak tanımlanan EPESIM indeksinin küme sayılarının artışı ile birlikte monotonik bir azalma eğilimi gösterdiği (Şekil 4.14) ve optimal küme sayısının bulunmasında etkin bir indeks olmadığı anlaşılmaktadır.

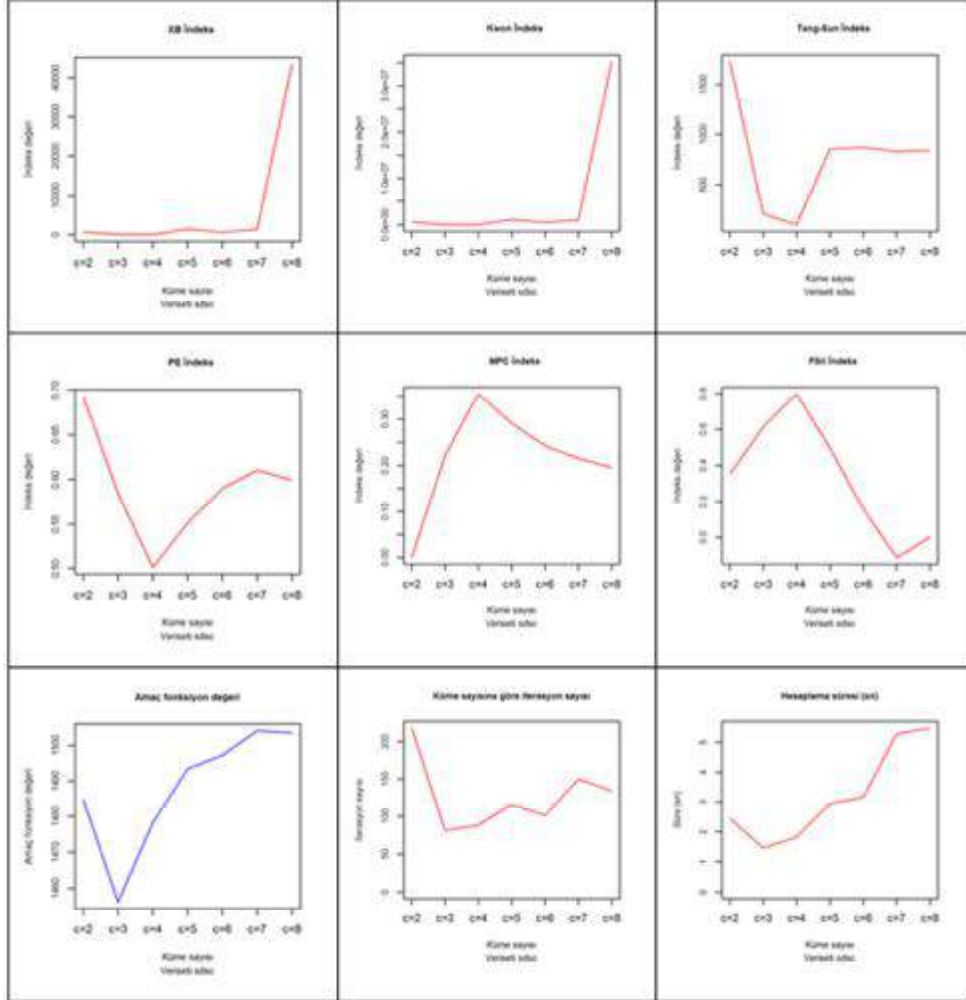
Çizelge 4.26. SDSC veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	1235.451	9.88e+04	3157.824	20.264	6.508
c=3	0.466	3.73e+01	314.717	16.844	5.112
c=4	0.203	1.62e+01	143.668	13.292	4.006
c=5	2312.763	1.85e+05	1450.240	12.799	3.852
c=6	3159.826	2.53e+05	1626.208	12.314	3.703
c=7	17490.151	1.40e+06	1787.334	11.575	3.478
c=8	8170.382	6.57e+05	1832.888	10.956	3.291

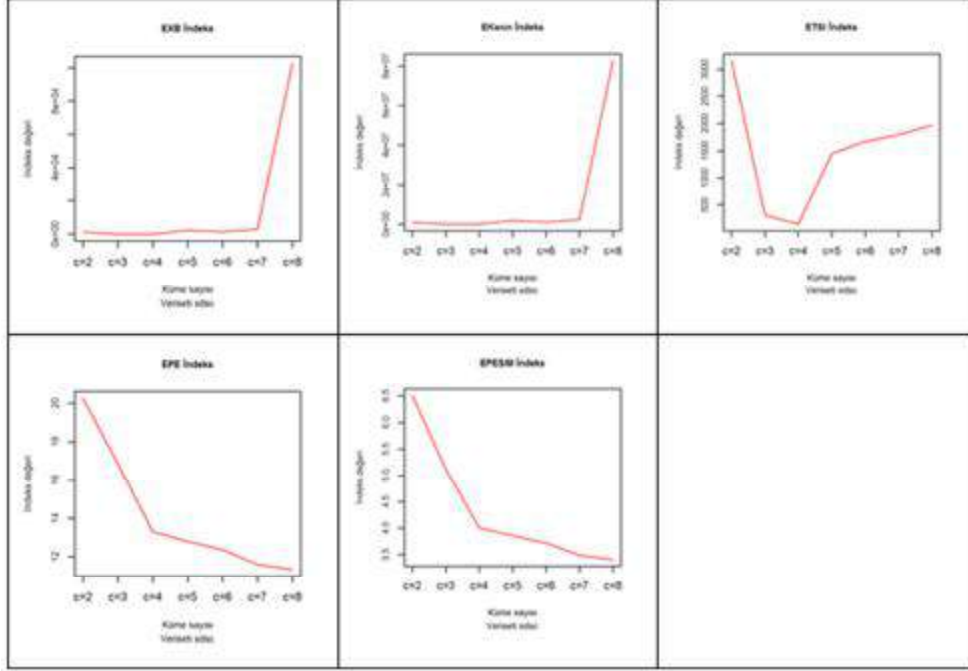
Çizelge 4.27’de iterasyon sayıları ve hesaplama süreleri incelendiğinde veri setindeki gerçek küme sayısına yakın küme sayısı seçildiğinde daha hızlı yaklaşma sağlandığından iterasyon sayıları azalmaktadır.

Çizelge 4.27. SDSC veri seti için OCO algoritması başarımı

<i>C</i>	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	218	7.380	1133.047
c=3	81	4.080	1068.875
c=4	88	5.910	753.857
c=5	116	9.780	845.810
c=6	134	13.270	901.616
c=7	156	18.110	912.770
c=8	161	21.230	902.383



Şekil 4.13. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSC)



Şekil 4.14. OCO algoritması uzatılmış indeks değerleri (SDSC)

4.2.1.2. Sentetik Veri Seti 2 (SDSCN)

Çizelge 4.28’de elde edilen indeks değerleri incelendiğinde, tüm değerlerin gerçek küme sayısını ($c=4$) isabetli bir şekilde tahmin ettiği gözlenmektedir. Bu durumda, uygulanan algoritmanın kümelerin tespitinde iyi bir başarıyı gösterdiğini söylemek mümkündür.

Çizelge 4.28. SDSCN veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
$c=2$	88718.97	$7.27e+06$	1640.047	0.693	0.000	-0.485
$c=3$	0.471	$3.86e+01$	305.061	0.639	0.148	0.596
$c=4$	0.131	$1.07e+01$	96.879	0.531	0.311	0.778
$c=5$	64.217	$5.29e+03$	958.948	0.580	0.244	0.456
$c=6$	148.997	$1.23e+04$	974.030	0.596	0.209	0.156
$c=7$	15306.53	$1.26e+06$	898.998	0.606	0.194	-0.172
$c=8$	5950.888	$4.92e+05$	949.822	0.592	0.175	0.000

Çizelge 4.29'da uzatılmış indeks değerlerinin, V_{EPE} ve V_{EPESIM} indeks değerleri dışındaki tüm indeks değerlerin gerçek küme sayısını ($c=4$) isabetli bir şekilde tahmin ettiği gözlenmektedir. V_{EPE} ve V_{EPESIM} indeksleri SDSC veri setine benzer yapı ve değerler oluşturmaktadır. Genel anlamda değerlerin SDSC veri setinden elde edilen değerlere yakın ve aynı küme sayılarına isabet etmesi, veri setinin gürültülü ya da gürültüsüz olması sonuçları çokta etkilemediğini göstermektedir.

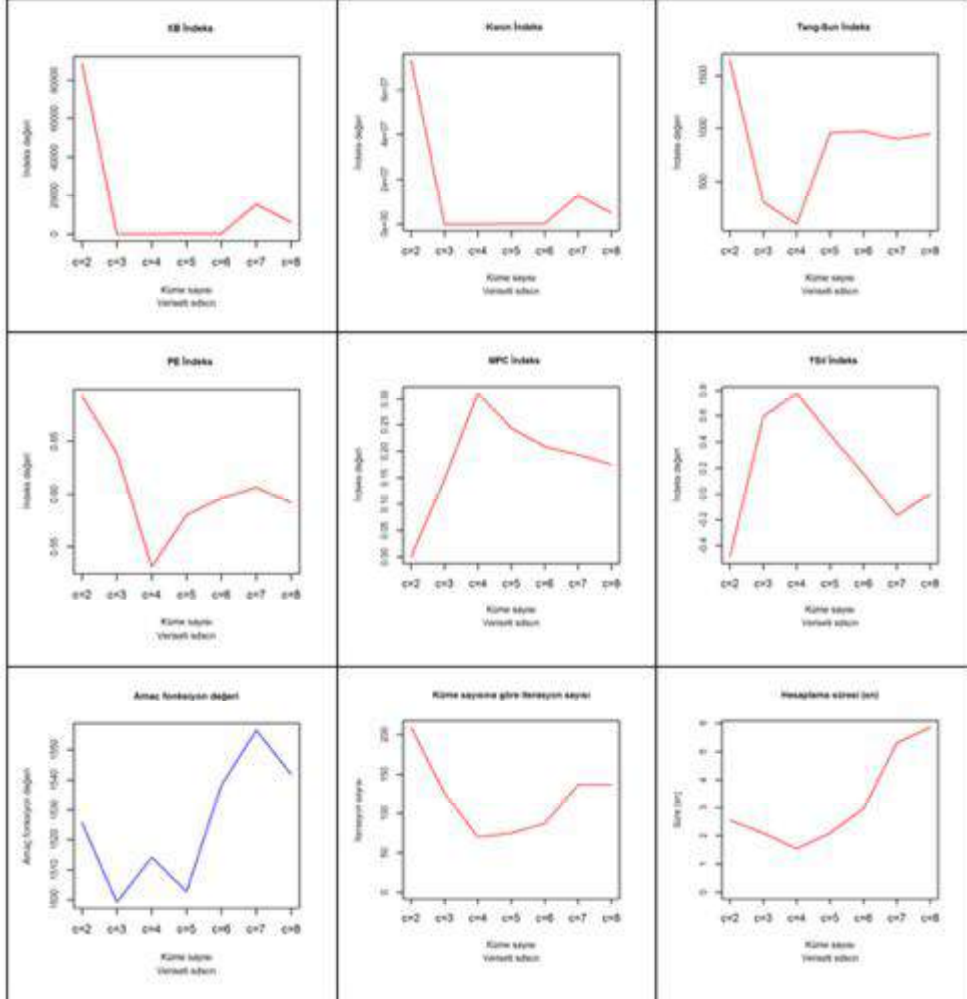
Çizelge 4.29. SDSCN veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	168896.252	1.38e+07	3122.194	20.971	6.738
c=3	0.751	6.16e+01	486.457	18.289	5.558
c=4	0.195	1.60e+01	143.592	14.230	4.289
c=5	111.318	9.15e+03	1657.379	13.847	4.168
c=6	298.313	2.45e+04	1942.207	13.360	4.017
c=7	33791.625	2.78e+06	1972.599	12.439	3.738
c=8	14443.396	1.18e+06	2290.084	12.212	3.669

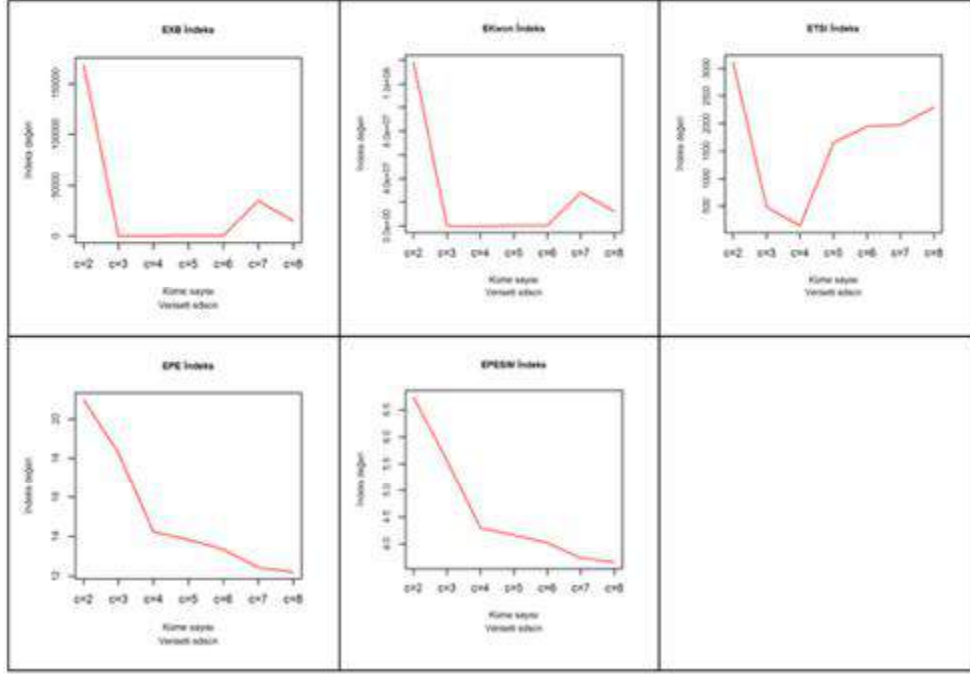
Çizelge 4.30'da elde edilen sonuçlar gerçek küme sayısına($c=4$) uygun değerleri göstermektedir.

Çizelge 4.30. SDSCN veri seti için OCO algoritması başarıımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	209	2.560	1525.677
c=3	126	2.100	1499.281
c=4	70	1.540	1514.102
c=5	75	2.090	1502.848
c=6	87	2.970	1538.370
c=7	137	5.300	1556.528
c=8	137	5.850	1541.976



Şekil 4.15. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSCN)



Şekil 4.16. OCO algoritması uzatılmış indeks değerleri (SDSCN)

4.2.1.3. Sentetik Veri Seti 3 (SDSE)

Çizelge 4.31’de sonuçlar incelendiğinde indeks değerlerinin isabet ettikleri kümeler arasında farklılıklar gözlenmektedir. Hiç bir indeks değerinin gerçek küme sayısını isabet edememesi durumu uygulanan algoritmanın veri seti üzerinde basarım gösteremediği anlamına gelse de V_{PE} ve V_{TS} dışındaki indeks sonuçları Şekil 4.17 üzerindeki donum noktaları da göz önüne alındığında gerçek küme sayısına en yakın tahminleri veren indeks değerleridir.

Çizelge 4.31. SDSE veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	5.05e+03	4.04e+06	1614.334	0.693	0.000	-0.298
c=3	0.28e+01	2.26e+03	1420.017	0.700	0.056	0.311
c=4	3.16e+03	2.53e+06	1392.874	0.659	0.113	0.246
c=5	1,51e+01	1.21e+04	1220.396	0.623	0.139	0.464
c=6	1.84e+05	1.48e+08	1182.281	0.610	0.135	0.134
c=7	8.42e+03	6.77e+06	1140.685	0.599	0.135	-0.101
c=8	3.67e+02	2.95e+05	1066.217	0.598	0.128	-0.318

Çizelge 4.32'deki uzatılmış indeks değerleri, genelleştirilmiş indeks değerlerinde olduğu gibi herhangi bir indeks tarafından isabetli bir tahminde bulunamamaktadır.

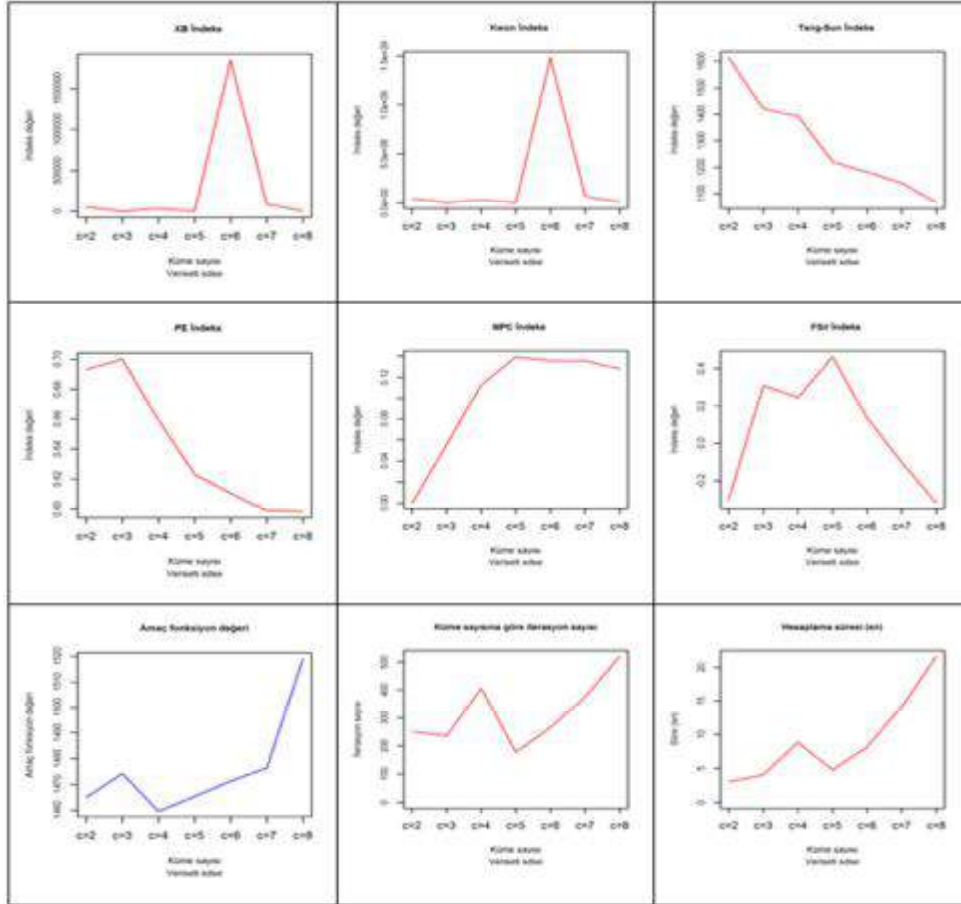
Çizelge 4.32. SDSE veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EFE}	V_{EPESIM}
c=2	9.53e+03	7.62e+06	3041.969	20.147	6.487
c=3	0.62e+01	5.00e+03	3129.513	19.041	5.807
c=4	7.11e+03	5.69e+06	3125.993	16.047	4.848
c=5	3.08e+01	2.47e+04	2476.254	15.388	4.635
c=6	4.25e+05	3.40e+08	2713.559	14.525	4.369
c=7	2.11e+04	1.69e+07	2847.741	13.365	4.017
c=8	1.00e+03	8.09e+05	2919.442	12.206	3.667

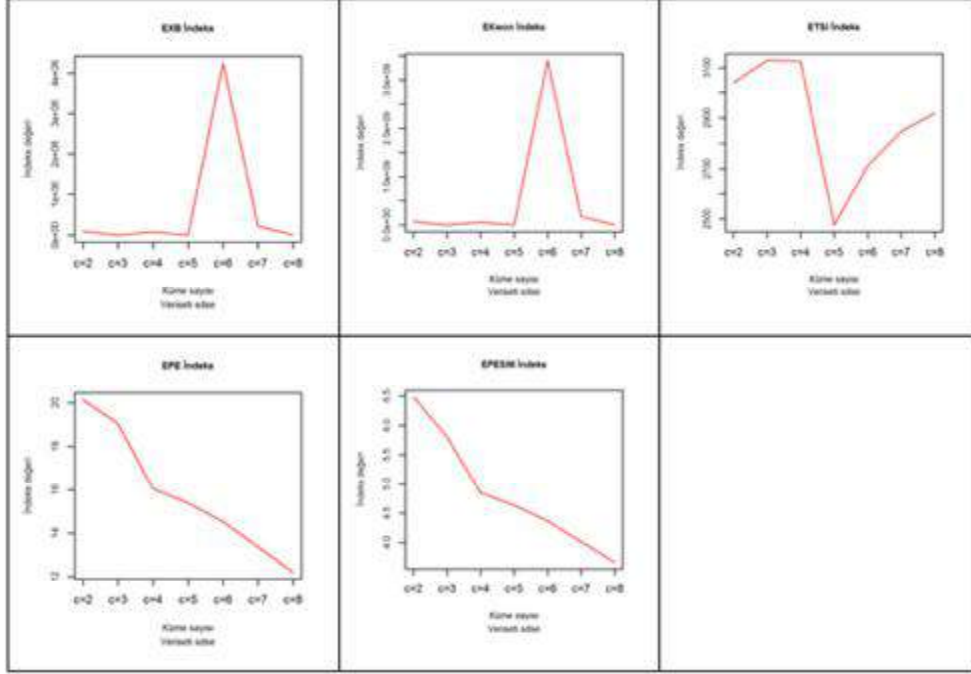
Çizelge 4.33'de veri seti için OCO algoritması başarımlarını göstermektedir. Çizelge gözlemlendiğinde elde edilen minimum değerlerin gerçek küme sayısının aksine farklı küme sayılarını göstermektedir.

Çizelge 4.33. SDSE veri seti için OCO algoritması başarıımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	251	3.050	1465.025
c=3	235	4.030	1474.214
c=4	405	8.850	1459.638
c=5	177	4.760	1465.333
c=6	265	8.220	1471.377
c=7	374	14.170	1476.355
c=8	520	21.670	1519.114



Şekil 4.17. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarıımı (SDSE)



Şekil 4.18. OCO algoritması uzatılmış indeks değerleri (SDSE)

4.2.1.4. Sentetik Veri Seti 4 (SDSEN)

Çizelge 4.34’te uygulanan indeks değerlerinin kümelenme sayısına göre değişimi gösterilmektedir. İndeks değerlerinin optimal değere ulaştıkları küme sayıları gerçek küme sayısına ($c=4$) isabet etmektedir. Durum itibariyle veri seti üzerine uygulanan OCO algoritmasının iyi bir basarım gösterdiğini söylemek mümkündür.

Çizelge 4.34. SDSEN veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	2.51e+05	2.06e+08	1639.603	0.693	0.000	-0.472
c=3	0.08e+01	7.09e+01	484.612	0.674	0.097	0.610
c=4	0.02e+01	1.92e+01	163.928	0.578	0.241	0.744
c=5	3.00e+02	2.47e+05	1088.395	0.593	0.209	0.471
c=6	9.17e+01	7.56e+04	1034.992	0.603	0.183	0.104
c=7	4.05e+04	3.35e+07	1028.653	0.602	0.175	-0.141
c=8	2.52e+04	2.08e+07	978.748	0.610	0.160	0.000

Çizelge 4.35 bize uzatılmış indeks değerlerini göstermektedir. Burada uzatılmış indeks değerlerinin minimuma (optimal) ulaştıkları değerlere bakıldığında V_{EPE} ve V_{EPESIM} uzatılmış indeksler tahminlerinde başarılı olamasa da diğer uzatılmış indeks değerleri tahminlerinde gerçek küme sayısını ($c=4$) göstermektedir.

Çizelge 4.35. SDSEN veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

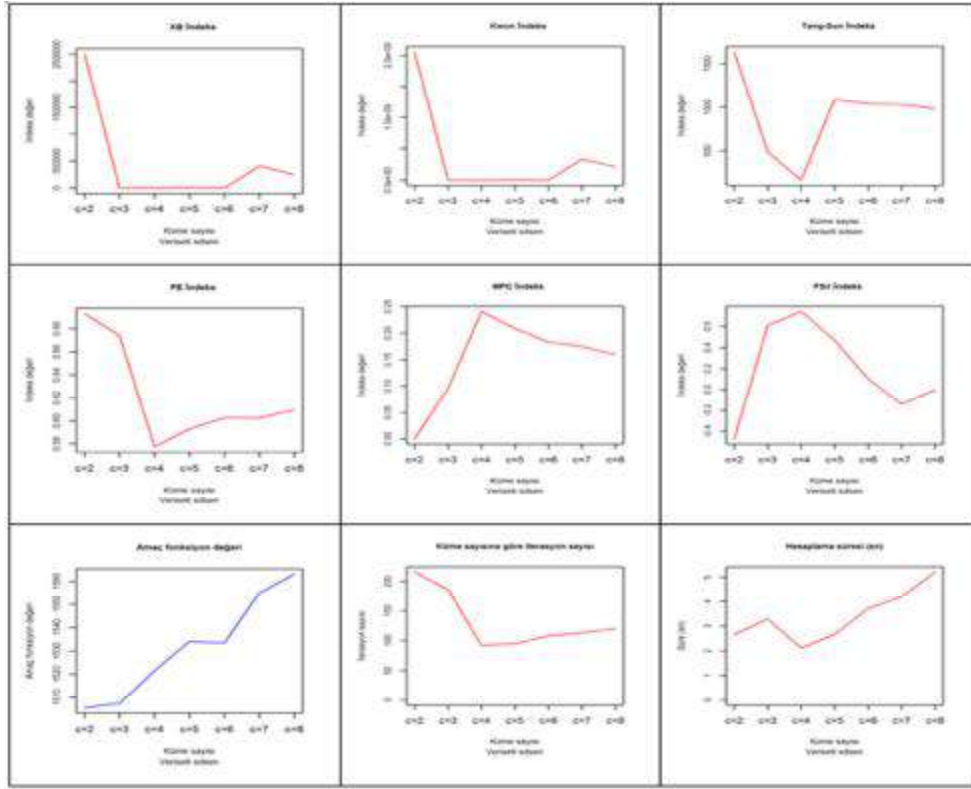
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	4.75e+05	3.89e+08	3102.207	20.653	6.645
c=3	0.01e+01	1.20e+02	826.208	18.885	5.745
c=4	0.03e+01	3.06e+01	260.715	15.082	4.549
c=5	5.60e+01	4.60e+05	2023.823	14.189	4.272
c=6	1.90e+02	1.56e+05	2145.572	13.407	4.032
c=7	9.25e+04	7.60e+07	2334.603	12.381	3.721
c=8	6.42e+04	5.28e+07	2475.878	11.176	3.357

Çizelge 4.36 ve Sekil 4.19 incelendiğinde, hesaplama süresi ve iterasyon sayılarının gerçek küme sayısına ulaştıktan sonra sabit bir şekilde artış

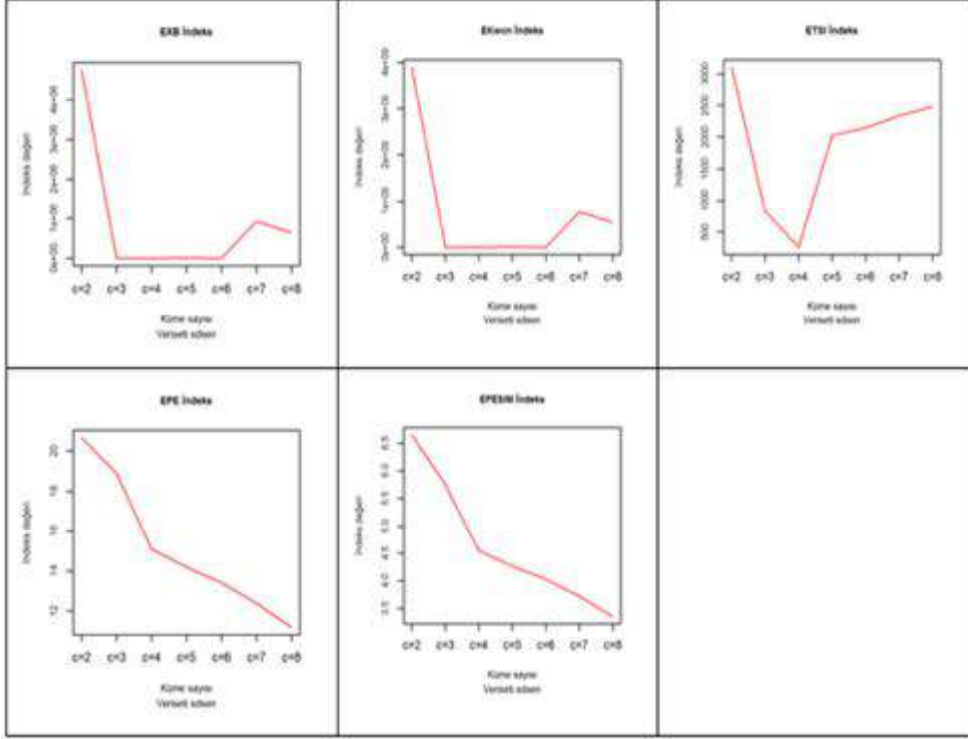
gösterdikleri, amaç fonksiyonun ise bastan sona küçük oranlarda artış gösterdiği gözlenmektedir.

Çizelge 4.36. SDSEN veri seti için OCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	216	2.650	1505.437
$c=3$	186	3.280	1507.231
$c=4$	92	2.110	1521.481
$c=5$	95	2.670	1533.959
$c=6$	108	3.700	1533.343
$c=7$	113	4.190	1554.856
$c=8$	120	5.210	1563.213



Şekil 4.19. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSEN)



Şekil 4.20. OCO algoritması uzatılmış indeks değerleri (SDSEN)

4.2.1.5. Sentetik Veri Seti 5 (SDSR)

Kare Şekilli Gürültüsüz sentetik veri setine uygulanan OCO kümeleme algoritmasından elde edilen indeks değerleri Çizelge 4.37’de gösterilmektedir. Genel anlamda indeks değerlerinin uygun min-max değerleri incelendiğinde algoritmanın veri seti üzerinde basarım gösterdiği saptanmaktadır..

Çizelge 4.37. SDSR veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	2157.993	1.72e+05	1703.325	0.693	0.000	-0.009
c=3	0.445	3.56e+01	283.328	0.620	0.172	0.613
c=4	0.151	1.21e+01	107.344	0.530	0.313	0.780
c=5	192.430	1.54e+04	928.936	0.577	0.249	0.458
c=6	277.333	2.23e+04	891.317	0.598	0.218	0.185
c=7	716896.79	5.79e+07	884.100	0.607	0.195	-0.192
c=8	6381.209	5.15e+05	942.373	0.593	0.182	-0.103

Çizelge 4.38 bize uzatılmış indeks değerlerini göstermektedir. Burada sonuçlar diğer gürültülü ve gürültüsüz şekilli veri setlerinden elde edilen sonuçlara benzer bir tablo ve değerler göstermektedir. Ayrıca Sekil 4.21 ve Sekil 4.22. arasında bir inceleme yapıldığında indeks değerlerinin oluşturdukları tablolar da benzer yapılar oluşturmaktadır.

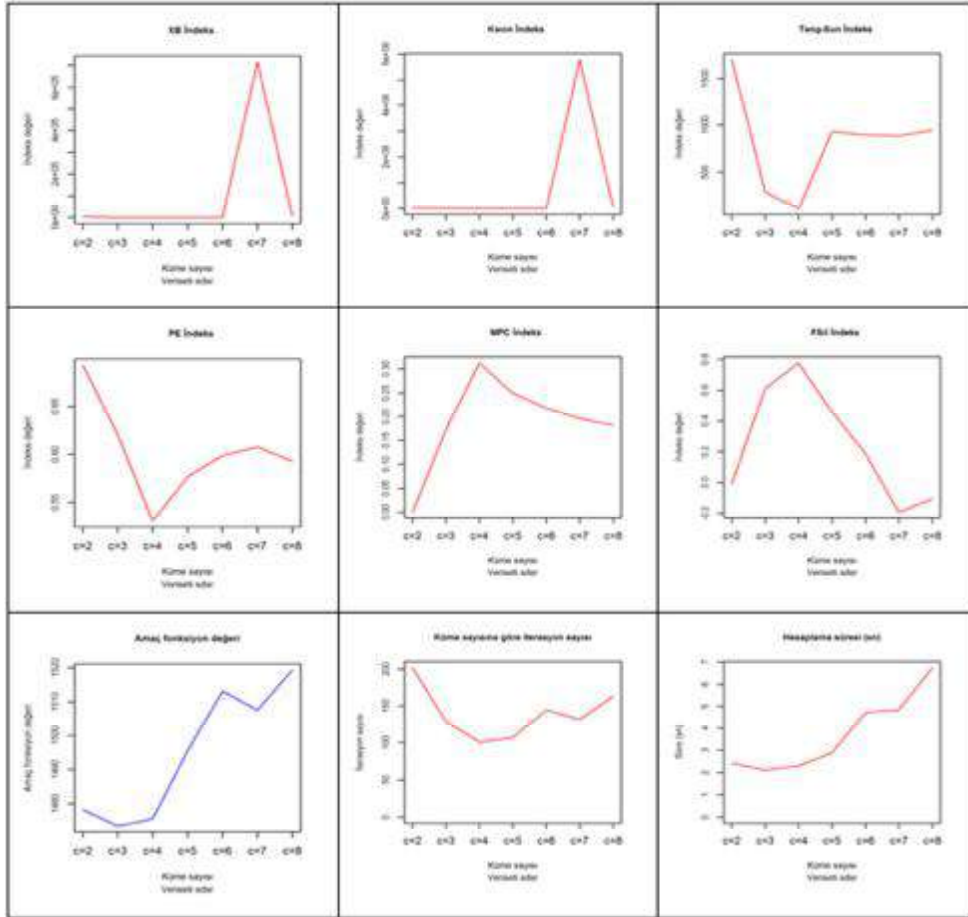
Çizelge 4.38. SDSR veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EFE}	V_{EPESIM}
c=2	3964.469	3.17e+05	3129.194	20.192	6.486
c=3	0.704	5.63e+01	447.198	17.660	5.365
c=4	0.226	1.81e+01	159.857	13.880	4.184
c=5	340.052	2.73e+04	1636.011	13.425	4.041
c=6	552.984	4.44e+04	1768.471	12.673	3.811
c=7	1618736.875	1.30e+08	1983.159	12.100	3.636
c=8	15272.876	1.22e+06	2239.892	11.655	3.501

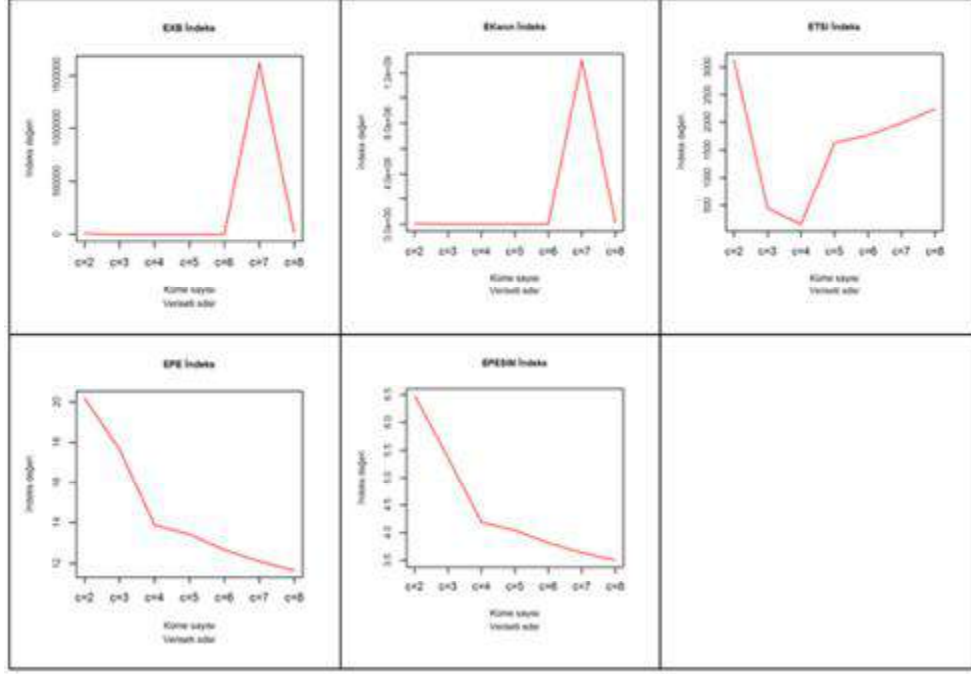
Çizelge 4.39'da iterasyon sayısının gerçek küme sayısında (c=4) minimum olduğu görülse de hesaplama suresinin 3 (c=3)'te daha hızlı çalıştığını göstermektedir.

Çizelge 4.39. SDSR veri seti için OCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	202	2.420	1478.281
$c=3$	129	2.110	1473.473
$c=4$	101	2.300	1475.568
$c=5$	107	2.890	1495.461
$c=6$	145	4.720	1513.170
$c=7$	132	4.860	1507.638
$c=8$	163	6.750	1519.379



Şekil 4.21. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR)



Şekil 4.22. OCO algoritması uzatılmış indeks değerleri (SDSR)

4.2.1.6. Sentetik Veri Seti 6 (SDSRN)

Çizelge 4.40 incelendiğinde Kare Şekilli Gürültülü sentetik veri setine uygulanan OCO kümeleme algoritmasından elde edilen indeks değerleri tahminlerinde SDSR' de olduğu gibi aynı başarıyı göstermektedir.

Çizelge 4.40. SDSRN veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	6588.817	5.40e+05	1689.688	0.693	0.000	-0.067
c=3	8.360	6.85e+02	1470.880	0.712	0.039	0.550
c=4	0.173	1.42e+01	124.389	0.537	0.302	0.773
c=5	52.294	4.31e+03	966.199	0.582	0.240	0.452
c=6	2888.683	2.38e+05	971.401	0.589	0.221	0.149
c=7	48727.589	4.03e+06	908.736	0.604	0.192	-0.202
c=8	40774.338	3.38e+06	860.316	0.616	0.176	-0.523

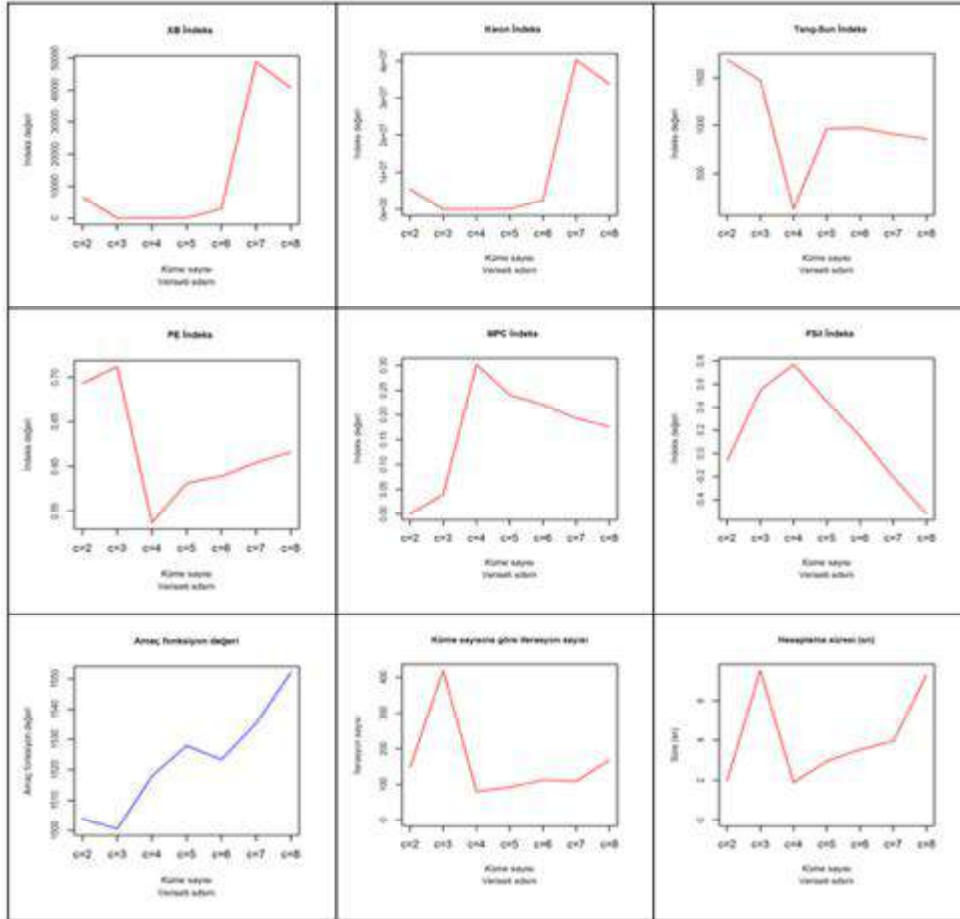
Çizelge 4.41 bize uzatılmış indeks değerleri sonuçlarını göstermektedir ki yine SDSR (Çizelge 4.38)'de olduğu gibi V_{EPE} ve V_{EPESIM} dışındaki indeksler tahminlerini gerçek küme sayısı olarak isabet ettirmişlerdir.

Çizelge 4.41. SDSRN veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

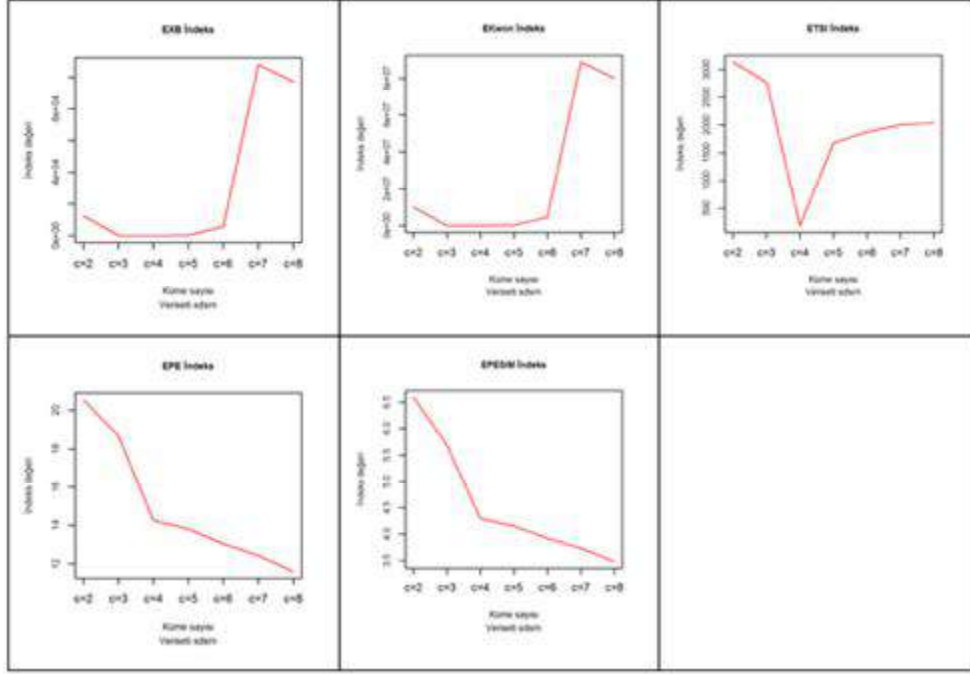
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	12228.996	1.00e+06	3136.100	20.539	6.594
c=3	15.789	1.29e+03	2777.481	18.654	5.692
c=4	0.258	2.12e+01	185.022	14.243	4.294
c=5	91.077	7.49e+03	1677.849	13.794	4.152
c=6	5602.259	4.61e+05	1876.324	13.041	3.922
c=7	107807.003	8.87e+06	1998.793	12.425	3.734
c=8	97323.473	8.02e+06	2037.341	11.604	3.486

Çizelge 4.42. SDSRN veri seti için OCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	147	1.970	1503.846
$c=3$	420	7.530	1500.544
$c=4$	80	1.880	1517.917
$c=5$	92	2.950	1527.872
$c=6$	112	3.530	1523.330
$c=7$	109	3.950	1535.162
$c=8$	167	7.280	1552.039



Şekil 4.23. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN)



Şekil 4.24. OCO algoritması uzatılmış indeks değerleri (SDSRN)

4.2.1.7. Sentetik Veri Seti 7 (SDSX)

Konkav Şekilli Gürültüsüz veri setinden elde edilen genelleştirilmiş indeks sonuçları Çizelge 4.43 gösterilmektedir. V_{XB} ve V_{Kwon} indeks değerleri isabetli tahminlerde bulunsalar da özellikle V_{MPC} ve V_{Fsil} indeksleri için Şekil 4.25, Ek 2.7 ve Ek 2.7(Devamı)'de küme sayısına göre oluşan yapılar üzerinden tahmin yapmak çok zordur.

Çizelge 4.43. SDSX veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	1.88e+06	1.51e+10	1602.144	0.693	0.000	0.000
c=3	1.29e+05	1.03e+08	1604.074	0.732	0.000	0.000
c=4	1.30e+05	1.04e+08	1605.972	0.693	0.001	0.000
c=5	2.84e+05	2.27e+08	1616.145	0.644	0.002	0.000
c=6	2.68e+05	2.15e+08	1612.879	0.598	0.001	0.000
c=7	2.70e+06	2.17e+09	1621.928	0.557	0.002	0.000
c=8	5.43e+05	4.35e+08	1628.932	0.522	0.002	0.000

Çizelge 4.44 incelendiğinde uzatılmış indeks değerlerinin Çizelge 4.43’de elde edilen geliştirilmiş indeks değerlerine göre daha net ve sağlıklı sonuçlar verdiğini söylemek mümkündür. V_{XB} ve V_{Kwon} indeksleri veri seti üzerinde geliştirilmiş indeks değerlerinde olduğu gibi tutarlı bir yapı ortaya koyarak tahminlerini isabetli bir şekilde yapmaktadır. V_{ETS} tahmini 2 (c=2) olarıktan yapmaktadır ve son olarak V_{EPE} ve V_{EPESIM} indeksleri daha önceki veri setleri üzerinde olduğu gibi gerçek küme (c=3)’den uzak bir tahminde bulunmaktadır.

Çizelge 4.44. SDSX veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

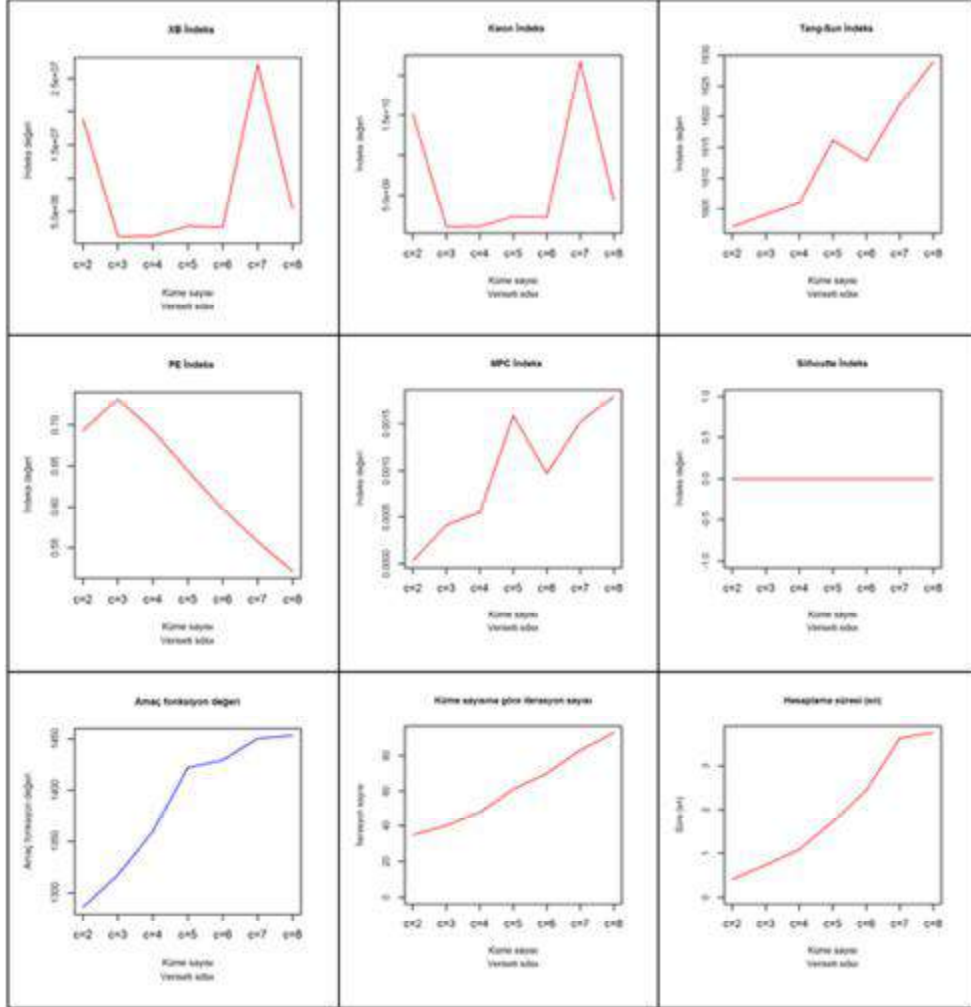
<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	3.38e+06	2.71e+09	2874.951	17.336	5.658
c=3	3.16e+05	2.54e+08	3937.268	17.587	5.419
c=4	4.28e+05	3.43e+08	5262.686	17.323	5.266
c=5	1.09e+06	8.75e+08	6207.122	16.429	4.967
c=6	1.14e+06	9.20e+08	6888.298	15.976	4.818
c=7	1.29e+07	1.04e+10	7774.510	15.385	4.633
c=8	2.88e+06	2.31e+09	8654.990	14.913	4.487

Çizelge 4.45’te uygulanan algoritmanın veri seti üzerindeki başarımları gösterilmektedir. Burada en çok dikkat çeken durumun, küme sayısının artışıyla

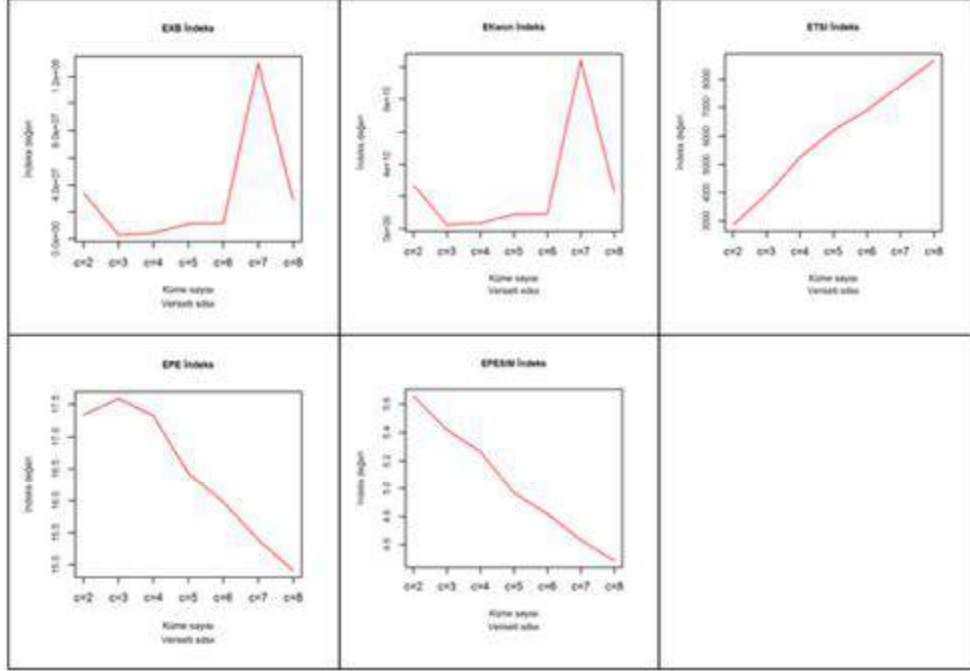
birlikte tüm basarım değerlerinin artış göstermesidir. Ek olarak küme sayısı 2 ($c=2$) olduğu durumda minimum hesaplama süresi ve iterasyon sayısı gözlenmektedir.

Çizelge 4.45. SDSX veri seti için OCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	35	0.410	1286.413
$c=3$	40	0.730	1317.492
$c=4$	48	1.080	1359.490
$c=5$	61	1.710	1422.186
$c=6$	70	2.450	1429.609
$c=7$	83	3.640	1450.271
$c=8$	93	3.760	1453.474



Şekil 4.25. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX)



Şekil 4.26. OCO algoritması uzatılmış indeks değerleri (SDSX)

4.2.1.8. Sentetik Veri Seti 8 (SDSXXN)

Çizelge 4.25 incelendiğinde elde edilen U' değerlerinden hesaplanan genelleştirilmiş indeks değerlerine göre uygun küme sayısının indeksler tarafından ortak bir öneri sunmadıkları görülmektedir. V_{XB} , V_{Kwon} ve V_{TS} indeksleri tahminlerini 2 ($c=2$) olarak yapmaktadır ki bu tahminler gerçek küme sayısına ($c=3$) en yakın isabeti gösteren indekslerdir. Genel anlamda indekslerin veri seti üzerinde sağlam bir tahminde bulunmadığı ve aynı zamanda uygulanan algoritmanın da veri seti üzerinde basarım göstermediği açık bir şekilde görülmektedir.

Çizelge 4.46. SDSXN veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	1.48e+04	1.22e+07	1648.265	0.693	0.000	-0.578
c=3	9.03e+04	7.42e+07	1656.318	0.732	0.001	0.000
c=4	3.39e+04	2.78e+07	1655.896	0.693	0.000	0.000
c=5	1.66e+04	1.37e+07	1673.959	0.644	0.002	0.000
c=6	3.76e+05	3.09e+08	1678.213	0.598	0.002	0.000
c=7	3.56e+04	2.92e+07	1676.242	0.557	0.001	0.000
c=8	8.96e+05	7.36e+08	1686.310	0.521	0.001	0.000

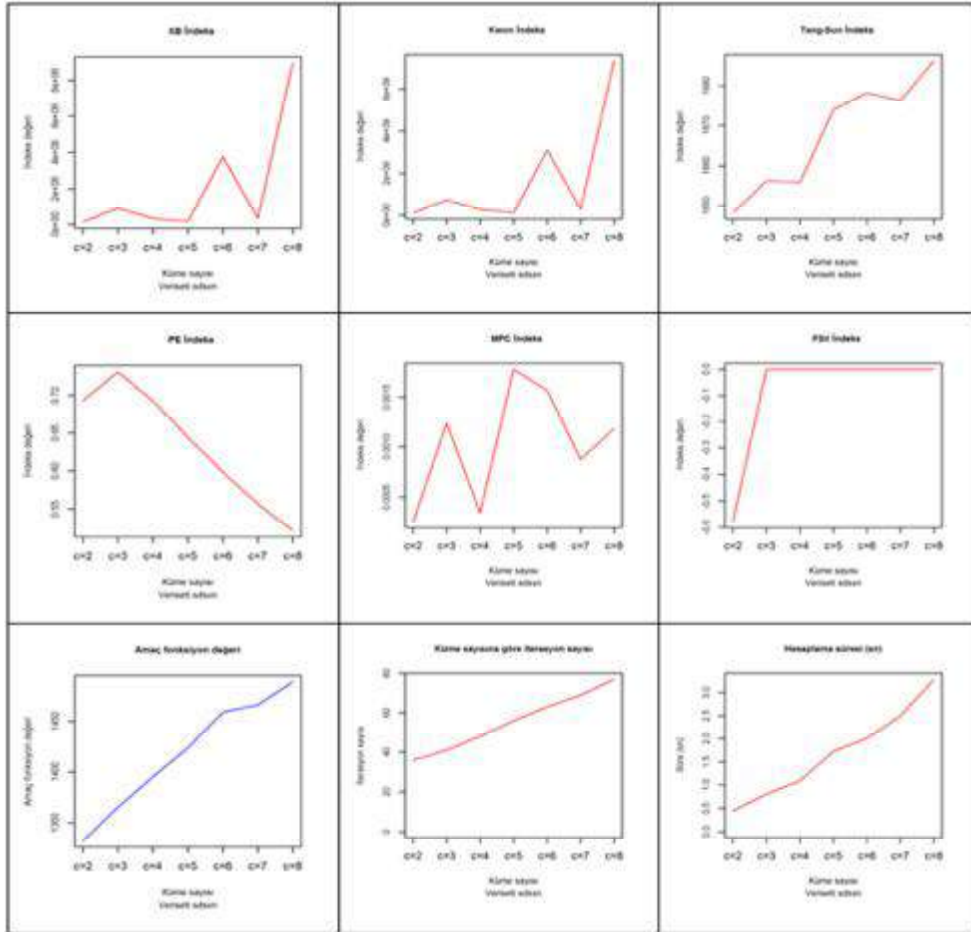
Çizelge 4.47’de elde edilen uzatılmış indeks değerleri gösterilmektedir. V_{EXB} , V_{Kwon} ve V_{TS} indeksleri tahminlerini 2 (c=2) olaraktan yapmaktadır. V_{EPE} ve V_{EPESIM} tahminlerini yine iyi yapamamaktadır ki hiç bir indeksin isabetli tahminde bulunamaması uygulamadaki genel başarısızlığı göstermektedir.

Çizelge 4.47. SDSXN veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

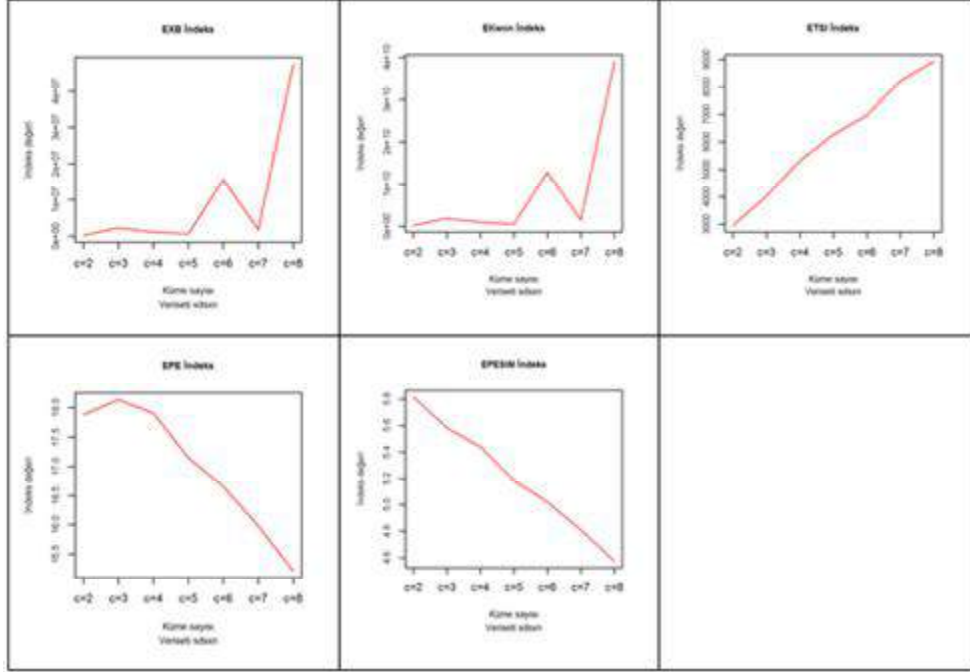
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	2.67e+04	2.20e+07	2965.686	17.883	5.816
c=3	2.20e+05	1.80e+08	4038.163	18.140	5.583
c=4	1.09e+05	8.96e+07	5325.522	17.906	5.441
c=5	6.24e+04	5.13e+07	6265.488	17.149	5.183
c=6	1.56e+06	1.28e+09	6969.410	16.653	5.021
c=7	1.75e+05	1.43e+08	8238.495	15.977	4.811
c=8	4.74e+06	3.90e+09	8930.191	15.212	4.577

Çizelge 4.48. SDSXN veri seti için OCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	36	0.450	1332.444
c=3	41	0.810	1365.598
c=4	48	1.090	1395.309
c=5	56	1.720	1423.357
c=6	63	2.000	1459.317
c=7	69	2.500	1466.533
c=8	77	3.290	1489.387



Şekil 4.27. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSXN)



Şekil 4.28. OCO algoritması uzatılmış indeks değerleri (SDSXN)

4.2.2. Gerçek Veri Setleri

4.2.2.1. Gerçek Veri Seti 1 (RDS1-Iris)

RDS1-Iris veri setinde ikisi birbiriyle çakışan 3 doğal küme bulunduğundan herhangi bir kümeleme algoritmasının 2-3 küme bulması başarılı bir kümeleme olduğuna işaret edecektir. Çizelge 4.49'dan tüm indeks değerlerine göre küme sayısının 2 olduğu önerilmiştir. Bu durum OCO algoritmasının başarılı olduğunu göstermektedir.

Çizelge 4.49. RDS1-Iris veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.006e+01	0.10e+01	10.275	0.280	0.680	0.847
c=3	1.85e+04	2.89e+06	265.570	0.690	0.350	0.000
c=4	7.35e+04	1.15e+07	263.080	0.970	0.270	0.000
c=5	7.88e+05	1.23e+08	263.510	1.000	0.240	0.000
c=6	1.78e+06	2.80e+08	268.810	1.000	0.220	0.000
c=7	3.03e+08	4.92e+10	269.930	1.000	0.130	0.000

Çizelge 4.50'deki uzatılmış indeks değerlerine göre küme sayıları 2 olarak saptanmıştır. Diğer analizlerde olduğu gibi V_{EPESIM} indeksi kümeleri bulmada yetersiz kalmaktadır.

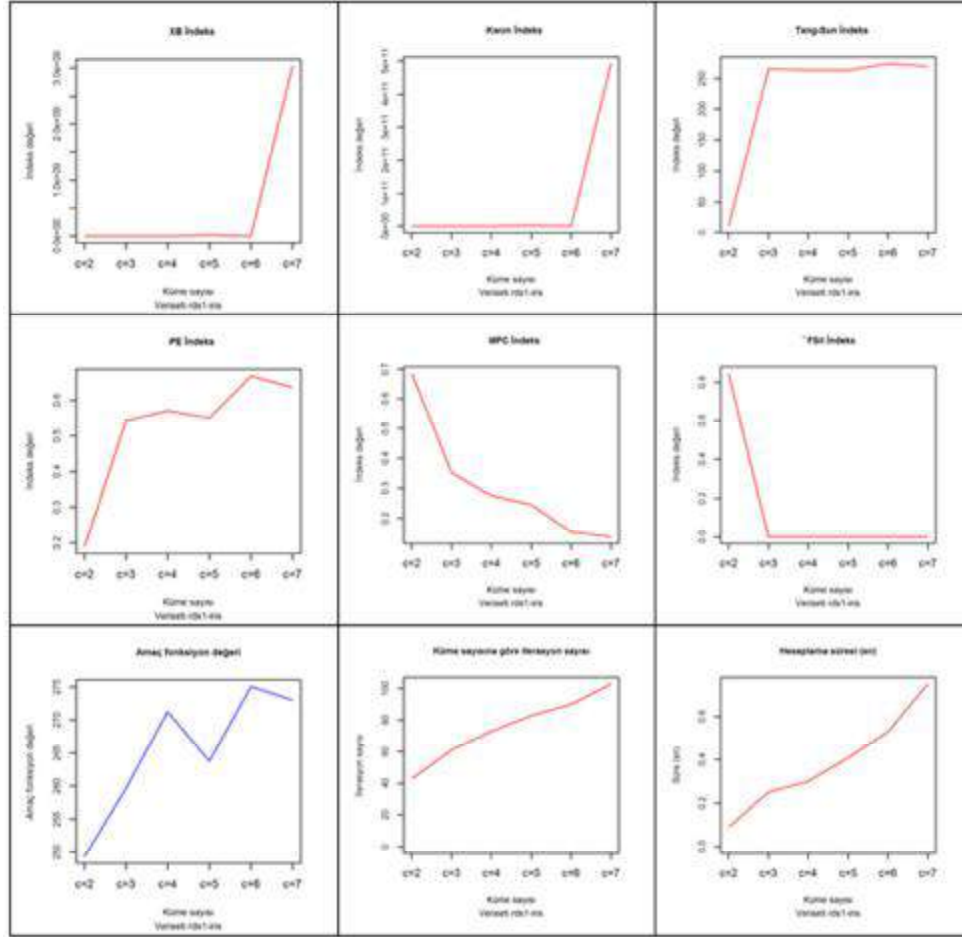
Çizelge 4.50. RDS1-Iris veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.008e+01	0.12e+01	12.362	3.156	1.003
c=3	3.24e+04	4.98e+06	453.677	3.314	1.036
c=4	1.58e+05	2.42e+07	551.096	3.384	1.043
c=5	2.05e+06	3.13e+08	663.783	3.393	1.037
c=6	5.24e+06	7.99e+08	762.205	3.413	1.038
c=7	9.98e+08	1.53e+11	834.054	3.186	0.964

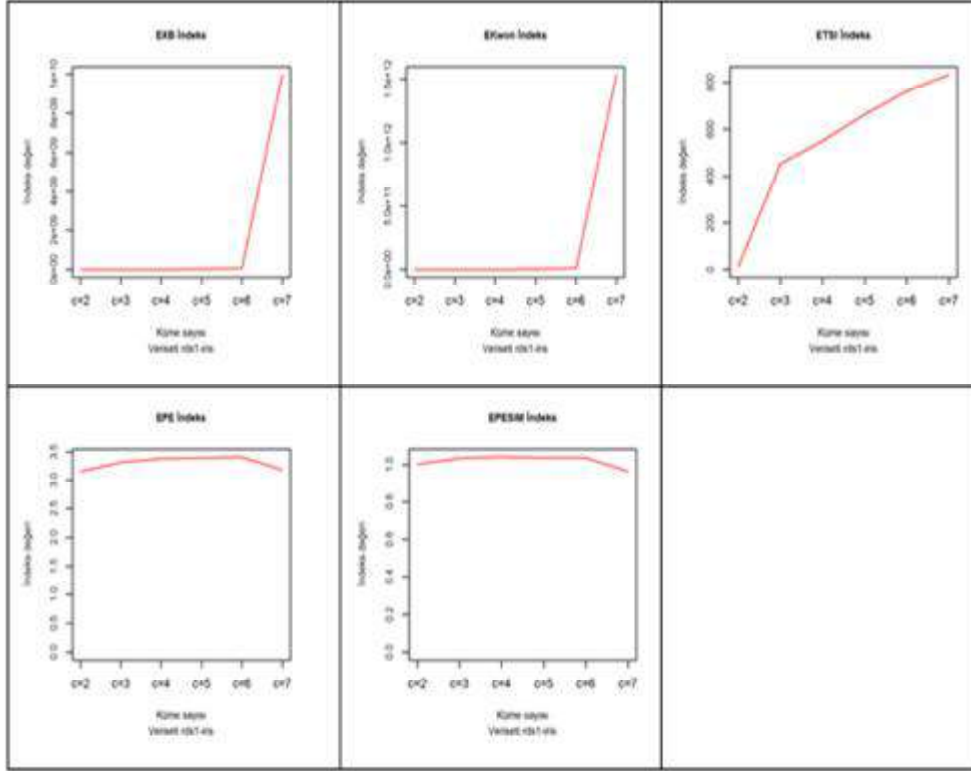
OCO algoritmasının başarımlarını gösteren Çizelge 4.51'de iterasyon sayısı ve hesaplama süreleri minimum değerleri indeks değerlerinde olduğu gibi isabetli bir tahminle ortaya koymaktadır.

Çizelge 4.51. RDS1-Iris veri seti için OCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	43	0.170	173.542
c=3	61	0.350	170.268
c=4	73	0.520	189.348
c=5	83	0.780	206.292
c=6	91	1.050	230.600
c=7	103	1.390	222.549



Şekil 4.29. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail)



Şekil 4.30. OCO algoritması uzatılmış indeks değerleri (RDS1-Iris)

4.2.2.2. Gerçek Veri Seti 2 (RDS2-Quail)

Çizelge 4.52 incelendiğinde hesaplanan genelleştirilmiş bulanık indeks değerlerine göre uygun küme sayısının V_{Fsil} indeksi dışında 3 ($c=3$) olarak önerildiği görülmektedir.

Çizelge 4.52. RDS2-Quail veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	2.95e+03	2.65e+05	1.11e+04	0.687	0.008	0.000
c=3	1.41e+01	1.30e+03	1.09e+03	0.574	0.287	-0.014
c=4	6.04e+01	5.58e+03	3.00e+03	0.591	0.238	0.012
c=5	1.64e+04	1.55e+06	6.92e+03	0.671	0.206	0.000
c=6	3.53e+03	3.39e+05	5.72e+03	0.660	0.174	0.000

Çizelge 4.53'de hesaplanan uzatılmış indeks değerlerine göre uygun küme sayısının V_{EPE} ve V_{EPESIM} indeksleri dışında 3 (c=3) olarak önerildiği görülmektedir. Genel anlamda OCO kümeleme algoritmasının veri seti üzerinde iyi bir başarımla gösterdiği görülmektedir.

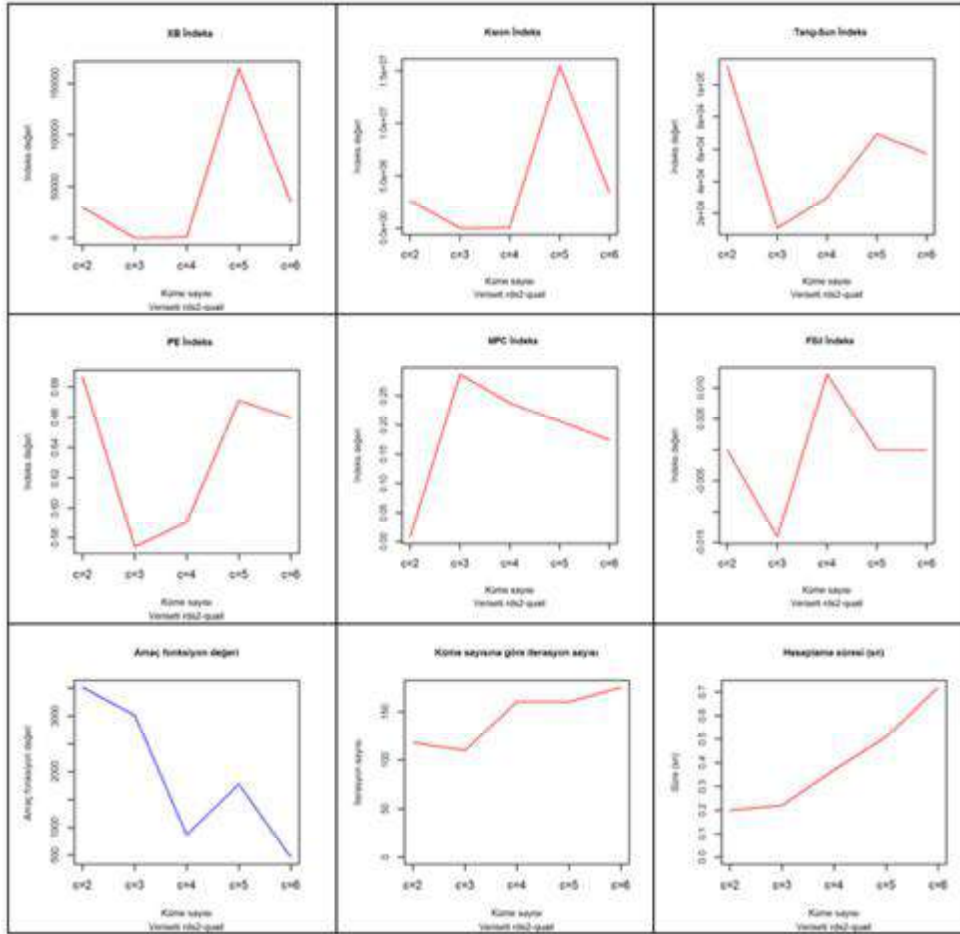
Çizelge 4.53. RDS2-Quail veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	5.60e+03	5.04e+05	2.12e+04	2.482	1.103
c=3	2.34e+01	2.14e+03	1.79e+03	2.258	0.723
c=4	1.31e+02	1.20e+04	6.45e+03	2.130	0.666
c=5	3.45e+04	3.18e+06	1.41e+04	1.879	0.574
c=6	1.02e+04	9.43e+05	1.57e+04	1.809	0.550

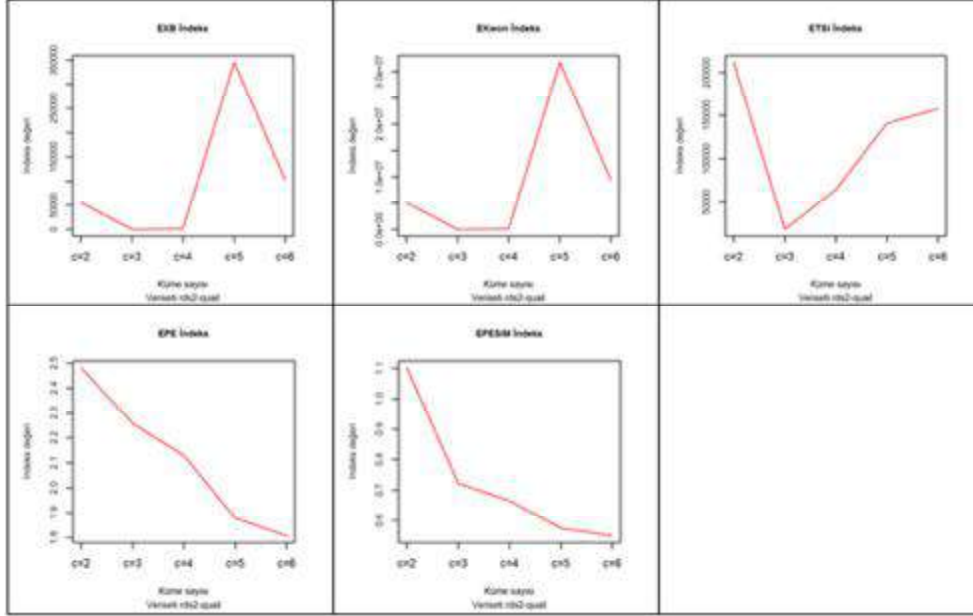
Çizelge 4.54'de veri setinin hesaplama süresinin küme sayısı 2 (c=2)'de gerçek küme sayısına (c=3) göre biraz daha hızlı surede hesaplama yaptığı gözlenmektedir.

Çizelge 4.54. RDS2-Quail veri seti için OCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	118	0.200	3512.309
$c=3$	110	0.220	3015.278
$c=4$	160	0.370	870.668
$c=5$	160	0.510	1777.673
$c=6$	175	0.720	466.375



Şekil 4.31. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail)



Şekil 4.32. OCO algoritması uzatılmış indeks değerleri (RDS2-Quail)

4.2.2.3. Gerçek Veri Seti 3 (RDS3-Wine)

Çizelge 4.55 incelendiğinde elde edilen geliştirilmiş indeks değerleri arasında isabetli tahminde bulunan bir indeks gözlenmemektedir.

Çizelge 4.55. RDS3-Wine veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	3.59e+01	6.39e+03	6.37e+03	0.663	0.519	0.038
c=3	1.31e+01	2.34e+03	2.34e+03	0.690	0.380	0.070
c=4	3.17e+03	5.76e+05	3.81e+05	0.592	0.433	0.244
c=5	0.48e+01	8.94e+02	8.98e+02	0.630	0.392	0.241
c=6	9.44e+04	1.89e+07	5.95e+05	0.589	0.429	0.315
c=7	2.69e+03	5.47e+05	2.88e+05	0.620	0.381	0.277

Çizelge 4.55 ve Çizelge 4.56’da oluşan durum itibariyle gözleniyor ki hem geliştirilmiş hem de uzatılmış indeksler de genel tahmin 5 (c=5) yönünde

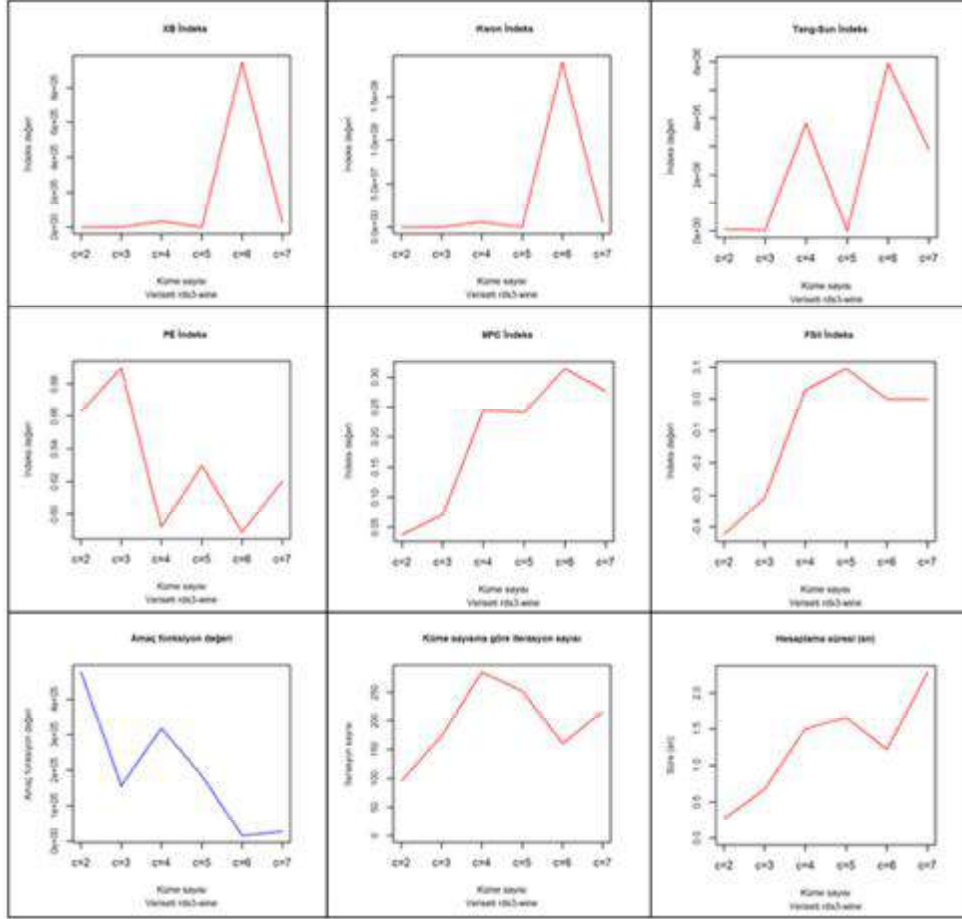
olmuştur ki bu durum uygulanan kümeleme algoritmasının veri seti üzerinde basarım göstermediği anlamına gelmektedir.

Çizelge 4.56. RDS3-Wine veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

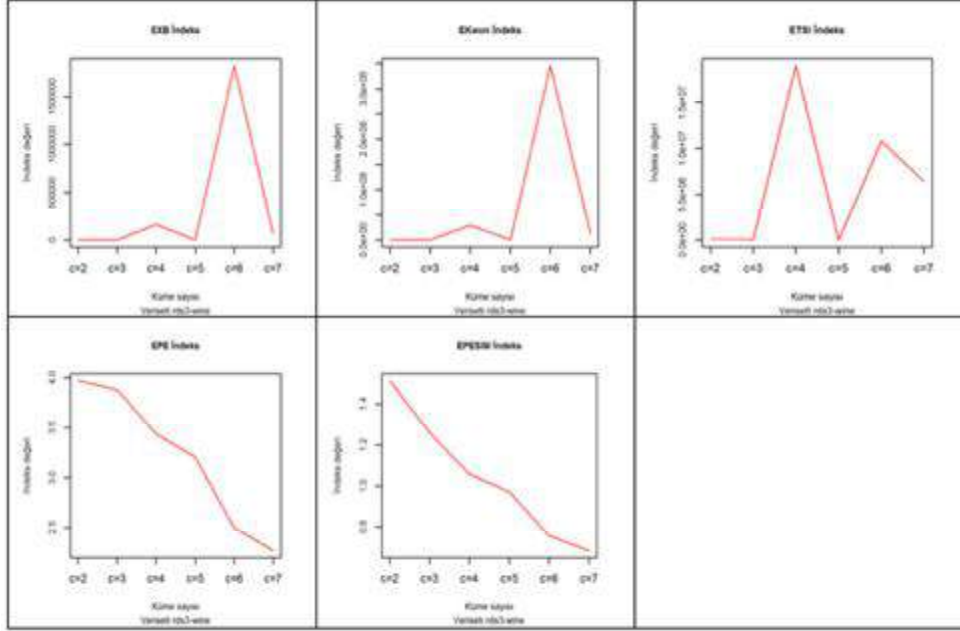
<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	6.08e+01	1.08e+04	1.07e+04	3.974	1.515
c=3	2.57e+01	4.58e+03	4.58e+03	3.884	1.262
c=4	1.61e+04	2.88e+06	1.90e+06	3.440	1.058
c=5	0.97e+01	1.76e+03	1.76e+03	3.206	0.972
c=6	1.82e+05	3.46e+07	1.07e+06	2.498	0.755
c=7	6.58e+03	1.23e+06	6.45e+05	2.268	0.684

Çizelge 4.57. RDS3-Wine veri seti için OCO algoritması başarımı

<i>C</i>	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	96	0.270	476233.747
c=3	173	0.680	157150.584
c=4	285	1.500	317442.302
c=5	252	1.660	184490.683
c=6	160	1.220	16053.511
c=7	216	2.280	28094.362



Şekil 4.33. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS3-Wine)



Şekil 4.34. OCO algoritması uzatılmış indeks değerleri (RDS3-Wine)

4.2.2.4. Gerçek Veri Seti 4 (RDS4-Glass)

Çizelge 4.58’de küme sayısına göre değişim gösteren indeks değerleri gösterilmektedir. Burada V_{Fsil} dışındaki tüm indeksler tahminlerini 2 ($c=2$) yönünde yapmaktadır. Aslında indekslerden elde edilen değerler incelendiğinde minimum değerler gerçek küme sayısına ($c=6$) isabet etmesi beklenirken oluşan tablonun tam aksi bir durum ortaya koyması, V_{Fsil} indeksinin de sağlam bir sonuç vermemesi OCO kümeleme algoritmasının veri seti üzerinde basarım gösteremediği anlamına gelmektedir.

Çizelge 4.58. RDS4-Glass veri setinde OCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.79e+01	1.69e+03	1716.540	0.640	0.102	-0.635
c=3	2.03e+01	4.34e+03	1850.430	1.000	0.056	0.000
c=4	6.42e+02	1.37e+05	2030.440	1.000	0.044	0.000
c=5	2.96e+03	6.34e+05	2071.080	1.000	0.033	0.000
c=6	6.53e+04	1.39e+07	2101.180	1.000	0.029	0.000
c=7	2.38e+03	5.11e+05	2049.010	1.000	0.022	0.000

Çizelge 4.59 incelendiğinde de küme sayısına göre değişim gösteren uzatılmış indeks değerleri V_{EPESIM} dışında tüm indeksler tahminlerini 2 (c=2) yönünde yapmaktadır.

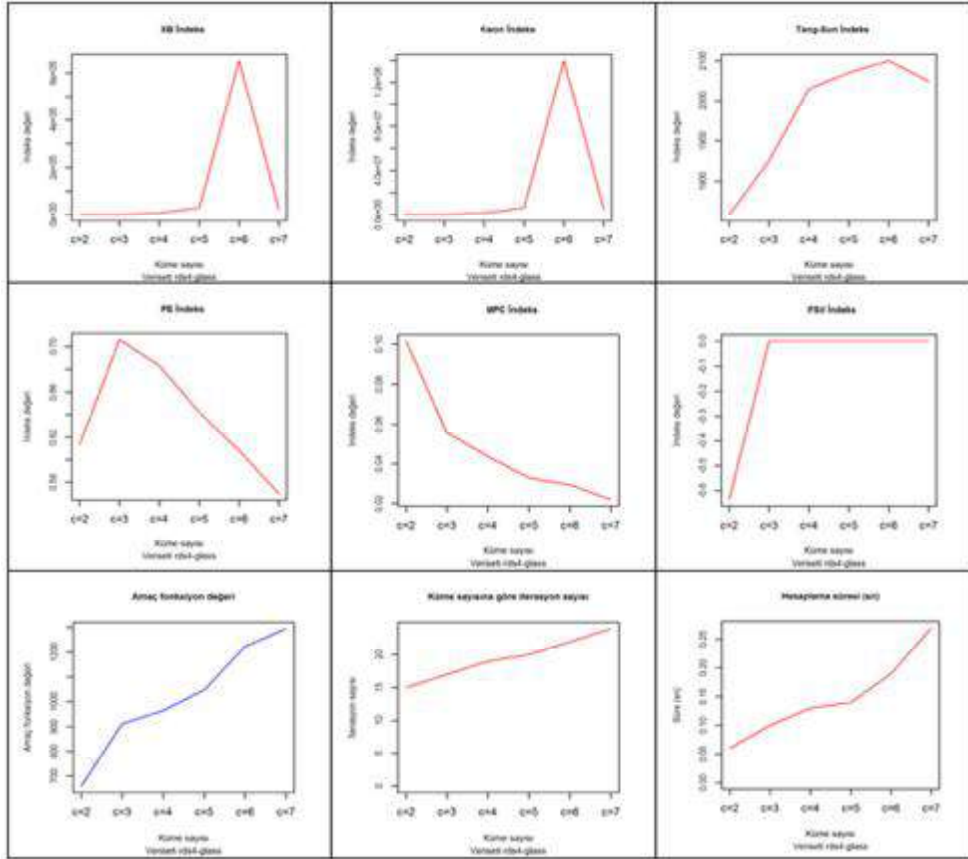
Çizelge 4.59. RDS4-Glass veri setinde OCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EFE}	V_{EPESIM}
c=2	1.00e+01	2.14e+03	2168.762	3.891	1.484
c=3	3.87e+01	8.28e+03	3526.216	3.954	1.292
c=4	1.42e+03	3.05e+05	4507.055	4.179	1.308
c=5	7.54e+03	1.61e+06	5268.531	4.319	1.331
c=6	2.08e+05	4.45e+07	6692.781	4.326	1.320
c=7	8.78e+03	1.87e+06	7531.899	4.403	1.337

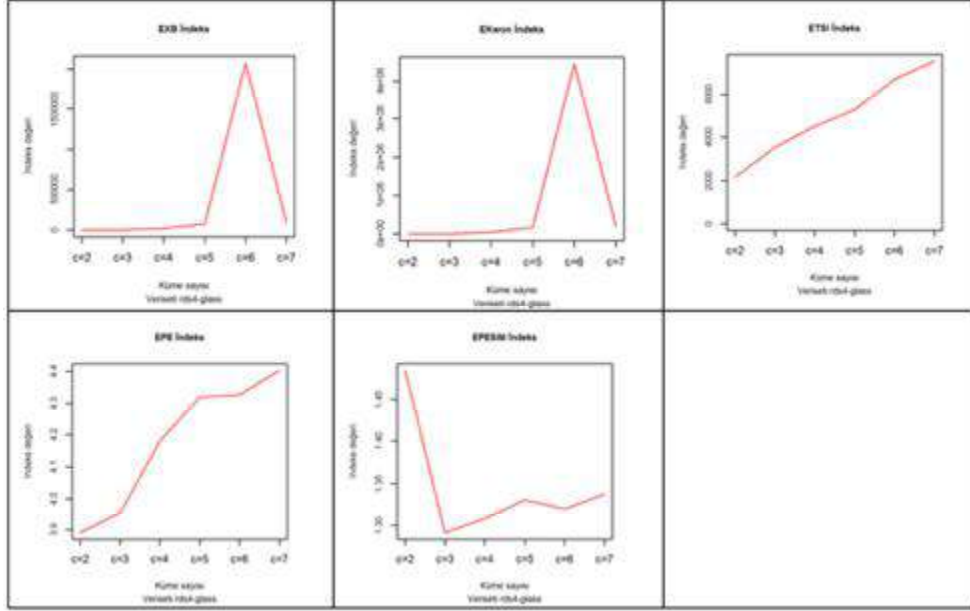
Çizelge 4.6'da OCO algoritmasının basarım değerleri gösterilmektedir. Burada iterasyon sayısı ve hesaplama sürelerinin tahminleri 2 (c=2) yönünde olmuştur.

Çizelge 4.60. RDS4-Glass veri seti için OCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	15	0.080	659.659
$c=3$	17	0.140	911.941
$c=4$	19	0.220	964.777
$c=5$	20	0.250	1046.981
$c=6$	22	0.410	1223.264
$c=7$	24	0.470	1295.156



Şekil 4.35. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS4-Glass)



Şekil 4.36. OCO algoritması uzatılmış indeks değerleri (RDS4-Glass)

4.3. BOCO Algoritması

4.3.1. Sentetik Veri Setleri

4.3.1.1. Sentetik Veri Seti 1 (SDSC)

Çizelge 4.61 incelendiğinde U ve T matrislerinin birleştirilmesi ile elde edilen U değerlerinden hesaplanan genelleştirilmiş bulanık indeks değerlerine göre optimal küme sayısının V_{PE} indeksi dışında 4 olarak önerildiği görülmektedir.

Çizelge 4.61. SDSC veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.236	188.944	166.484	0.436	0.428	0.468
c=3	0.113	90.927	83.921	0.564	0.535	0.617
c=4	0.060	48.810	45.735	0.569	0.632	0.799
c=5	0.350	283.916	202.894	0.738	0.555	0.696
c=6	0.240	194.785	151.840	0.845	0.519	0.603
c=7	0.246	201.095	154.233	0.931	0.494	0.543
c=8	0.221	180.981	142.441	1.000	0.481	0.559

Çizelge 4.62’da hem U ve hem de T değerlerini kullanarak hesaplanan uzatılmış indeks değerlerine göre genel anlamda optimal küme sayısının 4 olduğunu söylemek mümkündür. Küme sayısındaki artış ile birlikte indeks değerlerinde bir azalma eğilimi gösteren V_{EPE} ve V_{EPESIM} indeksleri tahminlerini 8 (c=8) olarak yapması küme sayısının bulunmasında etkin bir indeks olmadığını göstermektedir.

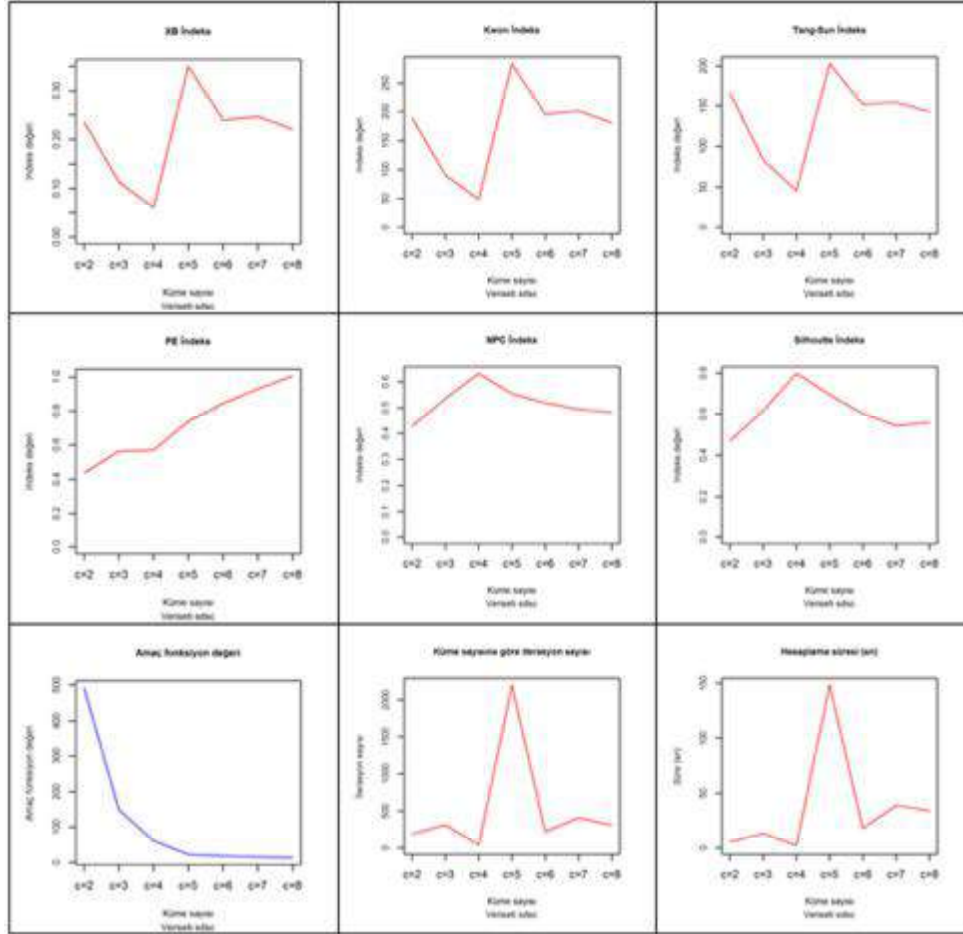
Çizelge 4.62. SDSC veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.330	264.490	232.962	11.696	3.651
c=3	0.157	125.803	116.045	7.255	2.199
c=4	0.092	73.936	69.193	4.542	1.369
c=5	0.494	398.889	284.795	4.561	1.373
c=6	0.321	259.796	202.353	4.192	1.261
c=7	0.318	258.652	198.232	3.810	1.145
c=8	0.282	229.400	180.428	3.387	1.018

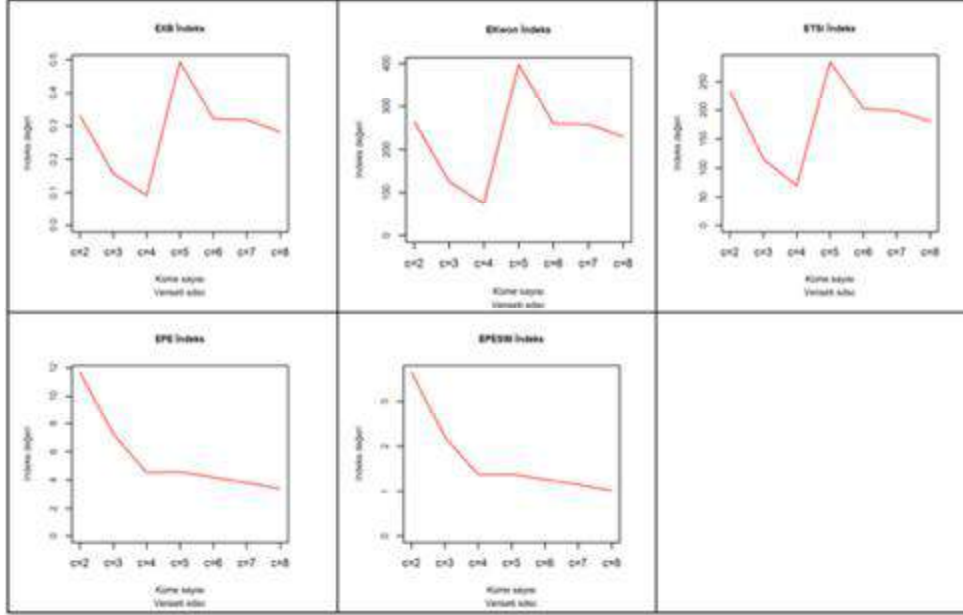
Çizelge 4.63’de uygulanan BOCO algoritmasının veri seti üzerindeki basarım sonuçları görülmektedir. Elde edilen sonuçlar incelendiğinde isabet edilen küme sayısı ($c=4$) diğer küme sayısı değerlerine göre çok daha kısa surede ve minimum iterasyon sayısı ile daha sağlam sonuçlar göstermektedir ki bu durumda uygulanan algoritmanın veri seti üzerinde iyi bir basarım gösterdiği anlaşılmaktadır.

Çizelge 4.63. SDSC veri seti için BOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	189	5.330	492.752
c=3	314	13.080	149.612
c=4	36	1.990	60.350
c=5	2200	148.710	22.056
c=6	219	17.920	17.518
c=7	406	38.810	15.354
c=8	308	33.980	13.422



Şekil 4.37. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSC)



Şekil 4.38. BOCO algoritması uzatılmış indeks değerleri (SDSC)

4.3.1.2. Sentetik Veri Seti 2 (SDSCN)

Çizelge 4.64 incelendiğinde bulanık indeks değerlerine göre optimal küme sayısı daire şekilli gürültüsüz veri setinden (SDSC) elde edilen sonuçlara benzer bir tablo göstermektedir. Aynı şekilde tahminler V_{PE} indeksi dışında 4 ($c=4$) olarak görülmektedir.

Çizelge 4.64. SDSCN veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.241	197.603	173.910	0.444	0.415	0.452
c=3	0.123	100.979	92.946	0.592	0.508	0.584
c=4	0.062	51.029	47.920	0.593	0.615	0.778
c=5	0.302	251.008	189.191	0.757	0.544	0.689
c=6	0.255	212.296	165.539	0.860	0.515	0.626
c=7	0.231	193.057	152.078	0.960	0.481	0.551
c=8	0.224	187.759	149.354	1.000	0.461	0.550

Çizelge 4.65’de hesaplanan uzatılmış indeks değerlerine göre optimal küme sayısının V_{EPE} ve V_{EPESIM} dışında 4 olduğu anlaşılmaktadır.

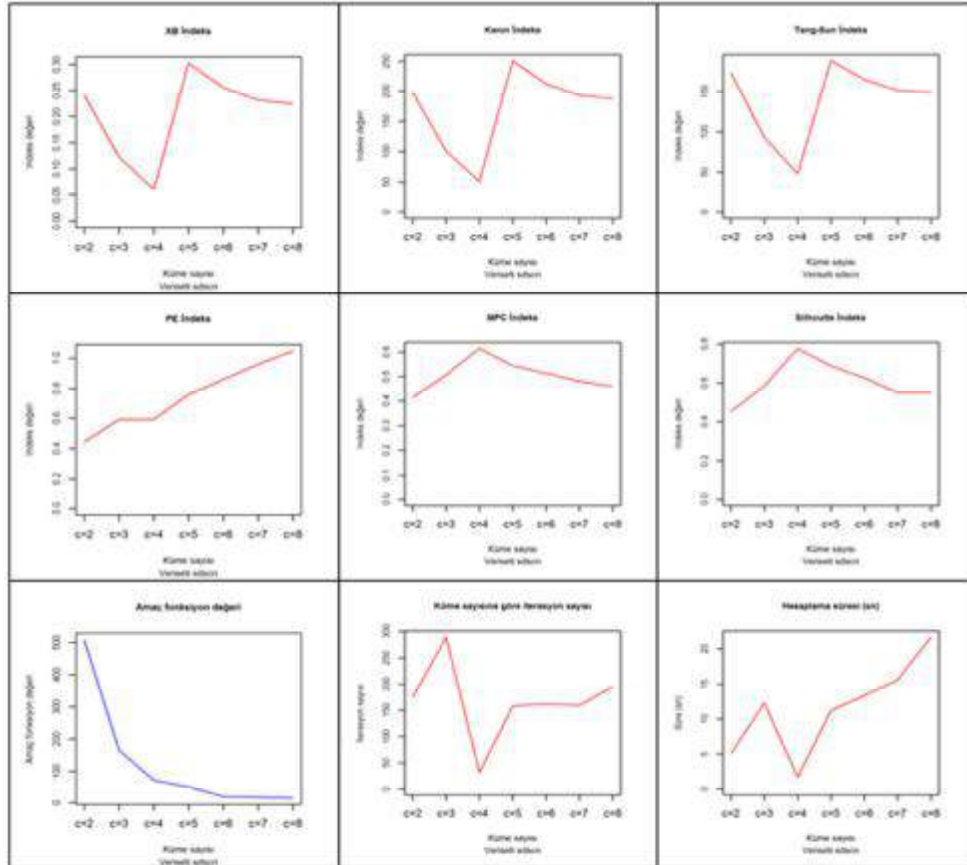
Çizelge 4.65. SDSCN veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.336	275.736	242.587	12.23	3.818
c=3	0.168	137.784	126.762	7.826	2.373
c=4	0.092	76.050	71.338	4.828	1.456
c=5	0.422	348.790	262.673	4.716	1.420
c=6	0.341	282.757	220.320	4.244	1.277
c=7	0.297	247.348	194.714	3.937	1.184
c=8	0.283	236.132	187.725	3.532	1.062

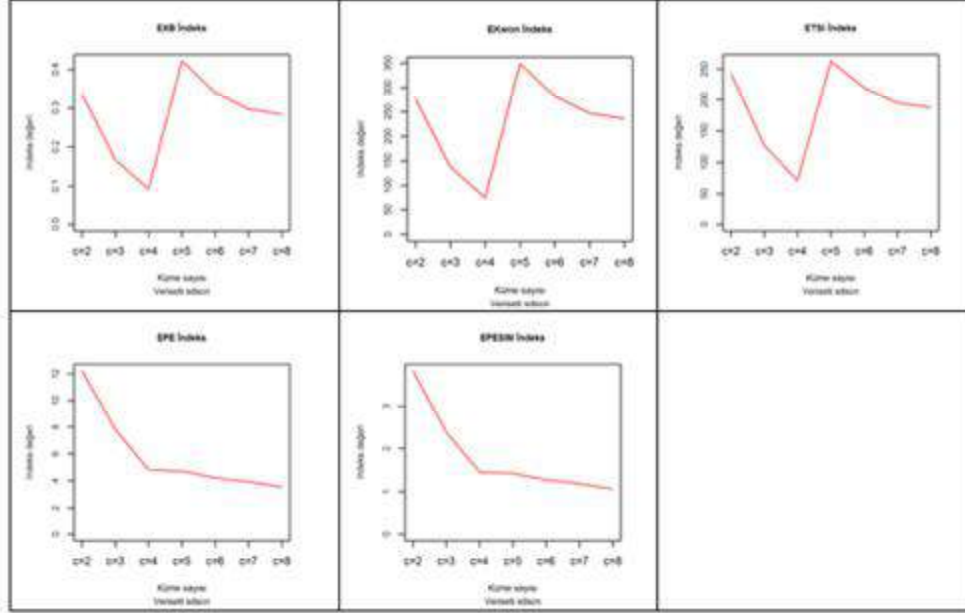
Çizelge 4.66’da hesaplanan algoritma basarım değerlerine göre optimal küme sayısının 4 olarak gösterilmektedir.

Çizelge 4.66. SDSCN veri seti için BOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	175	5.110	508.917
$c=3$	289	12.240	165.329
$c=4$	32	1.740	71.604
$c=5$	158	11.180	49.618
$c=6$	162	13.360	18.812
$c=7$	159	15.640	17.503
$c=8$	195	21.660	15.445



Şekil 4.39. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSCN)



Şekil 4.40. BOCO algoritması uzatılmış indeks değerleri (SDSCN)

4.3.1.3. Sentetik Veri Seti 3 (SDSE)

Çizelge 4.67’de elips şekilli gürültüsüz veri setinden elde edilen bulanık indeks değerlerinin optimal küme sayısının V_{PE} indeksi dışında 4 olarak önerildiği görülmektedir. Ek 3.3’te verilen küme yapılarında da veri noktalarının kümedeki ($c=4$) dağılımı gözlemlendiğinde uygulanan algoritmanın iyi bir basarım gösterdiği anlaşılmaktadır.

Çizelge 4.67. SDSE veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.227	182.214	160.889	0.432	0.438	0.517
c=3	0.135	108.689	99.457	0.625	0.479	0.596
c=4	0.082	65.901	61.447	0.685	0.539	0.685
c=5	0.191	154.007	131.052	0.844	0.486	0.631
c=6	0.144	116.700	102.425	0.930	0.476	0.609
c=7	0.123	100.551	88.937	0.968	0.486	0.607
c=8	0.105	85.766	76.426	1.000	0.493	0.631

Çizelge 4.68 ve Sekil 4.42 incelendiğinde uzatılmış indeksler arasında ilişkili sonuçlar gözlenmektedir. Burada V_{EXB} , E_{Kwon} ve V_{ETS} uzatılmış indeksleri tahminlerini 4 olarak tahmin etmektedir ki bu sonuç indekslerin veri seti üzerinde etkin bir durum ortaya koyduğunu göstermektedir. EPESIM ve EPE ise daha önceki çizelgelerde gösterilen sonuçlardan farklı bir durum ortaya koymayarak tahminlerini 8 (c=8) olarak yapmaktadır ki bu durum indekslerin veri seti için uygun indeksler olmadığını bir kez daha göstermektedir.

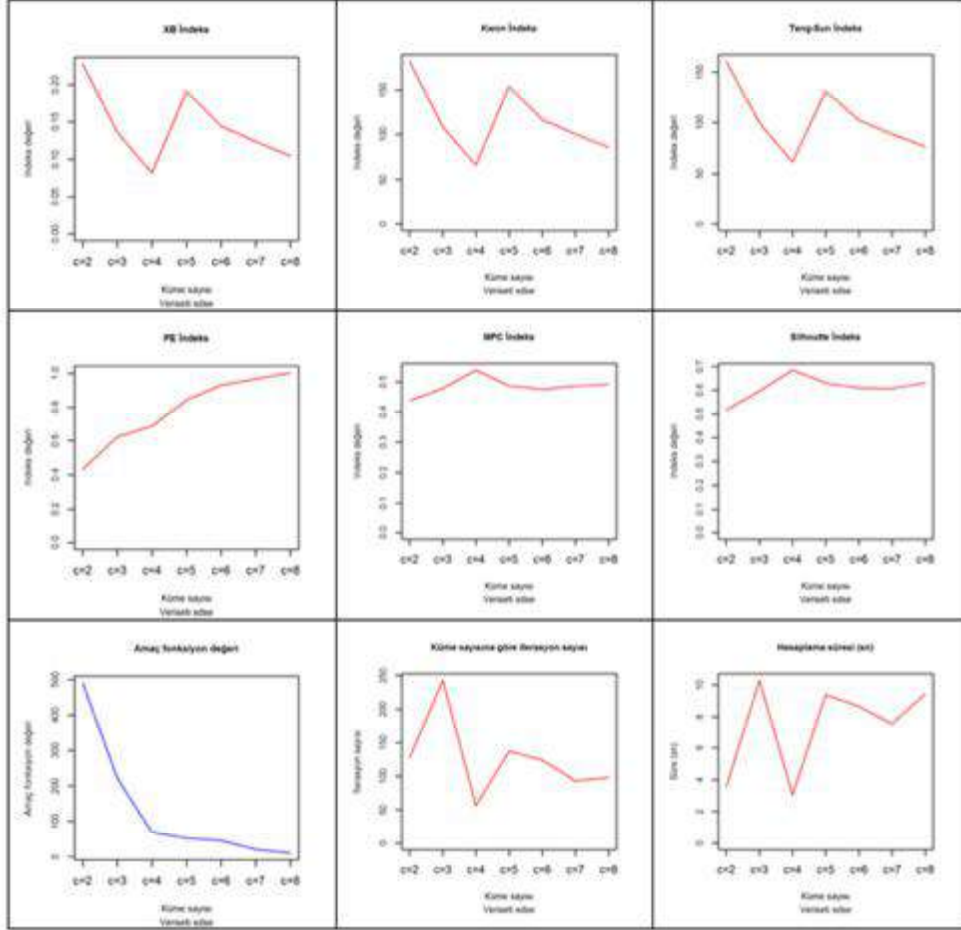
Çizelge 4.68. SDSE veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.327	261.653	230.935	11.412	3.549
c=3	0.187	149.898	137.100	8.133	2.466
c=4	0.115	92.618	86.293	5.652	1.705
c=5	0.254	204.695	174.078	5.133	1.546
c=6	0.189	152.351	133.632	4.412	1.328
c=7	0.159	128.711	113.768	3.743	1.125
c=8	0.136	110.881	98.729	3.269	0.982

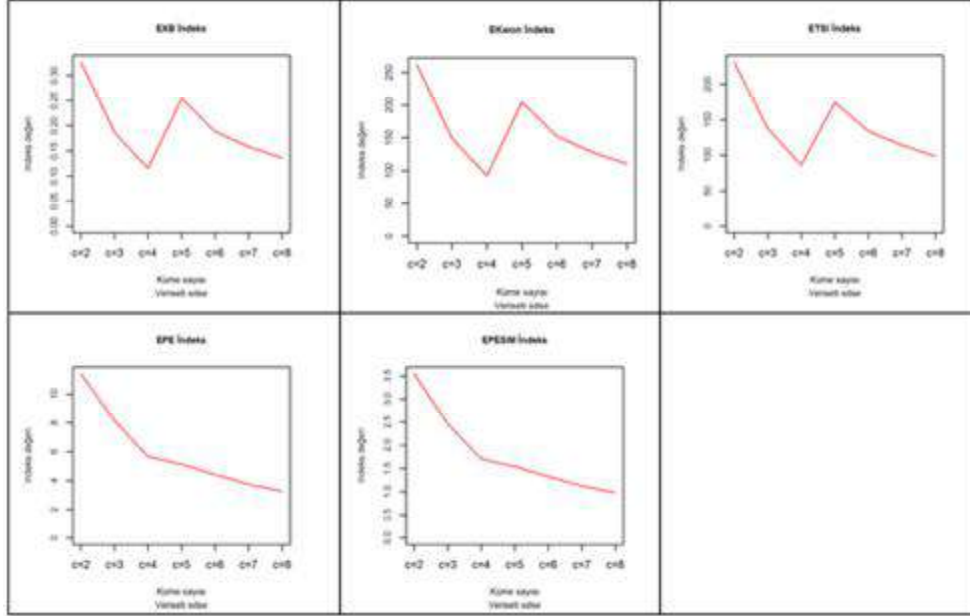
Çizelge 4.69'da BOCO kümeleme algoritmasından elde edilen başarımların sonuçları gösterilmektedir. Minimum iterasyon sayısı ve hesaplama süresi gerçek küme sayısına ($c=4$) isabet etmektedir. Şekil 4.41'de de küme sayısının artmasıyla amaç fonksiyonunda azalma eğilimi görülmektedir.

Çizelge 4.69. SDSE veri seti için BOCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	127	3.540	490.702
$c=3$	243	10.290	221.042
$c=4$	56	3.050	69.517
$c=5$	137	9.390	51.974
$c=6$	124	8.640	45.532
$c=7$	93	7.570	20.310
$c=8$	98	9.430	10.540



Şekil 4.41. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSE)



Şekil 4.42. BOCO algoritması uzatılmış indeks değerleri (SDSE)

4.3.1.4. Sentetik Veri Seti 4 (SDSEN)

Çizelge 4.70. incelendiğinde BOCO ile veri seti üzerinde hesaplanan genelleştirilmiş bulanık indeks değerlerine göre optimal küme sayısının V_{PE} indeksi dışında isabet edilen küme sayısı 4 olarak önerildiği görülmektedir.

Çizelge 4.70. SDSEN veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.227	186.342	164.711	0.434	0.432	0.493
c=3	0.123	101.231	93.151	0.594	0.510	0.617
c=4	0.068	56.552	52.983	0.625	0.588	0.739
c=5	0.196	162.599	135.098	0.752	0.551	0.680
c=6	0.153	127.574	109.161	0.834	0.538	0.676
c=7	0.127	106.001	91.894	0.896	0.527	0.649
c=8	0.246	206.006	159.133	0.993	0.498	0.619

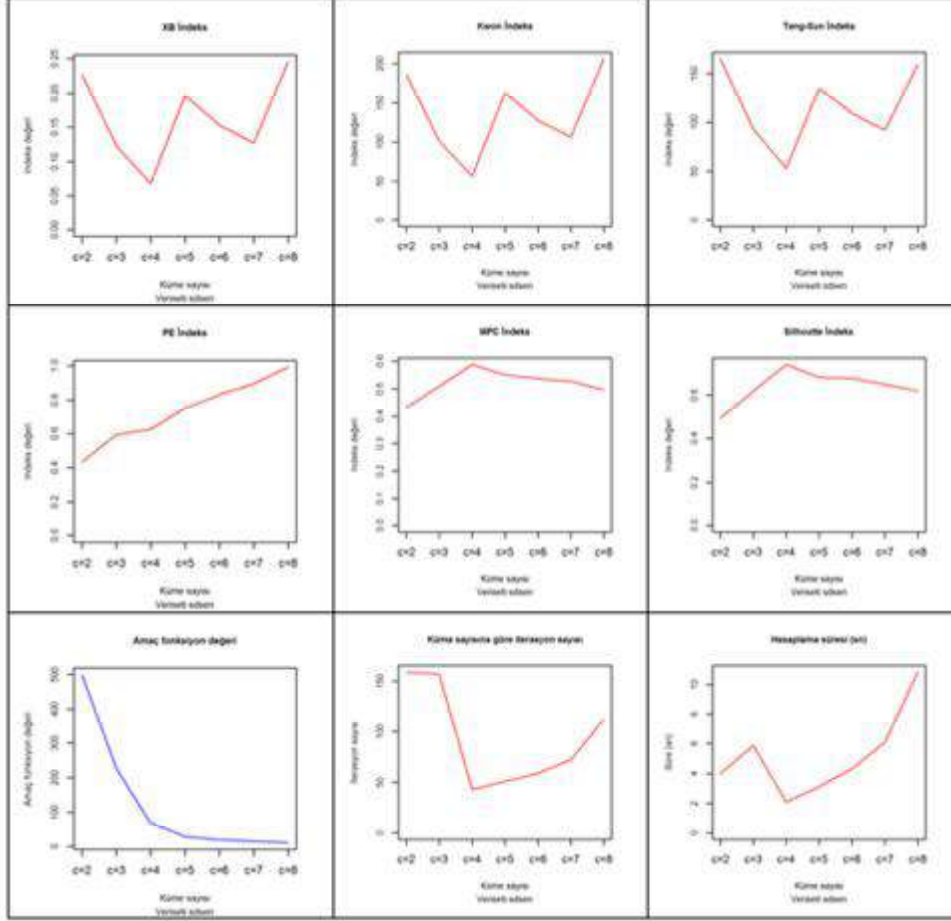
Çizelge 4.71’de hesaplanan uzatılmış indeks değerlerine göre optimal küme sayısının V_{EPE} ve V_{EPESIM} indeksleri dışında isabet edilen küme sayısının 4($c=4$) olarak önerildiği görülmektedir.

Çizelge 4.71. SDSEN veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

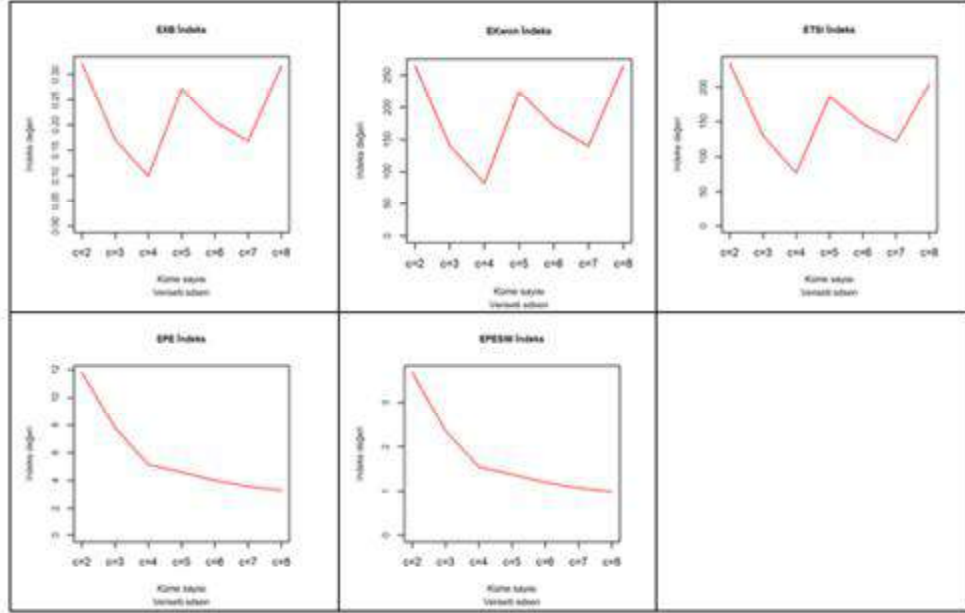
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.322	264.694	233.874	11.811	3.676
c=3	0.171	140.603	129.314	7.774	2.357
c=4	0.099	82.088	76.833	5.133	1.548
c=5	0.272	224.906	186.710	4.581	1.380
c=6	0.206	171.204	146.378	3.985	1.199
c=7	0.168	139.842	121.134	3.571	1.073
c=8	0.317	264.534	204.206	3.289	0.988

Çizelge 4.72. SDSEN veri seti için BOCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	159	3.980	499.369
c=3	157	5.870	226.823
c=4	43	2.080	67.734
c=5	51	3.120	28.347
c=6	59	4.310	19.085
c=7	72	6.100	14.964
c=8	112	10.830	11.446



Şekil 4.43. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSEN)



Şekil 4.44. BOCO algoritması uzatılmış indeks değerleri (SDSEN)

4.3.1.5. Sentetik Veri Seti 5 (SDSR)

Kare şekilli gürültüsüz veri setine uygulanan karma kümeleme algoritması (BOCO) ile elde edilen genelleştirilmiş indeks değerlerinin isabet ettiği küme sayısı 4 ($c=4$) olarak gösterilmektedir (Çizelge 4.73).

Çizelge 4.73. SDRS veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.237	189.920	167.121	0.436	0.432	0.512
c=3	0.125	100.216	92.075	0.586	0.514	0.588
c=4	0.065	52.419	49.067	0.593	0.613	0.780
c=5	0.261	211.130	164.717	0.756	0.543	0.678
c=6	0.282	228.988	174.449	0.844	0.526	0.653
c=7	0.223	181.686	143.184	0.926	0.502	0.582
c=8	0.179	146.493	118.730	0.977	0.495	0.541

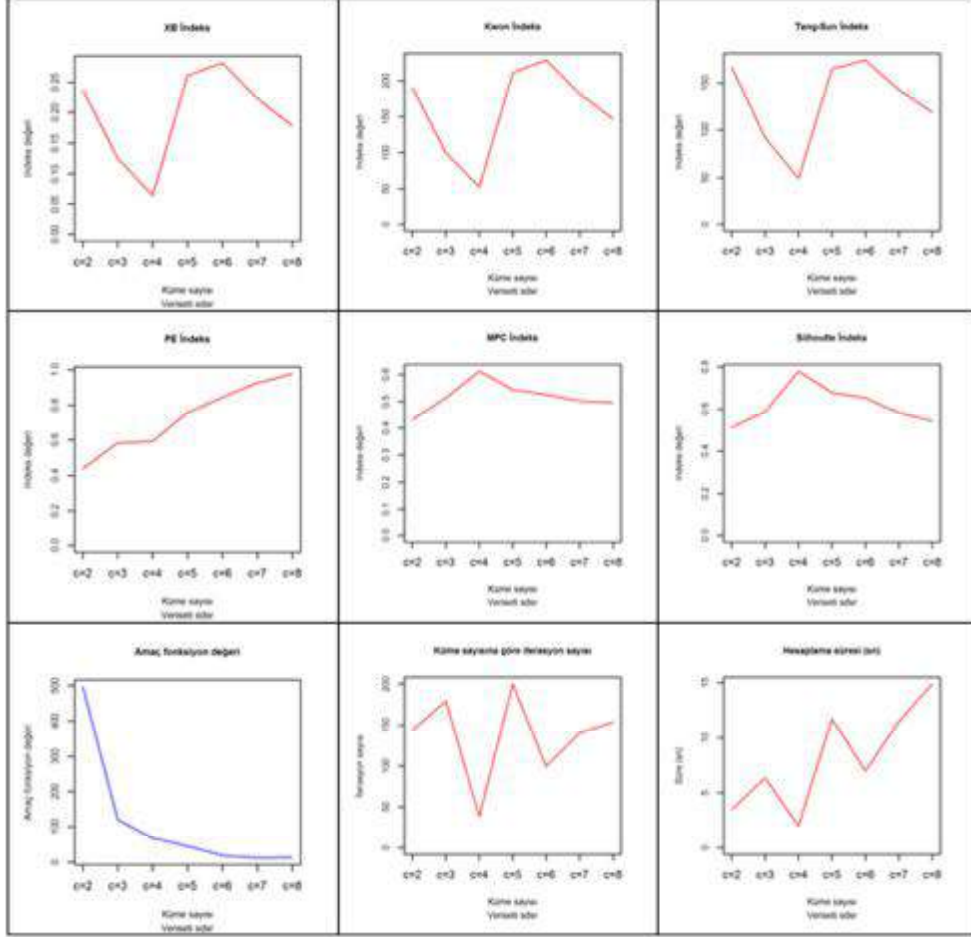
Uzatılmış indeks değerleri Çizelge 4.74’de optimal kümenin 4 ($c=4$) olduğunu ifade etse de EPESIM ve EPE indeks değerleri tutarlı bir yapı göstererek tahminleri diğer sentetik veri setlerine benzer şekilde 8 ($c=8$) olarak göstermektedir.

Çizelge 4.74. SDSR veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

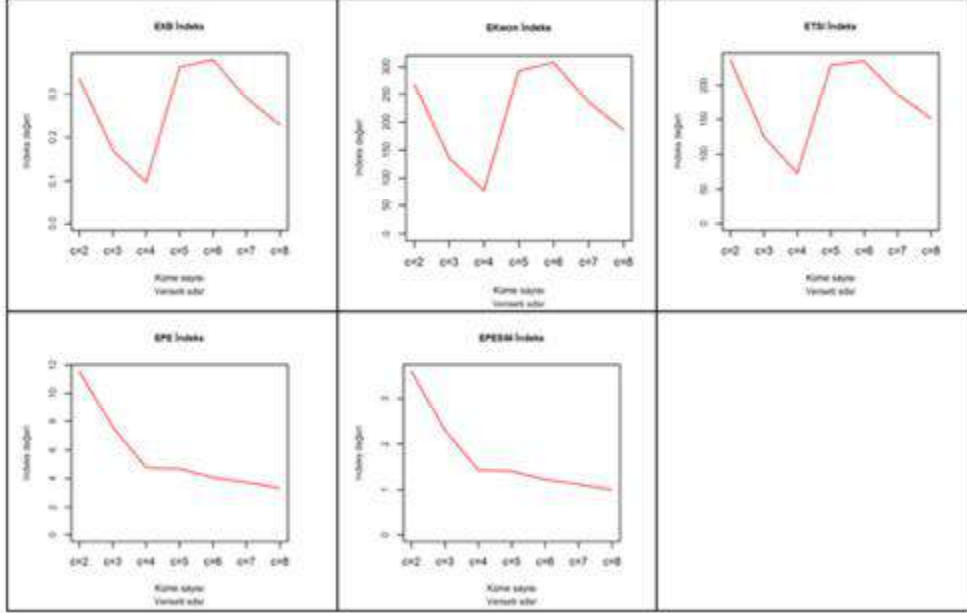
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.336	269.037	236.649	11.556	3.601
c=3	0.170	136.538	125.383	7.583	2.300
c=4	0.097	77.757	72.707	4.769	1.438
c=5	0.363	293.361	228.675	4.665	1.405
c=6	0.381	308.229	234.636	4.082	1.228
c=7	0.291	236.085	185.912	3.732	1.122
c=8	0.228	186.346	150.915	3.334	1.002

Çizelge 4.75. SDSR veri seti için BOCO algoritması başarıımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	144	3.470	496.990
c=3	179	6.370	119.572
c=4	38	2.000	67.169
c=5	200	11.740	44.398
c=6	99	7.000	18.715
c=7	140	11.500	12.932
c=8	154	14.890	13.131



Şekil 4.45. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR)



Şekil 4.46. BOCO algoritması uzatılmış indeks değerleri (SDSR)

4.3.1.6. Sentetik Veri Seti 6 (SDSRN)

Çizelge 4.76 incelendiğinde elde edilen U değerlerinden hesaplanan genelleştirilmiş bulanık indeks değerlerine göre optimal küme sayısının V_{PE} indeksi dışında 4 olarak önerildiği görülmektedir. Ne kadar BOCO algoritması ile BCO' da karşılaşılan gürültü sorununa çözüm sağlansa da Ek 3.6 ve Ek 3.6 (Devamı)'da gözlemlendiğinde optimal ve gerçek küme sayılarının ($c=4$) oluşturduğu yapılarda gürültülere rastlanmaktadır.

Çizelge 4.76. SDSRN veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.241	198.091	174.056	0.440	0.424	0.518
c=3	0.128	105.517	96.837	0.591	0.509	0.583
c=4	0.065	53.415	50.032	0.591	0.615	0.773
c=5	0.248	205.790	162.206	0.744	0.554	0.687
c=6	0.246	204.940	161.365	0.857	0.519	0.653
c=7	0.295	245.494	187.393	0.975	0.483	0.632
c=8	0.242	203.228	158.557	1.000	0.469	0.559

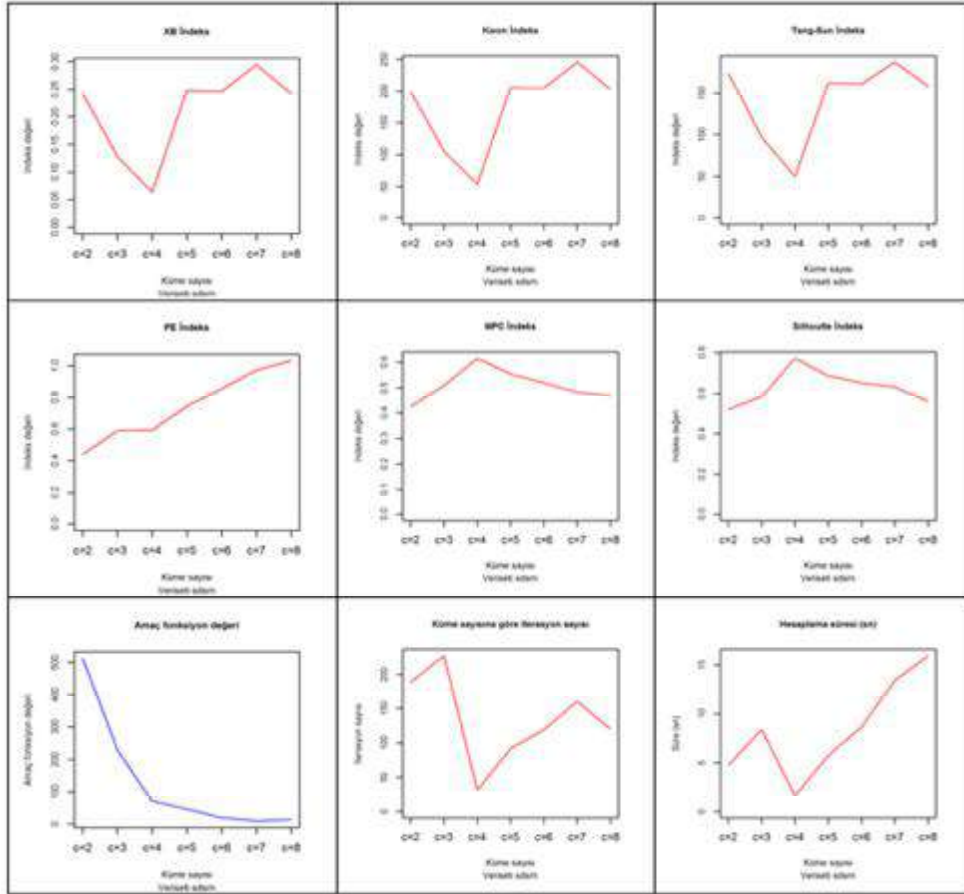
Çizelge 4.77. SDSRN veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.342	280.788	246.628	11.964	3.726
c=3	0.175	143.740	131.853	7.800	2.365
c=4	0.096	79.131	74.040	4.807	1.449
c=5	0.348	287.485	226.407	4.615	1.389
c=6	0.333	275.661	216.89	4.204	1.265
c=7	0.386	320.423	244.438	3.839	1.154
c=8	0.307	256.367	199.887	3.513	1.056

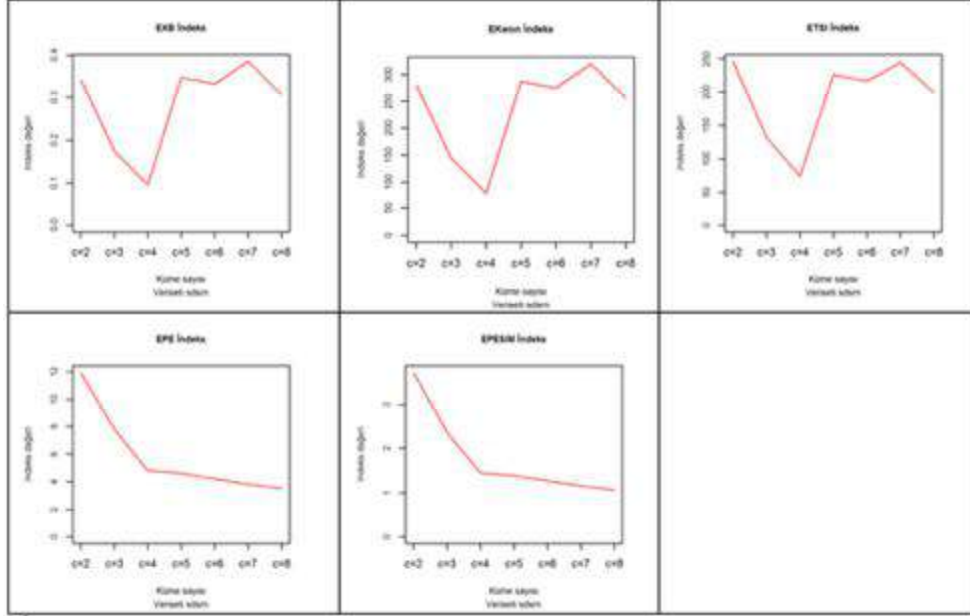
Çizelge 4.78’de iterasyon sayıları ve hesaplama süreleri incelendiğinde veri setindeki gerçek küme sayısına yakın küme sayısı seçildiğinde işlemlerin çok daha hızlı basarım gösterdiği anlaşılmaktadır.

Çizelge 4.78. SDSRN veri seti için BOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	189	4.780	512.539
c=3	227	8.380	229.486
c=4	32	1.670	69.492
c=5	92	5.730	45.492
c=6	119	8.620	18.744
c=7	160	13.410	8.987
c=8	120	15.940	12.433



Şekil 4.47. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN)



Şekil 4.48. BOCO algoritması uzatılmış indeks değerleri (SDSRN)

4.3.1.7. Sentetik Veri Seti 7 (SDSX)

Çizelge 4.79’da konkav şekilli gürültüsüz veri seti üzerine uygulanan kümeleme algoritması ile elde edilen genelleştirilmiş indeks sonuçları görülmektedir. Sonuçlar incelendiğinde hiç bir indeksin gerçek küme sayısına isabet etmediğini göstermektedir. Bu durum uygulanan kümeleme algoritmasının veri seti üzerinde iyi bir başarımlı göstermediği anlamına gelmektedir. İndekslerin genel anlamdaki tahminleri, V_{PE} dışında 5 ($c=5$) olarak gösterilmektedir.

Çizelge 4.79. SDSX veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.298	239.433	205.771	0.484	0.365	0.466
c=3	0.142	114.409	105.535	0.721	0.383	0.481
c=4	0.111	89.645	83.352	0.837	0.421	0.544
c=5	0.091	74.192	69.021	0.841	0.492	0.655
c=6	0.121	98.690	89.299	0.947	0.469	0.626
c=7	0.202	164.598	138.333	1.000	0.459	0.605
c=8	0.182	148.689	127.061	1.000	0.443	0.584

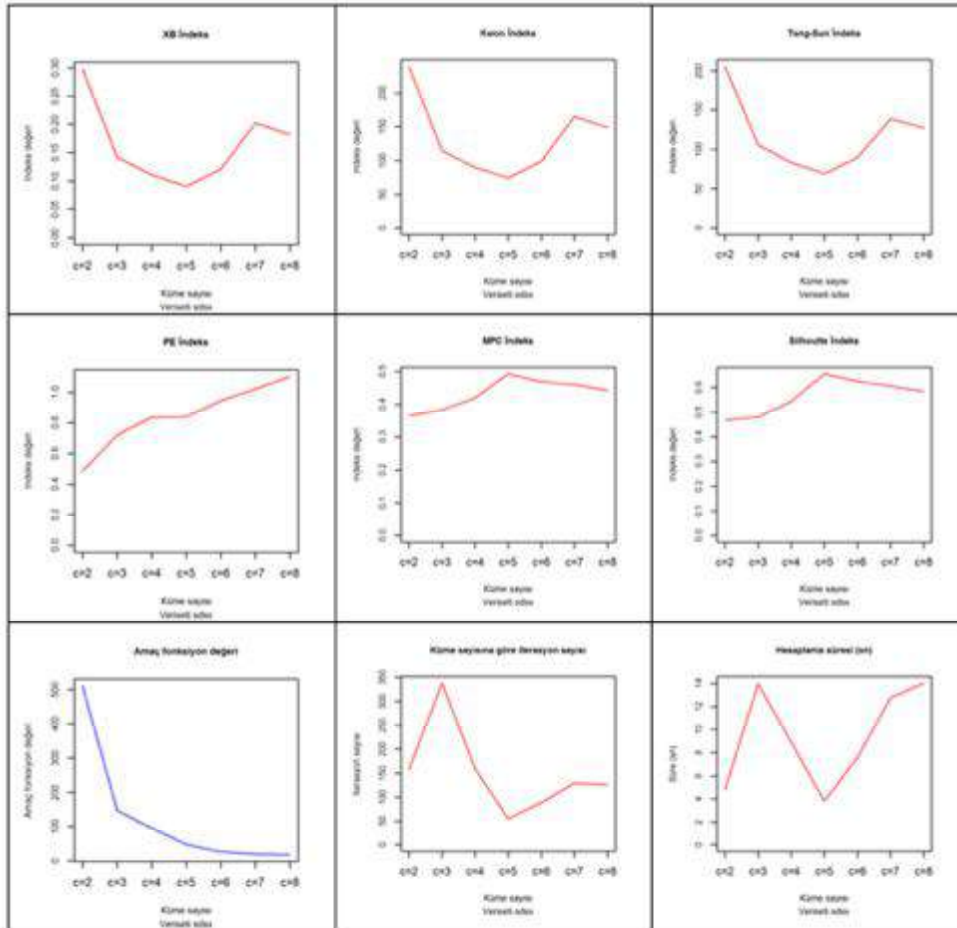
Çizelge 4.80'de minimum indeks değerleri EPE ve EPESIM uzatılmış indeksleri dışında tahminlerini genelleştirilmiş indekslerde olduğu gibi 5 (c=5) olarak göstermektedir.

Çizelge 4.80. SDSX veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

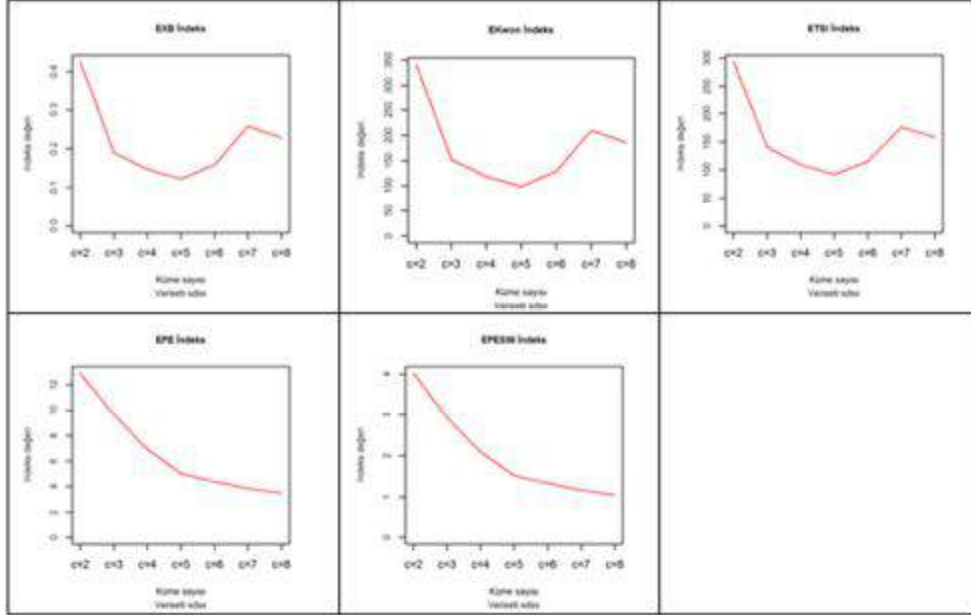
<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.425	341.148	293.095	12.929	4.028
c=3	0.190	152.669	140.775	9.631	2.927
c=4	0.147	118.393	110.024	6.935	2.096
c=5	0.123	99.089	92.119	5.033	1.516
c=6	0.158	128.112	115.845	4.407	1.326
c=7	0.258	209.535	175.994	3.864	1.162
c=8	0.227	185.310	158.268	3.473	1.044

Çizelge 4.81. SDSX veri seti için BOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	158	4.800	511.058
c=3	338	13.980	147.980
c=4	159	8.900	95.833
c=5	55	3.810	46.321
c=6	89	7.560	25.369
c=7	129	12.750	19.287
c=8	126	14.030	18.199



Şekil 4.49. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX)



Şekil 4.50. BOCO algoritması uzatılmış indeks değerleri (SDSX)

4.3.1.8. Sentetik Veri Seti 8 (SDSXN)

Çizelge 4.82’de veri setinin gürültülü bir veri seti üzerinde olmasına karşın elde edilen indeks değerleri gürültüsüz yapıya göre genel anlamda daha yakın bir tahminde bulunarak optimal küme sayısını 4 ($c=4$) olarak göstermektedir. Ancak SDSX (Çizelge 4.80)’de olduğu gibi hiç bir genelleştirilmiş bulanık indeks değerinin gerçek küme sayısına isabet edememesi uygulanan karma kümeleme algoritmasının iyi bir basarım göstermediği açıkça görülmektedir.

Çizelge 4.82. SDSXN veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	<i>V_{XB}</i>	<i>V_{Kwon}</i>	<i>V_{TS}</i>	<i>V_{PE}</i>	<i>V_{MPC}</i>	<i>V_{Fsil}</i>
c=2	0.303	249.653	214.043	0.486	0.363	0.484
c=3	0.150	123.292	113.051	0.718	0.386	0.498
c=4	0.109	90.130	83.975	0.849	0.410	0.527
c=5	0.123	101.742	92.886	0.887	0.459	0.610
c=6	0.238	198.269	165.493	0.994	0.440	0.548
c=7	0.186	155.682	133.556	1.000	0.447	0.584
c=8	0.166	138.626	120.857	1.000	0.422	0.548

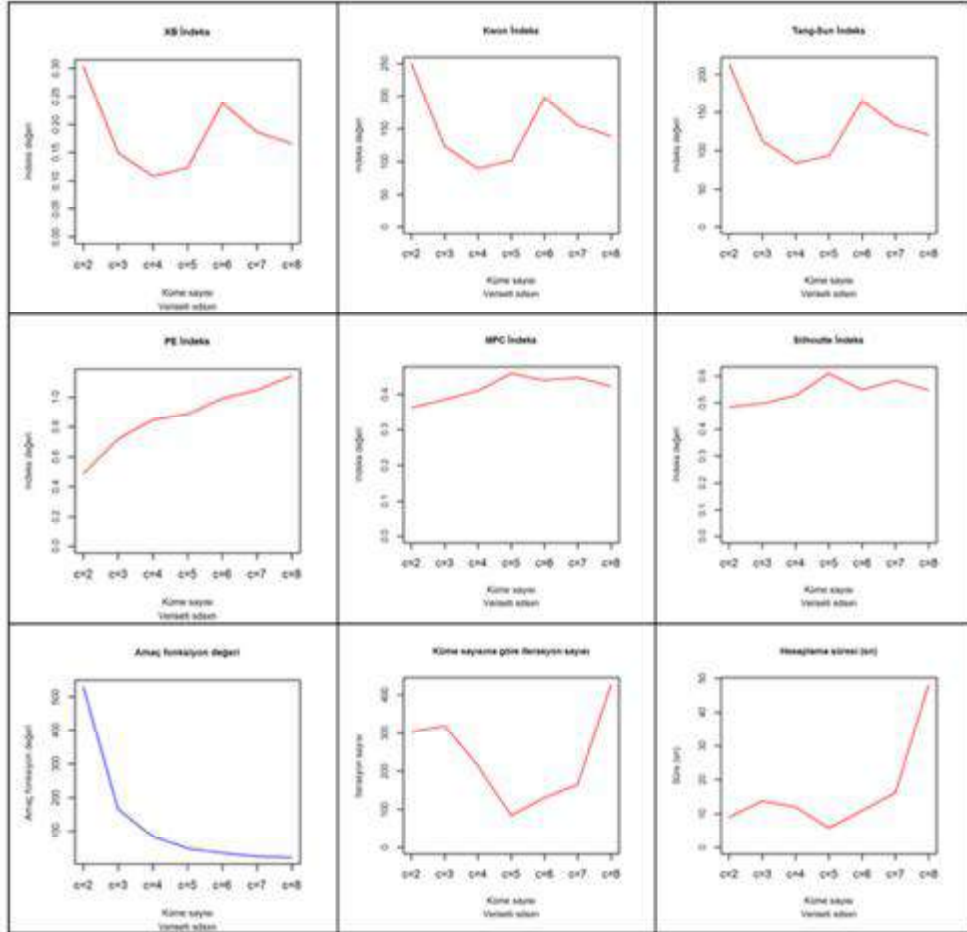
Çizelge 4.83. SDSXN veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	0.434	357.287	306.232	13.241	4.115
c=3	0.199	164.087	150.405	9.784	2.972
c=4	0.143	118.312	110.180	7.255	2.192
c=5	0.160	132.600	120.990	5.439	1.639
c=6	0.300	249.135	207.850	4.695	1.413
c=7	0.235	196.000	168.055	4.000	1.203
c=8	0.204	170.140	148.262	3.631	1.091

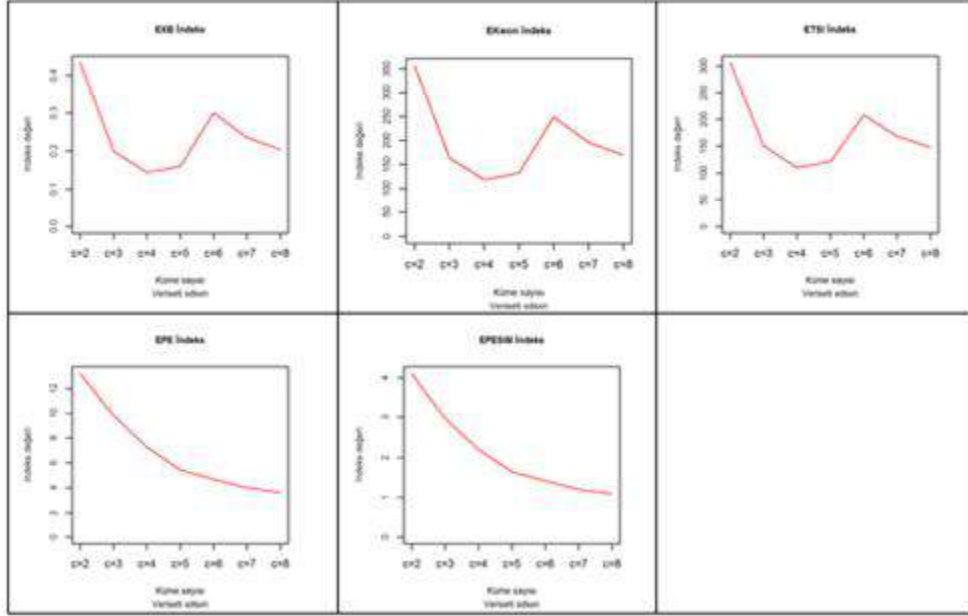
Çizelge 4.84’de iterasyon sayıları ve hesaplama süreleri incelendiğinde veri setindeki gerçek küme sayısından farklı bir tahmin olan küme sayısı (c=5), işlemlerinde daha iyi bir basarım göstermektedir.

Çizelge 4.84. SDSXN veri seti için BOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	303	8.990	529.634
c=3	319	13.820	165.187
c=4	214	12.030	83.674
c=5	84	5.850	47.768
c=6	131	11.020	34.148
c=7	165	16.200	23.212
c=8	428	48.300	21.711



Şekil 4.51. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSXN)



Şekil 4.52. BOCO algoritması uzatılmış indeks değerleri (SDSXN)

4.3.2. Gerçek Veri Setleri

4.3.2.1. Gerçek Veri Seti 1 (RDS1-Iris)

RDS1-Iris gerçek veri setine uygulanan karma kümeleme algoritması ile elde edilen bulanık indeks değerlerinin sonuçları Çizelge 4.85'de gösterilmektedir. Sonuçların isabetli tahminlerde bulunması BOCO algoritmasının başarılı olduğunu göstermektedir.

Çizelge 4.85. RDS1-Iris veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.052	8.056	8.062	0.194	0.786	0.850
c=3	0.121	19.455	18.271	0.391	0.680	0.732
c=4	0.177	29.130	25.861	0.555	0.614	0.679
c=5	0.202	34.093	29.566	0.664	0.590	0.670
c=6	0.271	48.883	39.711	0.787	0.524	0.551
c=7	0.273	50.203	40.249	0.894	0.493	0.529

Çizelge 4.86’da uzatılmış indeks değerlerine göre küme sayıları 2 olarak saptanmaktadır. Diğer analizlerde olduğu gibi V_{EPESIM} indeksi kümeleri bulmada iyi bir başarıyı gösterememektedir.

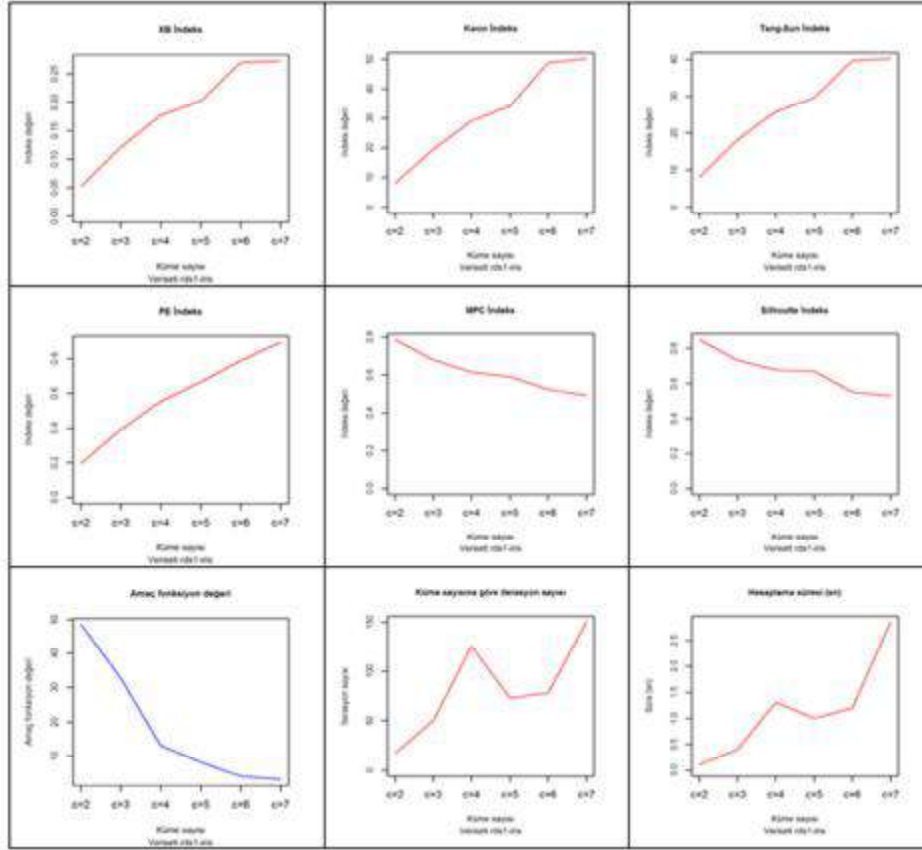
Çizelge 4.86. RDS1-Iris veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.088	13.439	13.288	1.127	0.368
c=3	0.188	29.475	27.381	1.445	0.446
c=4	0.263	42.042	37.000	1.404	0.428
c=5	0.296	48.205	41.475	1.232	0.373
c=6	0.362	62.566	50.462	1.274	0.385
c=7	0.356	62.677	49.951	1.167	0.352

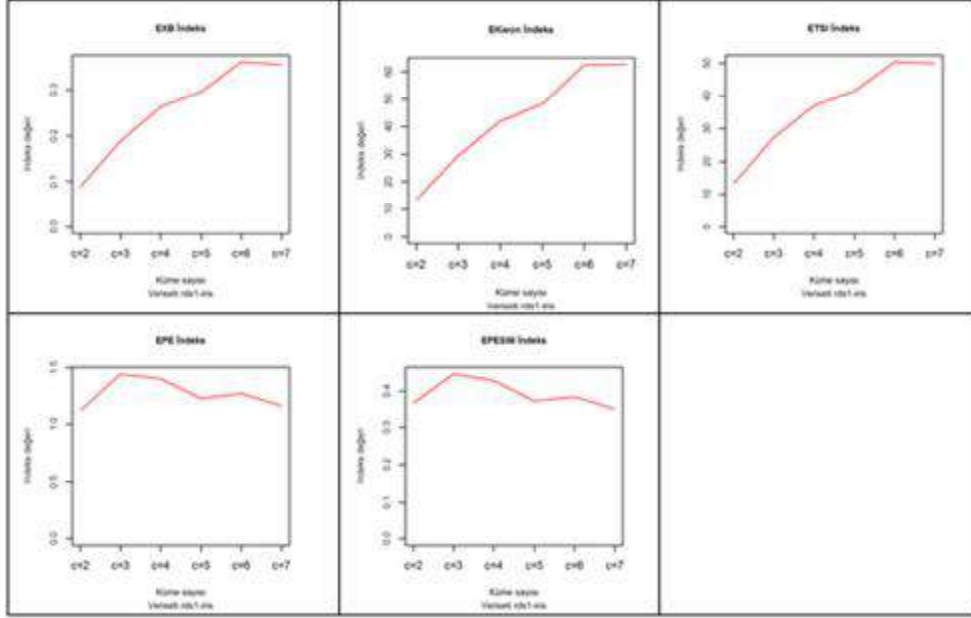
BOCO algoritmasının başarımlarını gösteren Çizelge 4.87’de iterasyon sayısı ve hesaplama süreleri minimum değerleri indeks değerlerinde olduğu gibi isabetli bir tahminle (c=2) ortaya koymaktadır.

Çizelge 4.87. RDS1-Iris veri seti için BOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	17	0.100	48.529
$c=3$	50	0.400	32.693
$c=4$	125	1.310	13.158
$c=5$	73	1.000	8.236
$c=6$	78	1.200	4.110
$c=7$	150	2.860	3.315



Şekil 4.53. OCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS1-Iris)



Şekil 4.54. BOCO algoritması uzatılmış indeks değerleri (RDS1-Iris)

4.3.2.2. Gerçek Veri Seti 2 (RDS2-Quail)

RDS2-Quail gerçek veri setinde indeksler tarafından küme sayısı 3 ($c=3$) olarak isabet etmesi beklenmektedir. Çizelge 4.88’de tüm indeks değerlerine göre küme sayısının 2 olduğu önerilmiştir. BOCO algoritmasının bu duruma çok yakın bir başarı gösterdiği yani genel anlamda uygulanan algoritmanın başarılı olduğundan bahsetmek mümkündür.

Çizelge 4.88. RDS2-Quail veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.072	6.705	6.954	0.243	0.712	0.772
c=3	0.074	7.558	7.986	0.362	0.704	0.749
c=4	0.117	12.763	13.512	0.463	0.683	0.737
c=5	0.097	11.919	12.710	0.511	0.681	0.713
c=6	0.092	13.462	14.500	0.539	0.684	0.679

Çizelge 4.89’da yalnızca V_{EXB} uzatılmış indeks değeri isabetli tahminde bulunmaktadır. Ancak V_{EKwon} ve V_{ETS} indekslerinde gerçek küme sayısına yakın tahminlerde bulunmaktadır. Genel anlamda uzatılmış indeksler üzerinden V_{EXB} dışında çok uzak olmasa da net bir başarıyı gösterdikleri söylenemez.

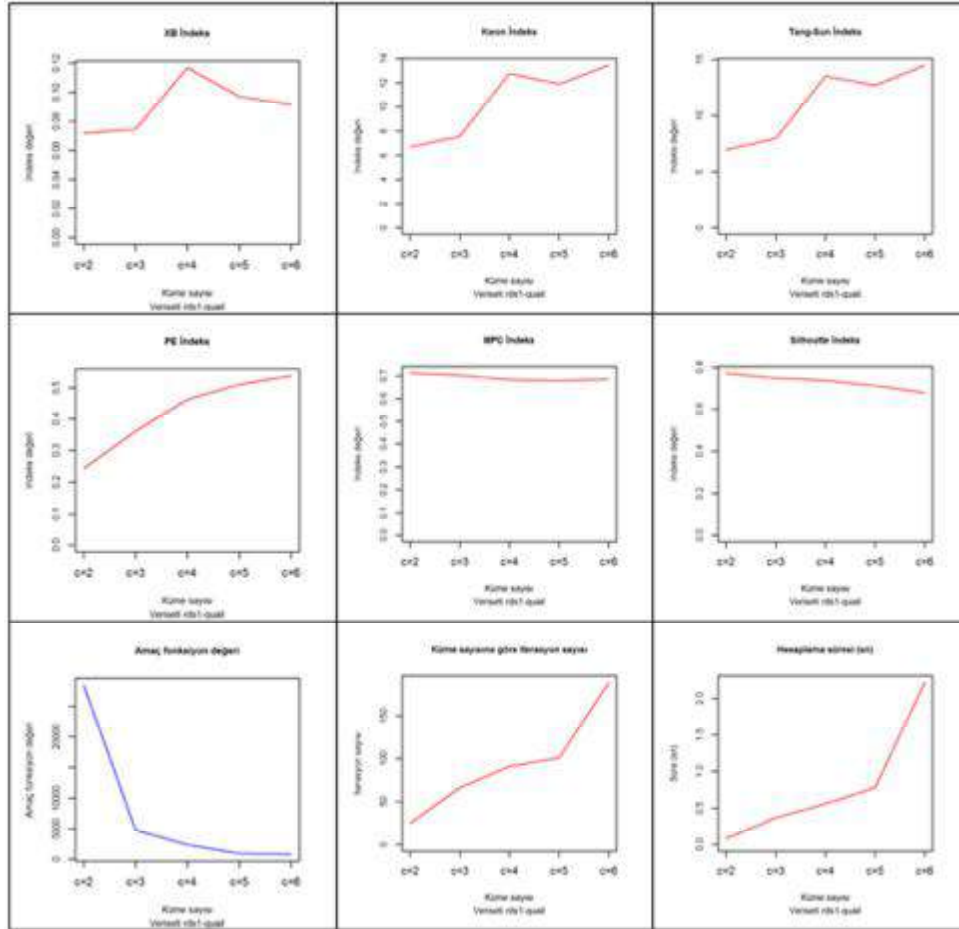
Çizelge 4.89. RDS2-Quail veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.116	10.733	10.981	1.088	0.373
c=3	0.114	11.102	11.530	0.944	0.295
c=4	0.175	18.033	18.779	0.836	0.256
c=5	0.142	15.998	16.787	0.723	0.219
c=6	0.130	16.920	17.956	0.625	0.189

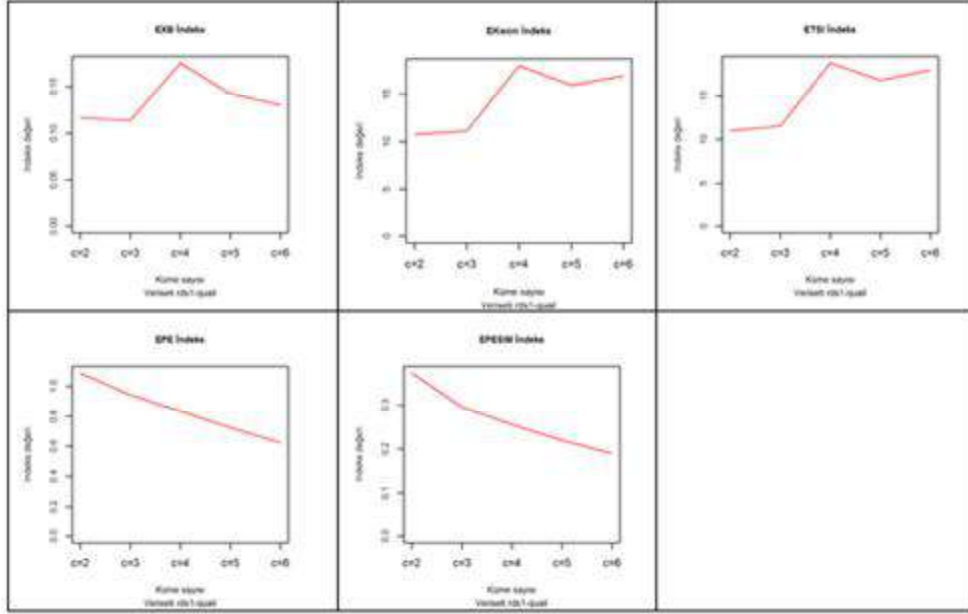
Çizelge 4.90’da BOCO algoritmasının başarımlarını gösteren iterasyon sayısı ve hesaplama süreleri minimum değerlerini 2 (c=2) olarak ortaya koymaktadır. Küme sayısının artışıyla iterasyon sayısı ve hesaplama sürelerinin artması bu iki durum arasında pozitif bir ilişki olduğunu ifade etmektedir ki Şekil 4.55’te bu durum net bir şekilde görülmektedir.

Çizelge 4.90. RDS2-Quail veri seti için BOCO algoritması başarıımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	25	0.080	28468.601
c=3	66	0.370	4756.891
c=4	91	0.560	2361.605
c=5	101	0.780	916.979
c=6	189	2.220	815.577



Şekil 4.55. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarıımı (RDS2-Quail)



Şekil 4.56. BOCO algoritması uzatılmış indeks değerleri (RDS2-Quail)

4.3.2.3. Gerçek Veri Seti 3 (RDS3-Wine)

RDS3-Wine veri setinden elde edilen tahminler RDS2-Quail'de (Çizelge 4.88) olduğu gibi benzer bir tablo oluşturmaktadır. Burada da elde edilen tüm indeks değerlerine göre küme sayısının 2 olduğu önerilmiştir. Ancak burada farklı olarak isabet edilen küme sayısına (c=2) yakın küme tahminleri (c=4 ve c=7) gerçek küme sayısına (c=3) göre daha iyi sonuçlar elde etmesi durumu söz konusudur. Yani genel anlamda algoritmanın basarimi istenilen düzeyde olmadığı gözlenmektedir.

Çizelge 4.91. RDS3-Wine veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.054	9.949	10.199	0.215	0.753	0.819
c=3	0.122	22.890	23.521	0.380	0.685	0.721
c=4	0.090	17.992	18.645	0.418	0.710	0.714
c=5	0.152	31.901	33.100	0.516	0.674	0.669
c=6	0.134	29.209	30.284	0.516	0.699	0.710
c=7	0.081	21.950	23.200	0.503	0.719	0.738

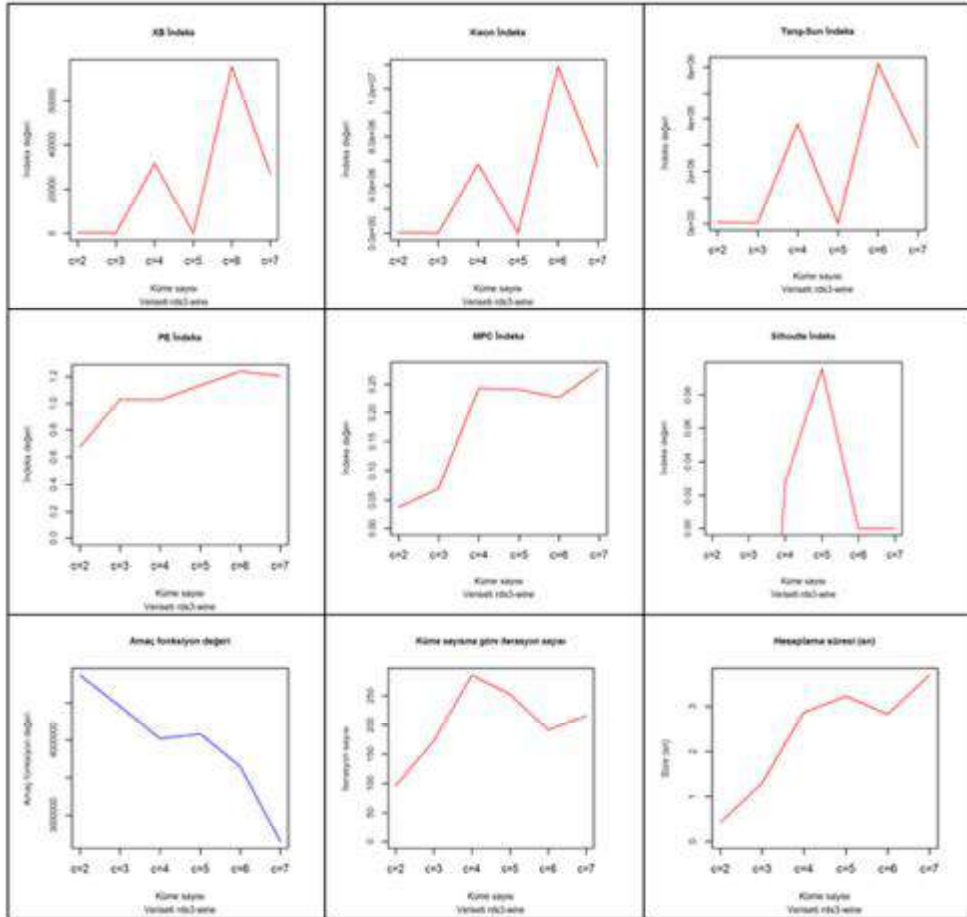
Çizelge 4.92’da yalnızca uzatılmış indeks değerleri arasında isabetli tahminde bulunmamaktadır. Ancak V_{EKwon} , V_{EXB} ve V_{ETS} indeksleri gerçek küme sayısına yakın tahminlerde bulunmaktadır. Genel anlamda uzatılmış indekslerinde genelleştirilmiş indeks değerleri gibi başarımlarını istenilen düzeyde gösterememektedir.

Çizelge 4.92. RDS3-Wine veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

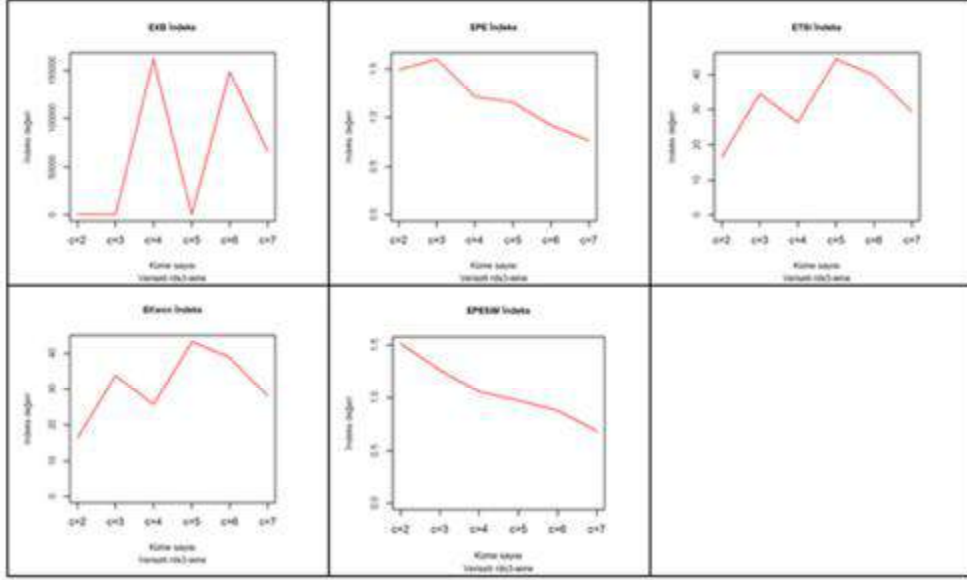
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.090	16.353	16.603	1.490	0.487
c=3	0.183	33.851	34.482	1.600	0.493
c=4	0.133	25.689	26.341	1.216	0.370
c=5	0.216	43.226	44.425	1.152	0.348
c=6	0.188	38.869	39.944	0.920	0.277
c=7	0.116	28.126	29.377	0.765	0.230

Çizelge 4.93. RDS3-Wine veri seti için BOCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	27	0.170	3508091.432
c=3	77	0.780	681929.340
c=4	58	0.800	357779.848
c=5	173	3.040	214958.235
c=6	68	1.300	107449.985
c=7	66	1.410	37261.367



Şekil 4.57. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımları (RDS3-Wine)



Şekil 4.58. BOCO algoritması uzatılmış indeks değerleri (RDS3-Wine)

4.3.2.4. Gerçek Veri Seti 4 (RDS4-Glass)

RDS4-Glass veri setinden elde edilen değerlerde (Çizelge 4.94) de görüldüğü üzere indeksler tahminlerini V_{PE} dışında 3 ($c=3$) olarak göstermektedir. Elde edilen indeks değerlerinin gerçek küme sayısından ($c=6$) uzak tahminlerde bulunması algoritmanın (BOCO) veri seti üzerinde iyi bir basarım gösteremediğini ortaya koymaktadır.

Çizelge 4.94. RDS4-Glass veri setinde BOCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.202	43.583	42.194	0.304	0.632	0.688
c=3	0.113	24.491	24.150	0.433	0.658	0.746
c=4	0.445	97.880	89.955	0.656	0.541	0.533
c=5	2.153	476.704	324.588	0.913	0.400	0.295
c=6	1.737	393.087	283.876	0.960	0.417	0.292
c=7	1.505	342.174	255.705	1.000	0.398	0.304

Çizelge 4.95'te verilen uzatılmış indeks değerlerinin de genelleştirilmiş bulanık indeks değerlerinden çok daha iyi sonuçlar verdiğiinden bahsetmek mümkün olmasa da, EPESIM ve EPE uzatılmış indeksleri küme sayısına (c=6) yakın tahminlerde bulunmaktadır. Ancak bu durum genel anlamda iyi bir başarıım anlamına gelmemektedir.

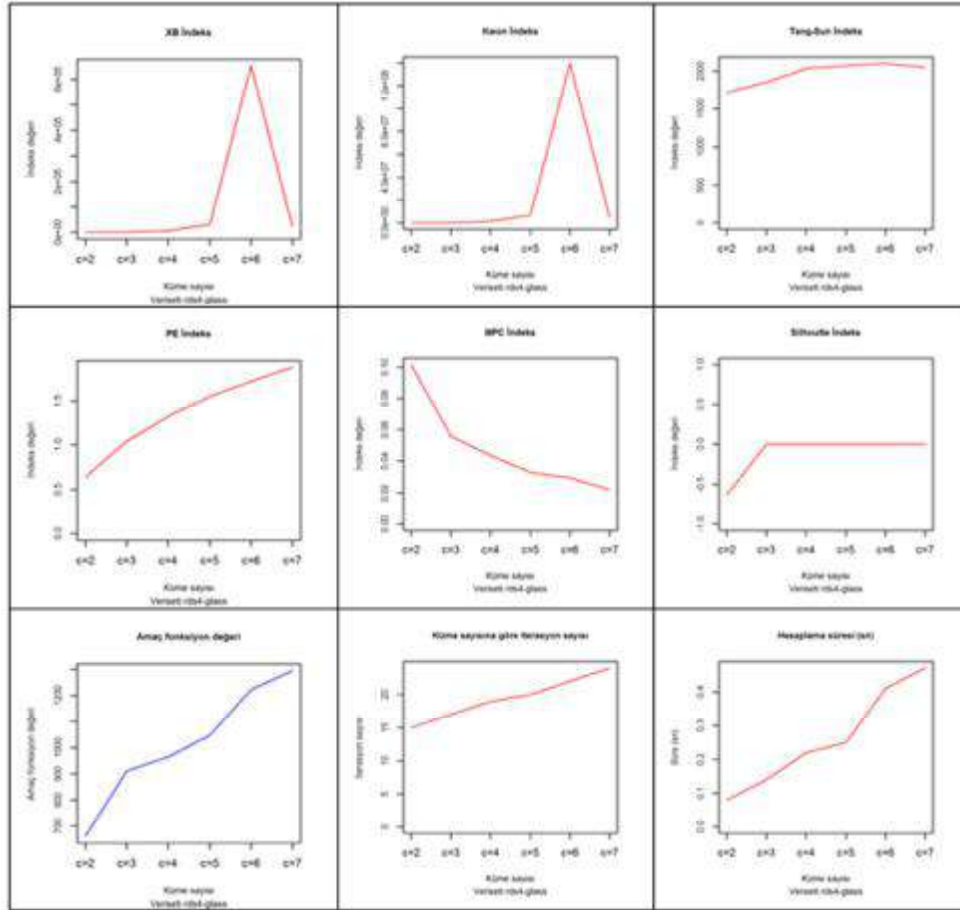
Çizelge 4.95. RDS4-Glass veri setinde BOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.29	62.210	60.126	2.573	0.838
c=3	0.154	33.276	32.742	1.875	0.579
c=4	0.558	122.009	111.932	2.092	0.638
c=5	2.53	557.415	379.088	2.278	0.692
c=6	2.032	456.216	328.974	1.904	0.576
c=7	1.73	390.416	291.405	1.671	0.505

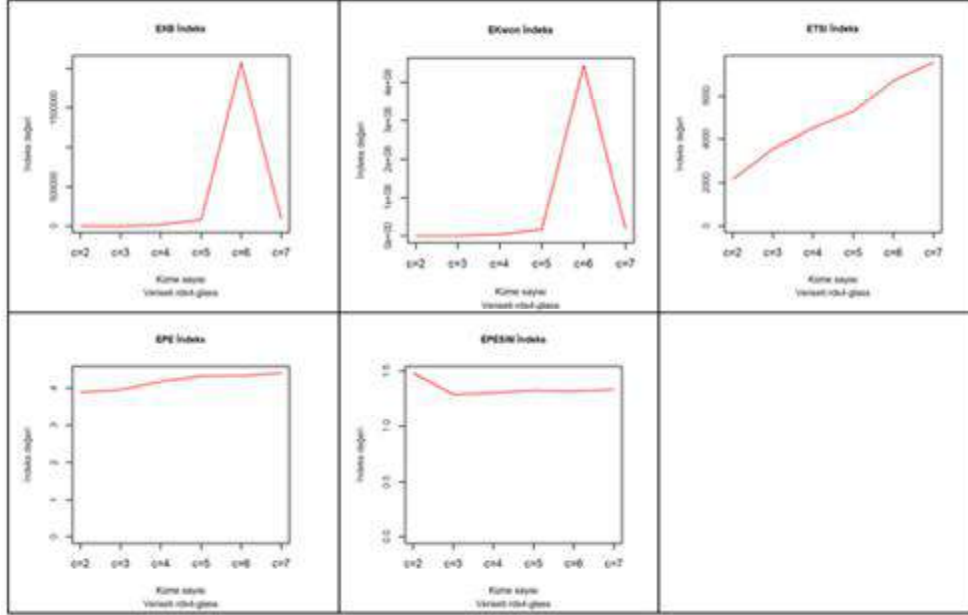
Çizelge 4.96 incelendiğinde BOCO algoritmasının başarıım sonuçları verilmektedir. Burada minimum iterasyon ve hesaplama sürelerinin küme sayısı 2 (c=2) olduğu durumlarda gözlenmektedir.

Çizelge 4.96. RDS4-Glass veri seti için BOCO algoritması başarıımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	40	0.370	406.000
c=3	54	0.590	247.074
c=4	89	1.330	72.510
c=5	143	2.720	40.320
c=6	84	1.890	28.464
c=7	68	1.900	26.099



Şekil 4.59. BOCO algoritması küme geçerlilik indeksleri ve kümeleme başarıımı (RDS4-Glass)



Şekil 4.60. BOCO algoritması uzatılmış indeks değerleri (RDS4-Glass)

4.4. OBCO Algoritması

4.4.1. Sentetik Veri Setleri

4.4.1.1. Sentetik Veri Seti 1 (SDSC)

Çizelge 4.97 incelendiğinde BOCO kümeleme algoritmasında, tüm verilerin bir kümeye tipiklik değerlerinin toplamının 1'e eşit olması kısıntının özellikle büyük veri setlerinde sorunlara yol açmasından dolayı Pal ve ark. (2005) tarafından bu kısıt yumuşatılarak BCO'nun amaç fonksiyonu (3.4) ile OCO'nun amaç fonksiyonunun (3.8) bir karması olan Olabilirlikli Bulanık C-Ortalamalar (OBCO) algoritmasıyla U ve T matrislerinin birleştirilmesi ile elde edilen U değerlerinden hesaplanan genelleştirilmiş bulanık indeks değerlerine göre optimal küme sayısının tüm indeksler tarafından 4 ($c=4$) olarak önerildiği görülmektedir. Bu durumda uygulanan yeni kümeleme algoritmasının veri seti üzerinde iyi bir basarım gösterdiği anlaşılmaktadır.

Çizelge 4.97. SDSC veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.315	251.923	215.980	0.240	0.728	0.592
c=3	0.144	115.935	105.963	0.247	0.800	0.615
c=4	0.078	63.040	58.647	0.128	0.930	0.799
c=5	0.456	368.298	257.042	0.204	0.880	0.717
c=6	0.439	355.262	250.363	0.269	0.840	0.650
c=7	0.365	296.763	217.284	0.323	0.806	0.581
c=8	0.320	261.143	193.979	0.366	0.780	0.531

Çizelge 4.98'da hem U ve hem de T değerlerini kullanarak hesaplanan uzatılmış indeks değerlerine göre optimal küme sayısının EPESIM ve EPE indeksleri dışında 4 olduğu görülmektedir. V_{EPESIM} ve V_{EPE} indekslerini tahminlerini 8 (c=8) olarak göstermektedir.

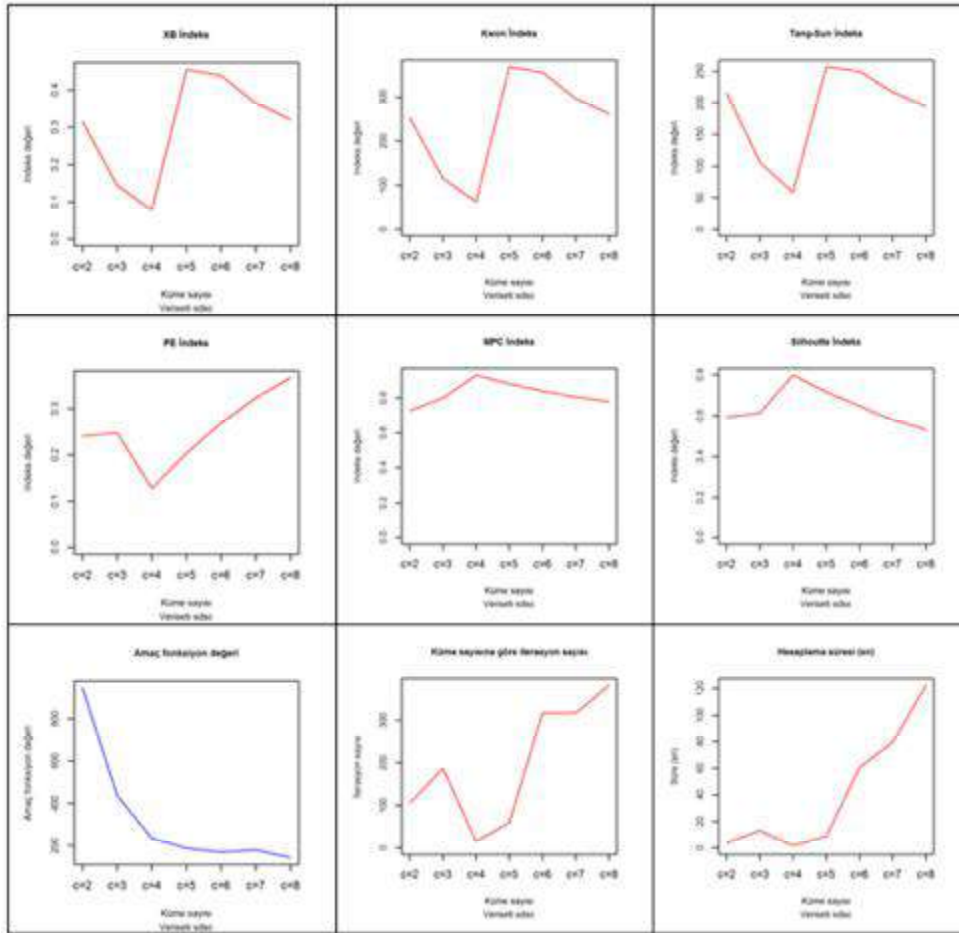
Çizelge 4.98. SDSC veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.472	378.166	324.103	18.865	5.833
c=3	0.225	180.170	164.579	15.860	4.796
c=4	0.116	93.009	86.450	12.515	3.766
c=5	0.729	586.474	408.935	12.068	3.628
c=6	0.753	606.214	426.800	11.585	3.481
c=7	0.662	533.811	390.409	10.959	3.291
c=8	0.607	490.945	364.208	10.343	3.106

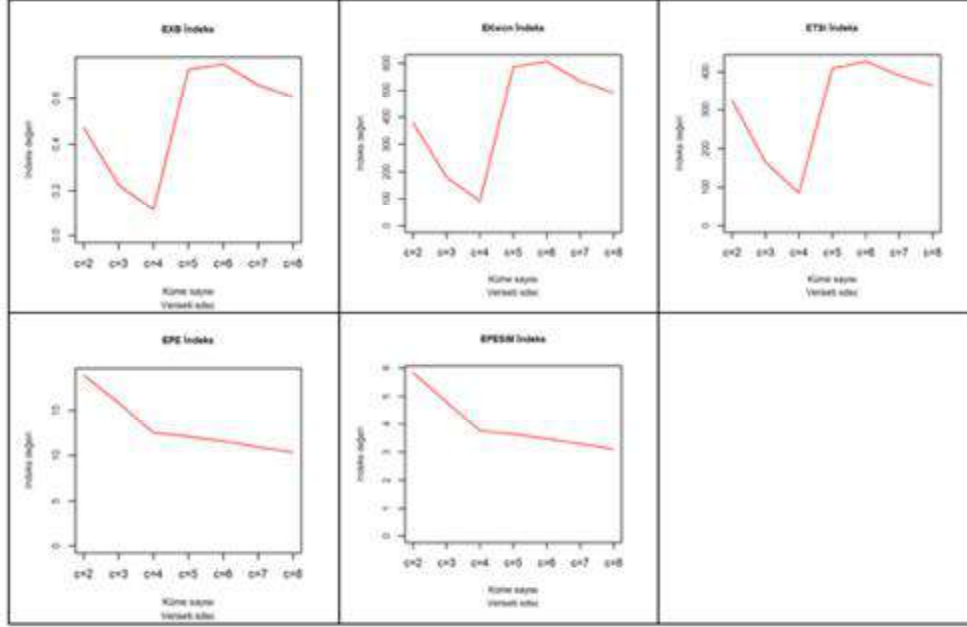
Çizelge 4.99’da OBCO kümeleme algoritmasının veri seti üzerindeki başarımı görülmektedir. Algoritmanın başarımı da gösteriyor ki minimum hesaplama süresi ve iterasyon sayısı gerçek küme sayısını ($c=4$) göstermektedir. Hesaplanan küme sayıları gerçek küme sayısına yaklaştıkça iterasyon sayıları da azalma göstermektedir.

Çizelge 4.99. SDSC veri seti için OBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	105	3.730	944.325
c=3	187	13.200	435.967
c=4	15	1.490	235.383
c=5	60	8.760	186.143
c=6	317	60.520	168.504
c=7	317	79.500	177.920
c=8	382	122.500	143.144



Şekil 4.61. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSC)



Şekil 4.62. OBCO algoritması uzatılmış indeks değerleri (SDSC)

4.4.1.2. Sentetik Veri Seti 2 (SDSCN)

Çizelge 4.100’de hesaplanan genelleştirilmiş bulanık indeks değerlerine göre optimal küme sayısının tüm indeksler tarafından 4 ($c=4$) olarak önerildiği görülmektedir. Ek 4.2’ de gerçek ve uygun küme sayısının ($c=4$) oluşturduğu küme yapıları üzerinde gürültülere karşı iyi bir basarım gösterdiği veri noktaları üzerinde görülmektedir.

Çizelge 4.100. SDSCN veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.340	279.138	237.033	0.254	0.709	0.577
c=3	0.153	125.844	114.828	0.269	0.783	0.596
c=4	0.080	65.899	61.466	0.154	0.911	0.778
c=5	0.613	506.893	329.523	0.251	0.849	0.684
c=6	0.525	435.050	295.852	0.326	0.803	0.606
c=7	0.425	353.556	252.392	0.361	0.785	0.563
c=8	0.336	280.252	208.808	0.396	0.766	0.518

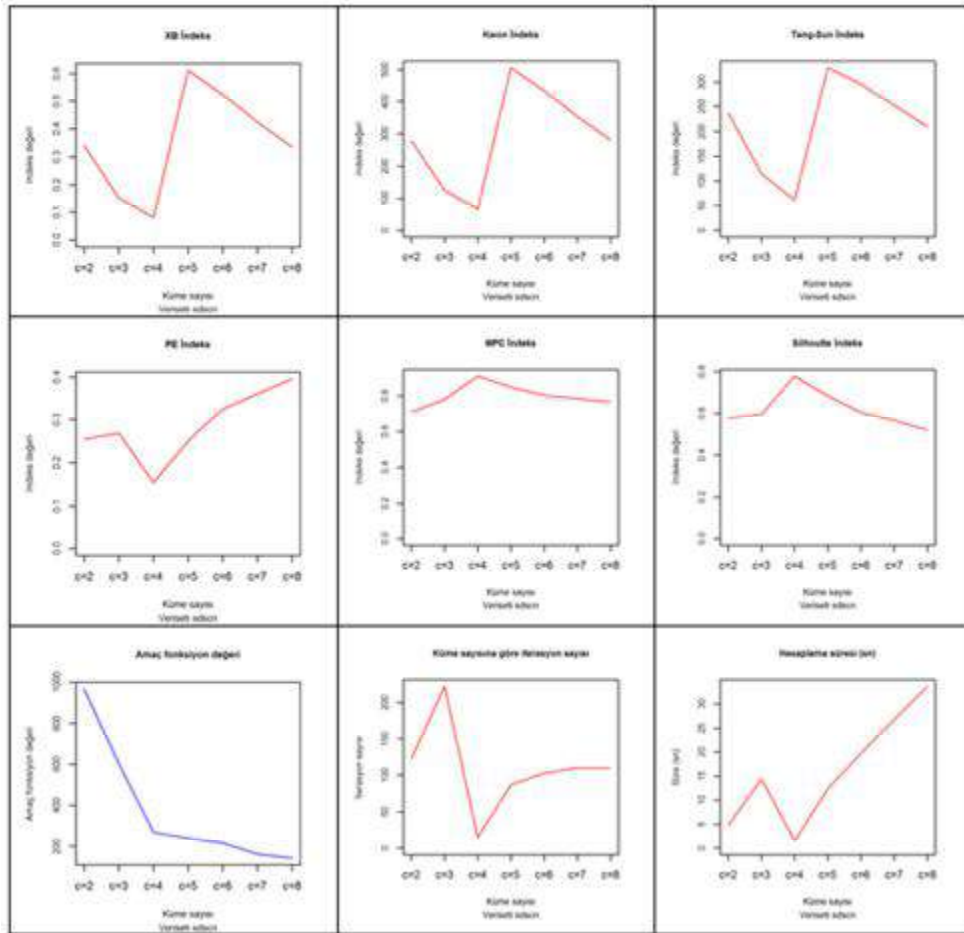
Çizelge 4.98’da hesaplanan uzatılmış indeks değerlerine göre optimal küme sayısının gürültüsüz daire şekilli veri setinden elde edilen sonuçlara benzer bir tablo görülmektedir. Burada da EPESIM ve EPE indeksleri dışında tahmin edilen uygun küme sayısı 4 olarak gözlemlenmektedir. V_{EPESIM} ve V_{EPE} indekslerini tahminlerini 8 (c=8) olarak göstermektedir.

Çizelge 4.101. SDSCN veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

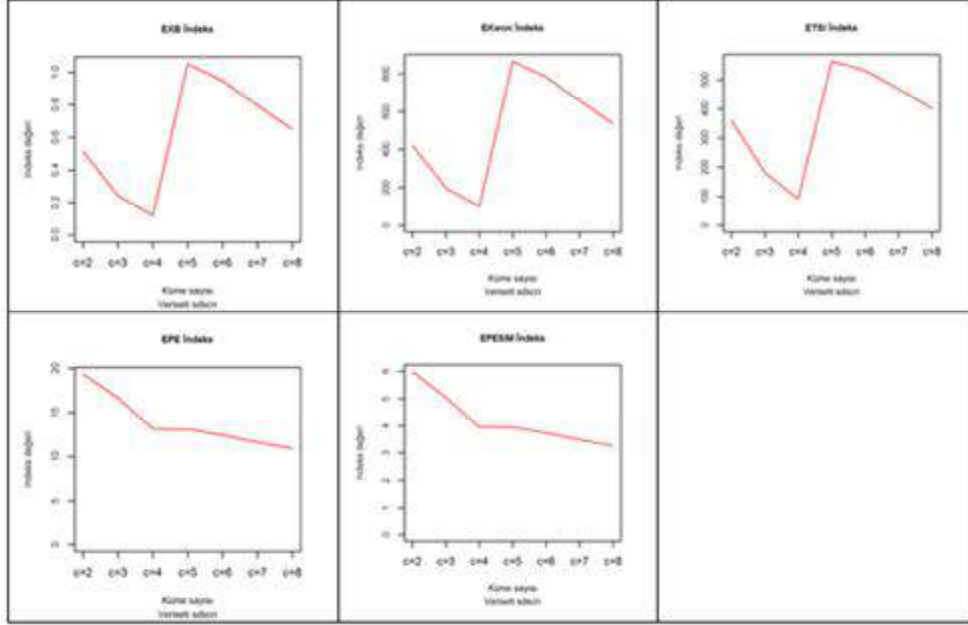
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.515	422.705	358.835	19.351	5.991
c=3	0.243	199.552	181.988	16.679	5.046
c=4	0.120	98.684	91.967	13.153	3.959
c=5	1.053	867.765	563.611	13.105	3.941
c=6	0.949	783.029	531.965	12.447	3.740
c=7	0.795	656.750	468.336	11.642	3.497
c=8	0.651	538.884	401.021	10.901	3.273

Çizelge 4.102. SDSCN veri seti için OBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	123	4.780	967.170
c=3	222	14.400	603.489
c=4	15	1.540	268.555
c=5	87	12.400	238.710
c=6	103	19.710	214.792
c=7	110	26.830	162.089
c=8	110	33.640	143.578



Şekil 4.63. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSCN)



Şekil 4.64. OBCO algoritması uzatılmış indeks değerleri (SDSCN)

4.4.1.3. Sentetik Veri Seti 3 (SDSE)

Çizelge 4.103’de elips şekilli gürültüsüz veri setinden hesaplanan genelleştirilmiş indeks değerleri verilmektedir. Sonuçlar, kullanılan indeks değerlerinin gerçek küme sayısına ($c=4$) isabet ettiklerini göstermektedir.

Çizelge 4.103. SDSE veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
$c=2$	0.309	247.459	211.599	0.276	0.655	0.521
$c=3$	0.183	147.098	132.660	0.283	0.778	0.598
$c=4$	0.110	88.606	81.843	0.243	0.841	0.686
$c=5$	0.322	259.138	209.167	0.312	0.810	0.628
$c=6$	0.271	218.887	180.046	0.344	0.799	0.594
$c=7$	0.219	176.940	150.509	0.352	0.801	0.601
$c=8$	0.162	131.442	114.629	0.325	0.821	0.631

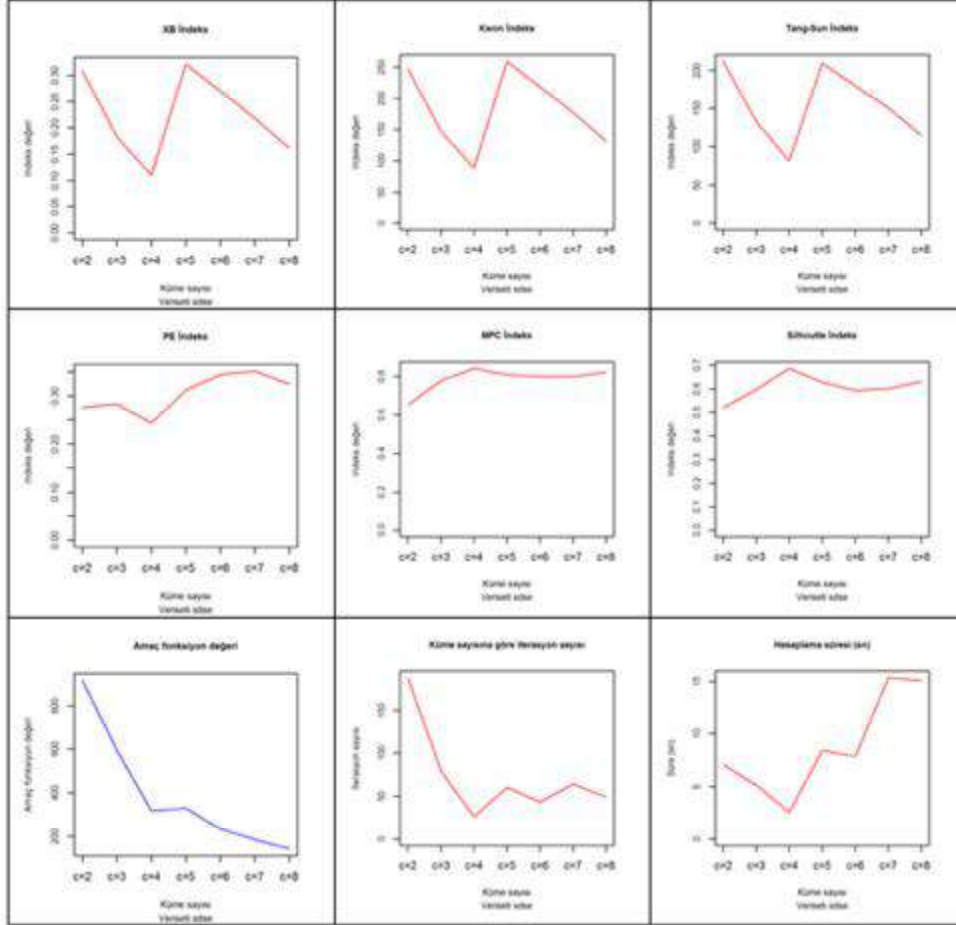
Uzatılmış indeks sonuçları Şekil 4.66'da oluşturdukları yapı ve Çizelge 4.104'deki sonuçlar itibariyle genel tahminlerini 4 ($c=4$) olarak yapmaktadırlar. Özellikle Şekil 4.66'da V_{EXB} , V_{EKwon} ve V_{ETS} indekslerinin küme sayıları arasındaki artış-azalış eğilimleri birbirlerine çok yakın olduğu açıkça görülmektedir. Buna benzer bir yapıda EPESIM ve EPE arasında görülse de, yapmış oldukları tahmin beklenen tahmini göstermemektedir.

Çizelge 4.104. SDSE veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

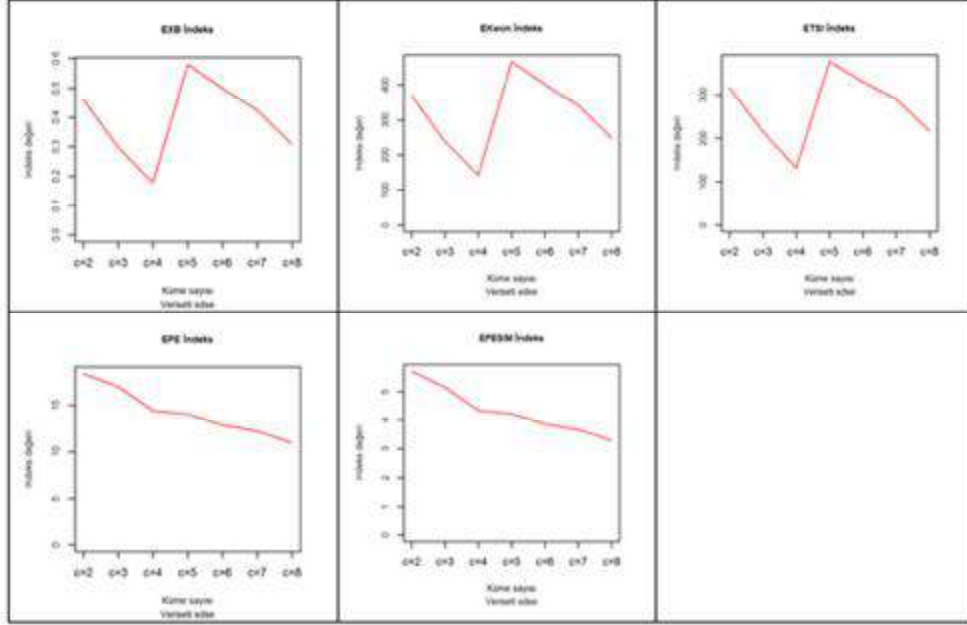
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.461	369.279	315.660	18.382	5.712
c=3	0.298	238.833	215.286	16.997	5.146
c=4	0.178	142.828	131.829	14.418	4.342
c=5	0.581	466.254	376.051	14.039	4.222
c=6	0.498	400.767	329.354	12.822	3.853
c=7	0.424	341.502	290.207	12.185	3.660
c=8	0.309	249.139	217.026	10.944	3.286

Çizelge 4.105. SDSE veri seti için OBCO algoritması başarıımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	188	7.080	917.134
c=3	79	5.130	594.638
c=4	26	2.560	317.988
c=5	60	8.440	329.949
c=6	43	7.850	233.483
c=7	64	15.390	184.765
c=8	49	15.060	143.642



Şekil 4.65. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSE)



Şekil 4.66. OBCO algoritması uzatılmış indeks değerleri (SDSE)

4.4.1.4. Sentetik Veri Seti 4 (SDSEN)

Çizelge 4.106'ya bakıldığında elips şekilli gürültülü veri setinden hesaplanan genelleştirilmiş indeks değerleri verilmektedir. İndeks değerleri tahminini gerçek küme sayısı ($c=4$) olarak göstererek uygulanan kümeleme algoritmasının da iyi bir basarım gösterdiğini desteklemektedir.

Çizelge 4.106. SDSEN veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	<i>V_{XB}</i>	<i>V_{Kwon}</i>	<i>V_{TS}</i>	<i>V_{PE}</i>	<i>V_{MPC}</i>	<i>V_{Fsil}</i>
c=2	0.322	264.243	225.079	0.264	0.679	0.539
c=3	0.159	131.142	119.406	0.262	0.794	0.618
c=4	0.091	74.892	69.539	0.195	0.879	0.742
c=5	0.291	240.958	192.847	0.255	0.848	0.692
c=6	0.250	207.281	169.801	0.287	0.834	0.649
c=7	0.190	158.273	133.872	0.284	0.840	0.657
c=8	0.163	136.567	116.101	0.286	0.841	0.652

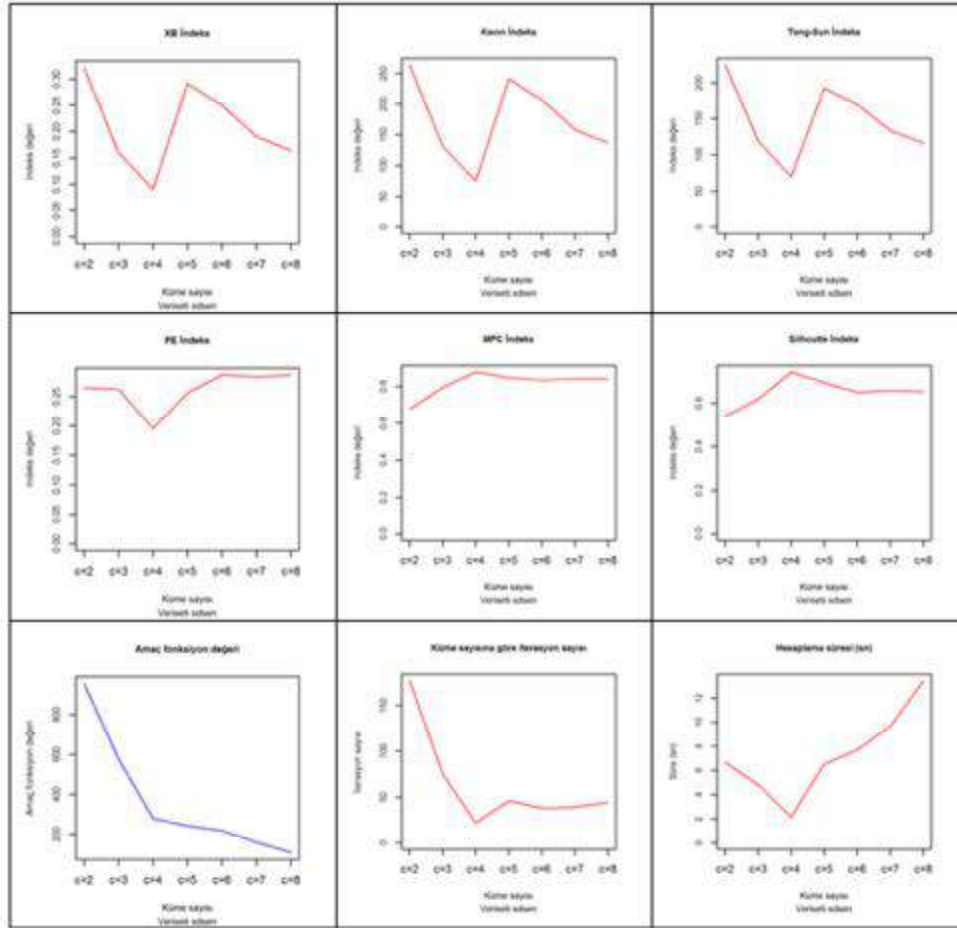
Çizelge 4.107. SDSEN veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	0.483	396.565	337.684	19.022	5.904
c=3	0.253	207.846	189.149	16.825	5.091
c=4	0.140	115.051	106.740	13.690	4.121
c=5	0.487	401.145	320.770	13.060	3.927
c=6	0.439	362.012	296.271	12.297	3.695
c=7	0.336	277.640	234.592	11.127	3.342
c=8	0.290	240.610	204.307	10.128	3.041

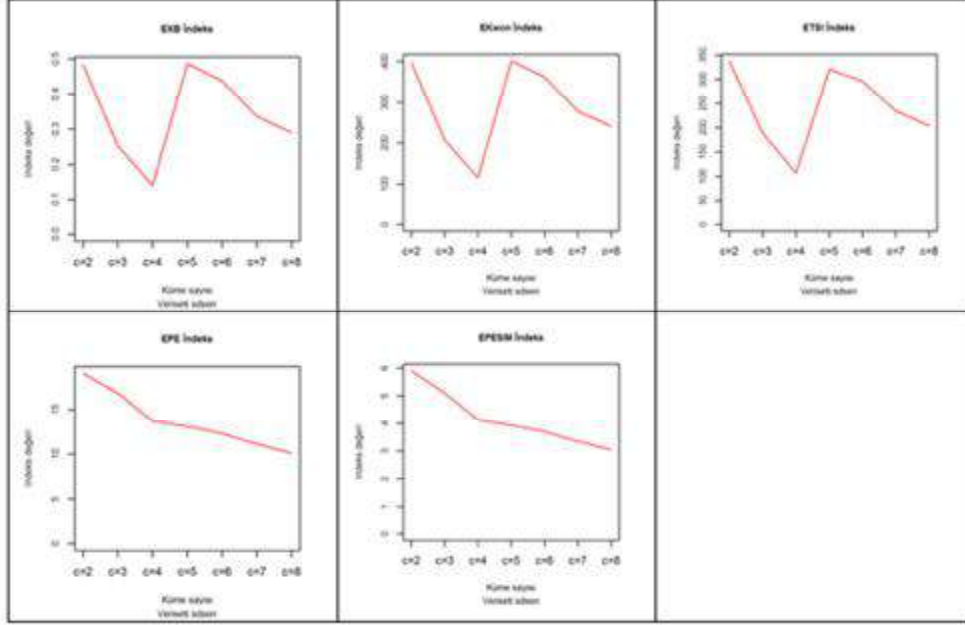
Çizelge 4.107’de hesaplanmış uzatılmış indeks değerleri ve 4.108’de hesaplanmış algoritma başarımları tahminlerindeki küme sayısını 4 (c=4), yani gerçek küme sayısı (c=4) olarak göstermektedir.

Çizelge 4.108. SDCEN veri seti için OBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	177	6.670	956.524
c=3	75	4.810	577.907
c=4	22	2.160	281.380
c=5	46	6.5100	242.484
c=6	38	7.680	219.304
c=7	39	9.670	159.899
c=8	44	13.440	108.753



Şekil 4.67. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSEN)



Şekil 4.68. OBCO algoritması uzatılmış indeks değerleri (SDSEN)

4.4.1.5. Sentetik Veri Seti 5 (SDSR)

Kare şekilli gürültüsüz veri setinden hesaplanan genelleştirilmiş bulanık indeks değerleri Çizelge 4.109’da gösterilmektedir. Elde edilen sonuçlara göre indeksler tahminlerini 4 ($c=4$) olarak göstermektedir.

Çizelge 4.109. SDRS veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
$c=2$	0.316	253.059	216.673	0.235	0.735	0.594
$c=3$	0.154	123.343	112.418	0.255	0.796	0.614
$c=4$	0.087	69.924	64.862	0.152	0.912	0.780
$c=5$	0.446	360.252	257.823	0.235	0.860	0.697
$c=6$	0.345	279.504	211.558	0.288	0.829	0.632
$c=7$	0.332	269.168	204.268	0.328	0.807	0.592
$c=8$	0.289	235.191	181.099	0.358	0.792	0.549

EPESIM ve EPE uzatılmış indeks değerlerinin tahminlerindeki küme sayısını 8 ($c=8$) olarak tahmin etse de genel olarak diğer indeksler tahminlerinde gerçek küme sayısına ($c=4$) isabet etmişlerdir.

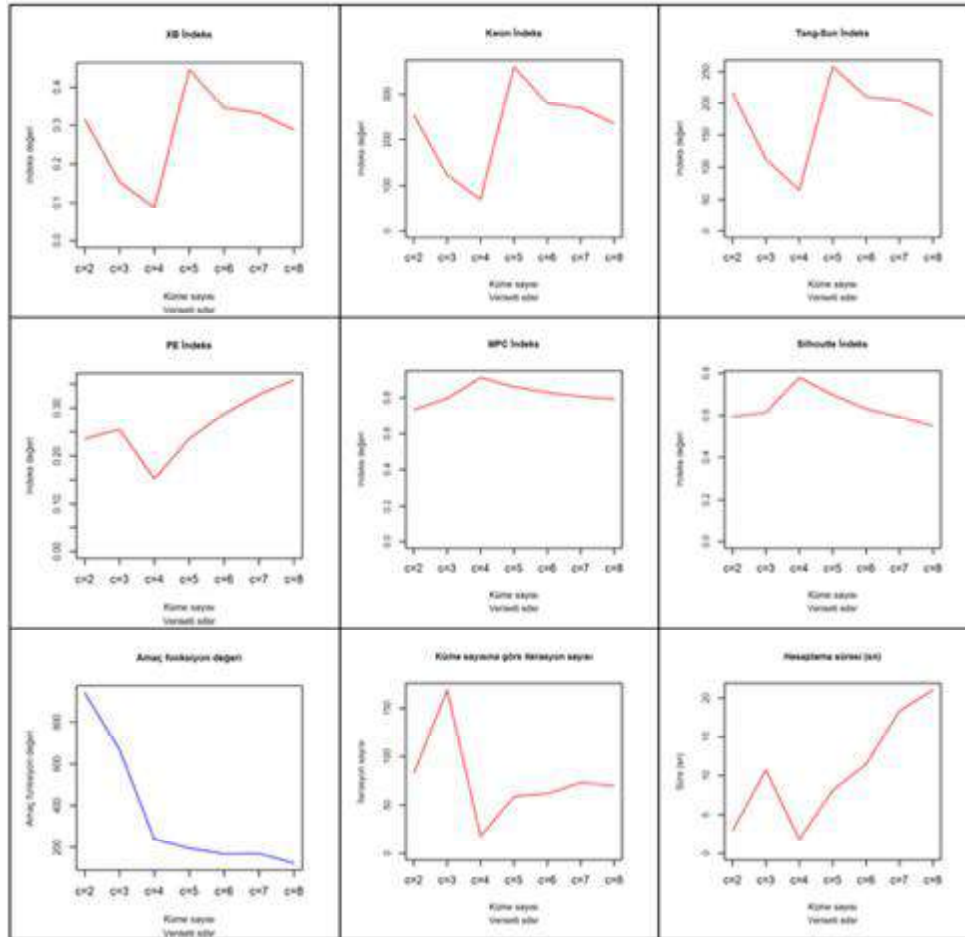
Çizelge 4.110. SDSR veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
$c=2$	0.473	378.612	324.068	18.734	5.792
$c=3$	0.243	194.392	177.075	16.223	4.907
$c=4$	0.130	104.602	96.947	12.898	3.882
$c=5$	0.744	598.569	427.994	12.626	3.796
$c=6$	0.606	488.162	369.111	11.848	3.560
$c=7$	0.604	487.126	369.277	11.174	3.356
$c=8$	0.546	441.036	339.184	10.462	3.141

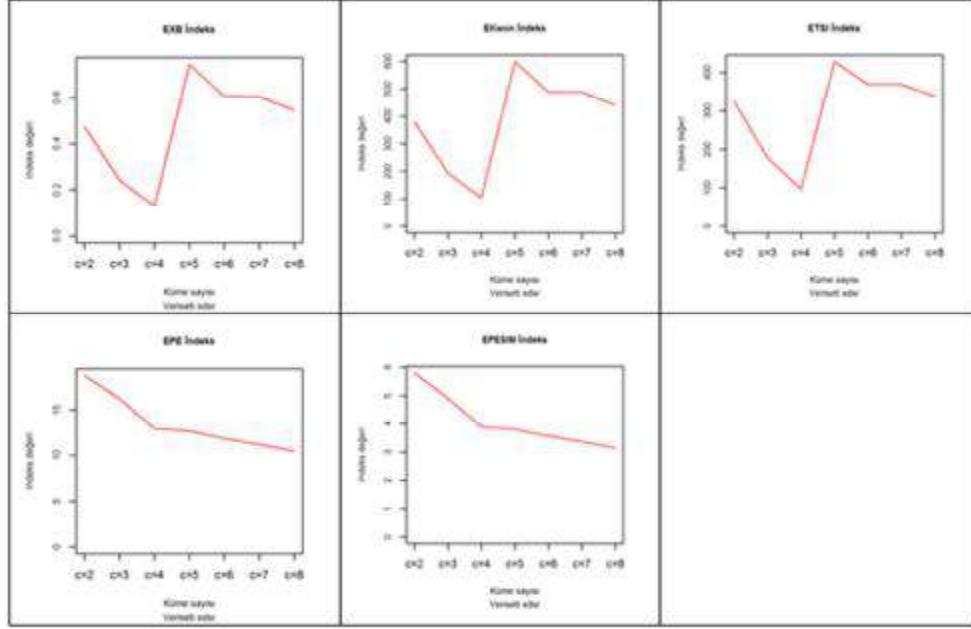
Çizelge 4.111’de OBCO algoritmasının başarımların değerleri gözlenmektedir. Burada hesaplanan minimum değer gerçek küme sayısına eşit olduğu ve diğer küme sayılarından çok daha başarılı gösterdiği anlaşılmaktadır.

Çizelge 4.111. SDSR veri seti için OBCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	83	2.950	940.378
$c=3$	169	10.740	666.840
$c=4$	18	1.830	238.534
$c=5$	59	8.140	194.867
$c=6$	62	11.50	167.833
$c=7$	73	18.340	169.907
$c=8$	70	21.040	124.003



Şekil 4. 69. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR)



Şekil 4.70. OBCO algoritması uzatılmış indeks değerleri (SDSR)

4.4.1.6. Sentetik Veri Seti 6 (SDSRN)

Kare şekilli gürültülü sentetik veri seti üzerinden hesaplanan indeks değerleri (Çizelge 4.112) tahminlerini 4 ($c=4$) yani isabetli bir şekilde yapmaktadır ki bu durum algoritmanın iyi bir basarım gösterdiğini ifade etmektedir.

Çizelge 4.112. SDSRN veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
$c=2$	0.329	269.950	229.968	0.240	0.729	0.589
$c=3$	0.162	133.317	121.164	0.260	0.793	0.606
$c=4$	0.085	70.281	65.284	0.162	0.905	0.773
$c=5$	0.439	362.869	260.882	0.243	0.854	0.695
$c=6$	0.367	303.840	228.444	0.309	0.814	0.611
$c=7$	0.360	298.879	223.807	0.353	0.792	0.573
$c=8$	0.279	232.896	181.815	0.374	0.783	0.546

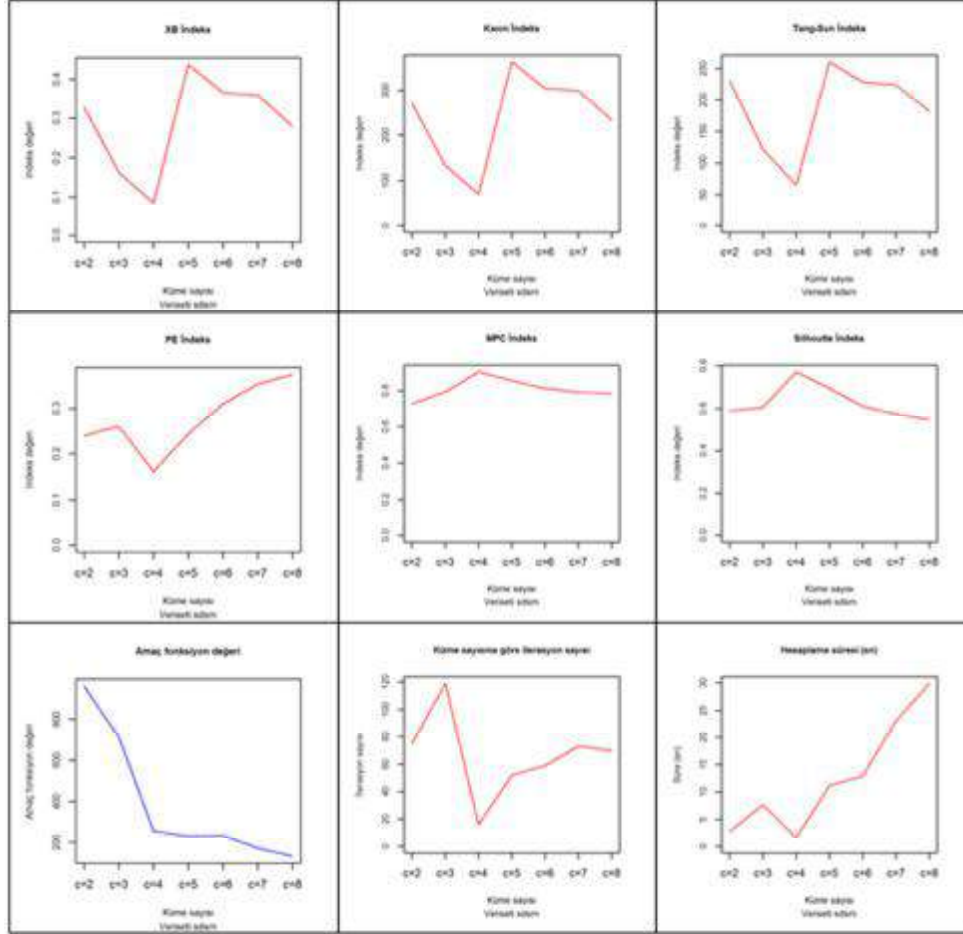
Çizelge 4.113. incelendiğinde görülüyor ki uzatılmış indeks değerleri tahminlerini EPESIM ve EPE dışında 4 (c=4) olarak göstermektedir.

Çizelge 4.113. SDSRN veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

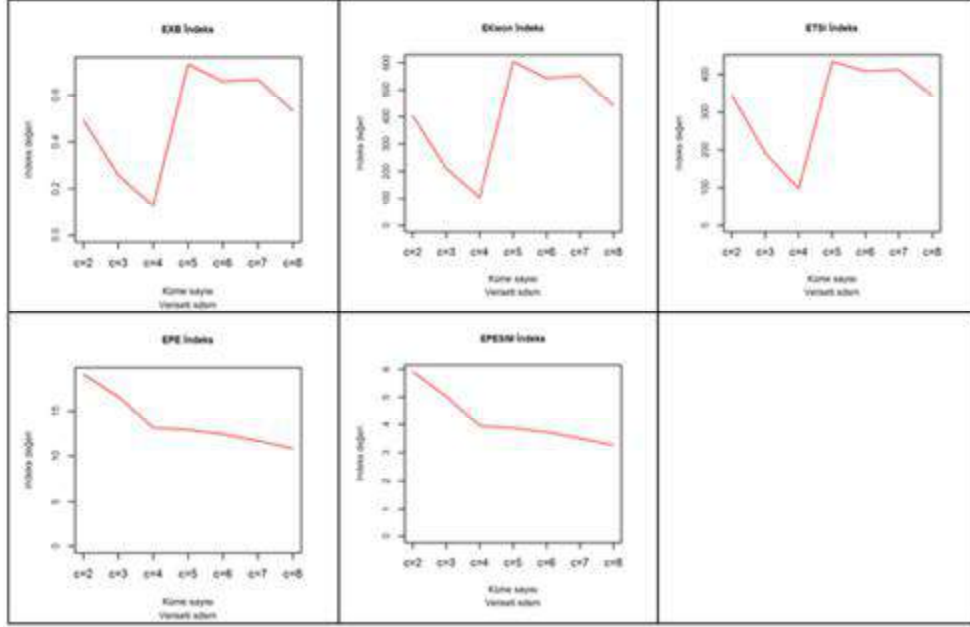
<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	0.493	404.320	344.331	19.085	5.904
c=3	0.256	210.620	191.321	16.611	5.025
c=4	0.127	104.931	97.389	13.126	3.950
c=5	0.733	604.337	434.096	12.863	3.868
c=6	0.659	543.563	408.290	12.379	3.719
c=7	0.668	551.713	412.709	11.647	3.498
c=8	0.533	441.323	344.126	10.857	3.260

Çizelge 4.114. SDSRN veri seti için OBCO algoritması başarımları

<i>C</i>	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	75	2.770	961.320
c=3	119	7.670	714.596
c=4	16	1.610	257.791
c=5	52	11.140	230.855
c=6	59	12.890	233.031
c=7	73	23.130	172.721
c=8	70	29.940	134.096



Şekil 4.71. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN)



Şekil 4.72. OBCO algoritması uzatılmış indeks değerleri (SDSRN)

4.4.1.7. Sentetik Veri Seti 7 (SDSX)

Çizelge 4.115 incelendiğinde hesaplanan indeks değerleri tahminlerini gerçek küme sayısından ($c=3$) farklı olarak 5 ($c=5$) olarak göstermektedir. V_{PE} indeksi ise tahminini 2 ($c=2$) yönünde yaparak diğer indekslere göre gerçek kümeye en yakın tahmini gösteren genelleştirilmiş bulanık indekstir. Genel anlamda veri seti üzerinde indeks değerlerinin gerçek küme sayısına herhangi bir isabet gösterememesi, uygulanan algoritmanın da iyi bir basarım göstermediği anlamına gelmektedir.

Çizelge 4.115. SDSX veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	<i>V_{XB}</i>	<i>V_{Kwon}</i>	<i>V_{TS}</i>	<i>V_{PE}</i>	<i>V_{MPC}</i>	<i>V_{Fsil}</i>
c=2	0.547	439.212	343.405	0.327	0.598	0.489
c=3	0.390	312.927	260.381	0.452	0.623	0.489
c=4	0.227	182.347	161.627	0.431	0.703	0.548
c=5	0.141	114.179	104.322	0.342	0.791	0.650
c=6	0.566	457.764	329.315	0.391	0.778	0.580
c=7	0.391	316.835	247.780	0.390	0.785	0.597
c=8	0.292	237.206	195.253	0.395	0.788	0.616

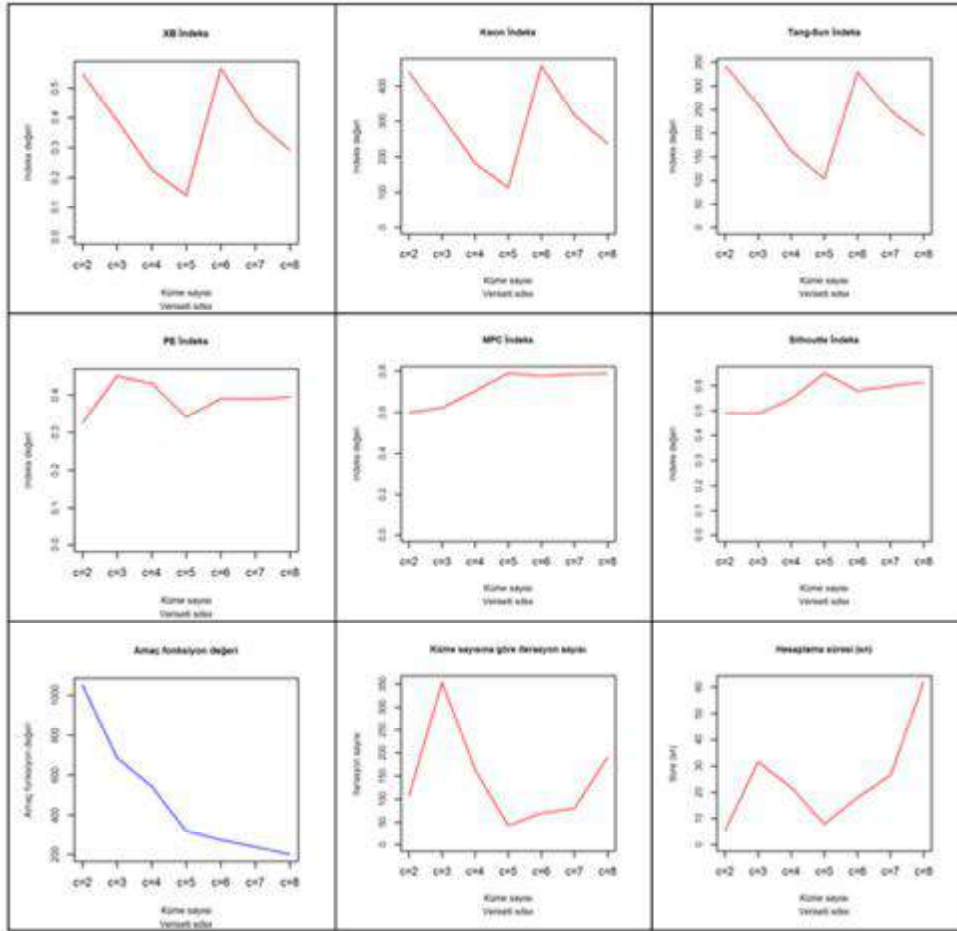
Çizelge 4.116. SDSX veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	0.839	673.396	526.401	18.028	5.662
c=3	0.706	566.746	471.442	18.225	5.541
c=4	0.433	347.736	308.049	17.029	5.136
c=5	0.259	208.793	190.608	14.641	4.405
c=6	1.107	891.455	640.790	13.999	4.208
c=7	0.787	634.339	495.649	13.098	3.935
c=8	0.617	498.209	409.701	12.488	3.750

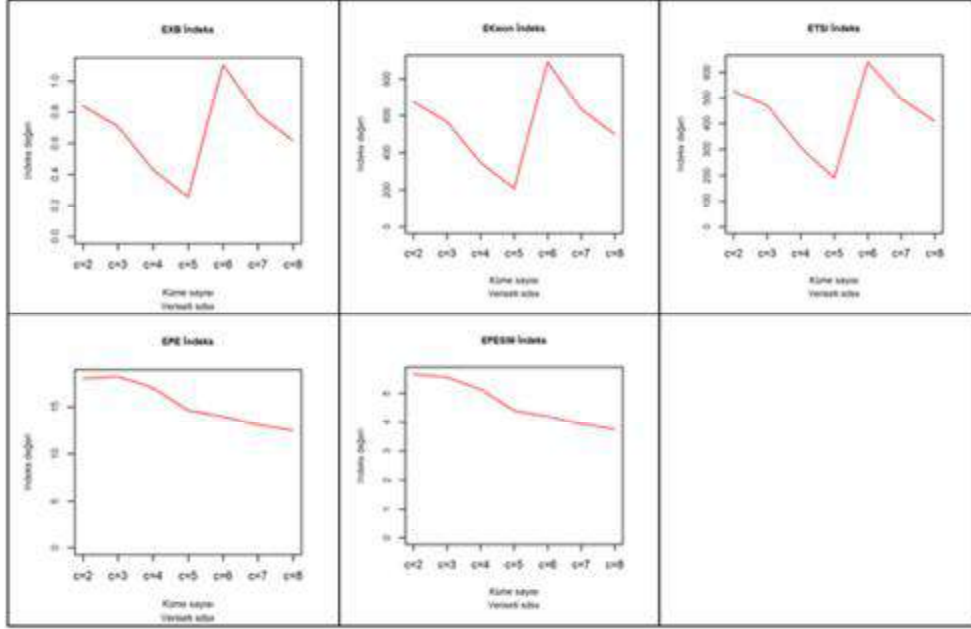
Çizelge 4.116'da gösterilen uzatılmış indeks tahminleri ve Çizelge 4.117'deki algoritmanın başarımları hiçbir şekilde beklenen küme sayısını tahminini isabetli bir şekilde gösterememektedir.

Çizelge 4.117. SDSX veri seti için OBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	107	5.530	1052.744
$c=3$	354	31.510	686.820
$c=4$	164	21.920	544.001
$c=5$	42	8.000	319.393
$c=6$	70	18.030	272.779
$c=7$	80	26.540	238.863
$c=8$	190	61.810	201.240



Şekil 4.73. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX)



Şekil 4.74. OBCO algoritması uzatılmış indeks değerleri (SDSX)

4.4.1.8. Sentetik Veri Seti 8 (SDSXN)

Çizelge 4.118’de hesaplanan indeks değerleri genel anlamda tahminlerini 5 ($c=5$) olarak yapabilirler. V_{PE} indeksi tahminini 2 ($c=2$), V_{PE} ve V_{PE} ise tahminlerini 8 ($c=8$) olarak göstermektedirler. Bu durumda herhangi bir indeksin gerçek küme sayısına ($c=3$) isabeti söz konusu olmamakla birlikte uygulanan algoritmanın veri seti üzerindeki başarımından bahsetmek mümkün değildir.

Çizelge 4.118. SDSXN veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.536	440.552	346.200	0.329	0.594	0.485
c=3	0.313	257.234	221.132	0.433	0.641	0.505
c=4	0.262	216.129	188.846	0.470	0.674	0.518
c=5	0.199	164.826	146.570	0.404	0.748	0.594
c=6	0.515	426.167	320.732	0.436	0.751	0.543
c=7	0.364	302.188	242.645	0.415	0.772	0.581
c=8	0.249	207.389	175.537	0.380	0.798	0.626

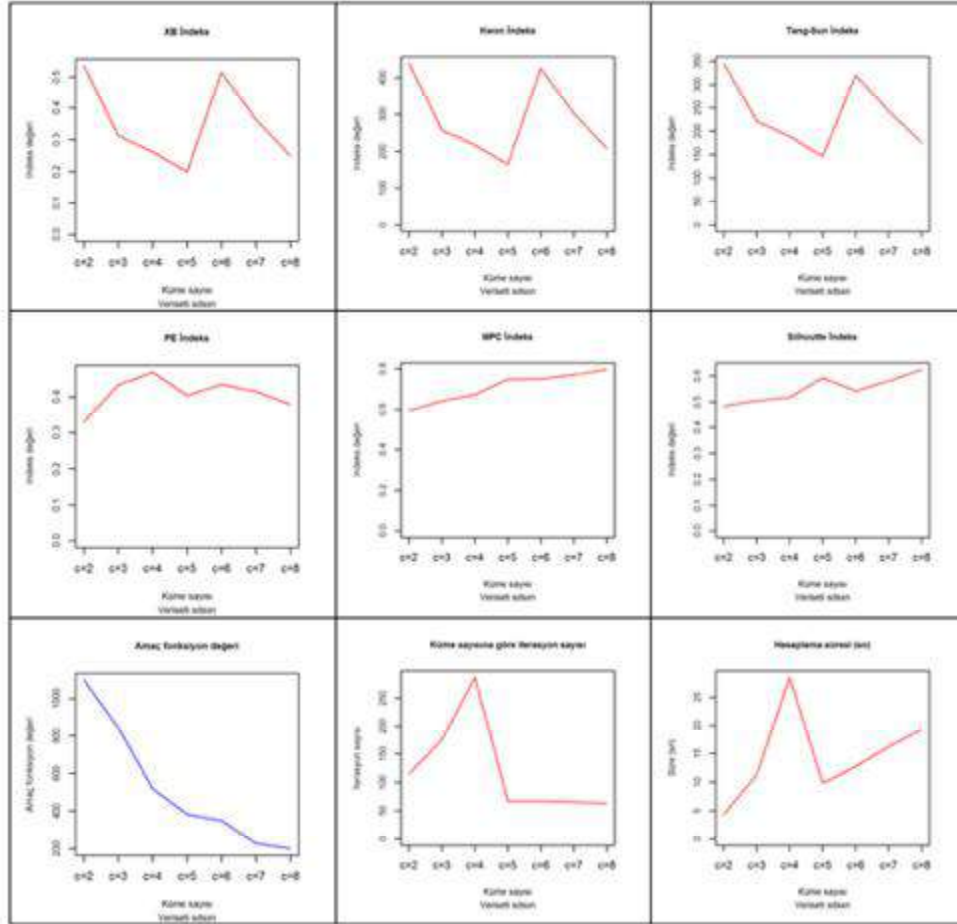
Çizelge 4.119’da uzatılmış indeks değerlerinden V_{EXB} , V_{Kwon} ve V_{ETS} tahminlerini 5 (c=5), V_{EPE} ve V_{EPESIM} indeksleri ise 8 (c=8) olarak göstermektedir.

Çizelge 4.119. SDSXN veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

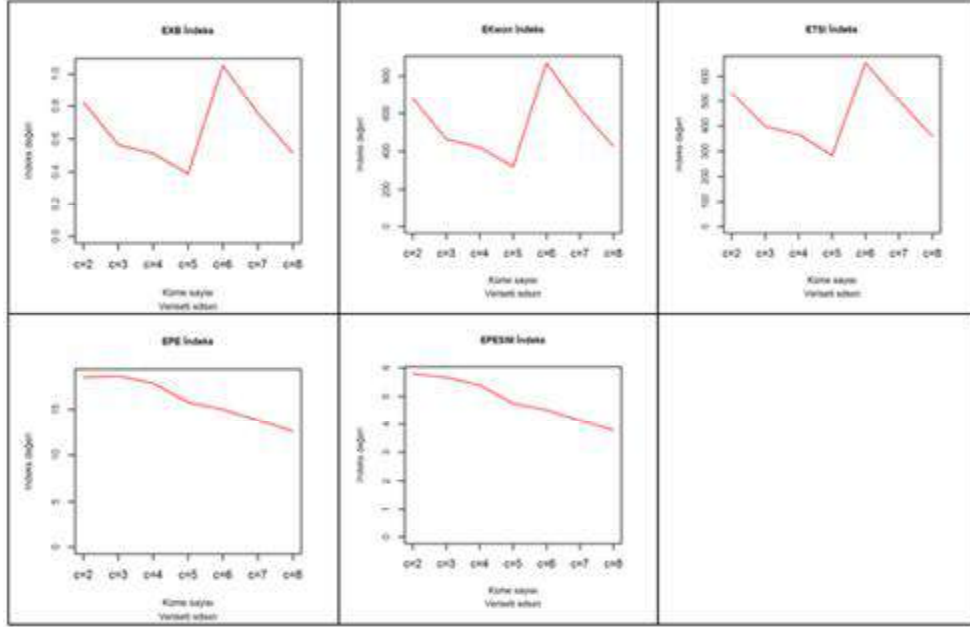
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.824	677.899	532.609	18.507	5.806
c=3	0.563	462.869	397.785	18.628	5.659
c=4	0.512	421.125	367.787	17.873	5.391
c=5	0.386	318.458	282.983	15.796	4.752
c=6	1.055	869.949	654.261	15.019	4.515
c=7	0.756	624.379	500.940	13.800	4.146
c=8	0.514	425.602	359.895	12.601	3.784

Çizelge 4.120. SDSXN veri seti için OBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	115	4.270	1100.491
c=3	175	11.300	843.755
c=4	287	28.500	519.533
c=5	67	9.830	382.222
c=6	67	12.760	349.371
c=7	66	16.160	227.208
c=8	63	19.210	199.763



Şekil 4.75. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSXN)



Şekil 4.76. OBCO algoritması uzatılmış indeks değerleri (SDSXN)

4.4.2. Gerçek Veri Setleri

4.4.2.1. Gerçek Veri Seti 1 (RDS1-Iris)

RDS1-Iris veri setinden hesaplanan genelleştirilmiş bulanık indeks değerleri Çizelge 4.121’de gösterilmektedir. Ek 4.9 ve Ek 4.9 (Devamı)’da oluşan küme yapıları incelendiğinde gürültüler gözlene de elde edilen değerler gerçek küme sayısına (2,3) isabet ederekten iyi bir başarı göstermiştir.

Çizelge 4.121. RDS1-Iris veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.058	8.956	8.925	0.057	0.938	0.850
c=3	0.167	26.518	24.280	0.158	0.864	0.728
c=4	0.288	46.232	38.759	0.218	0.846	0.668
c=5	0.383	62.471	49.431	0.295	0.806	0.646
c=6	1.098	179.439	101.948	0.337	0.800	0.636
c=7	0.712	123.929	80.117	0.397	0.764	0.532

Çizelge 4.122’de verilen uzatılmış indeks değerlerinin sonuçları tahminlerini genel anlamda 2 (c=2) olarak yapmaktadırlar. EPE ve EPESIM indeksleri ise tahminlerini 7 (c=7) olarak göstermektedir.

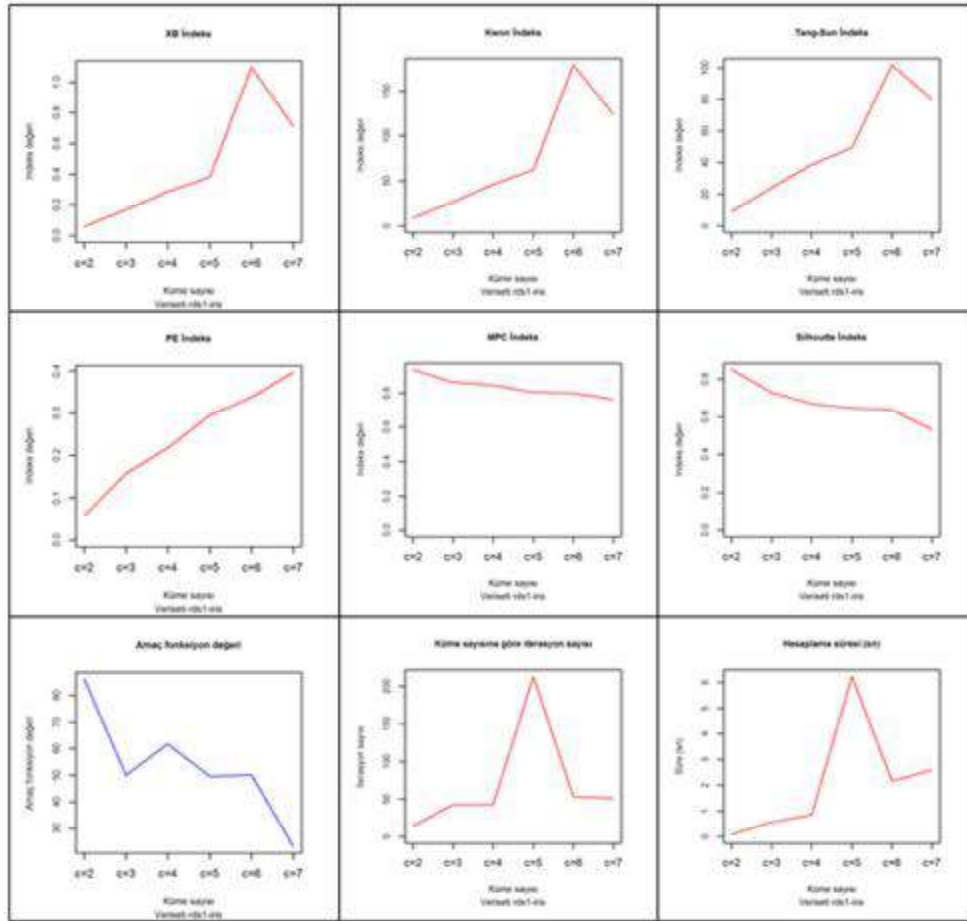
Çizelge 4.122. RDS1-Iris veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.072	11.117	11.020	3.042	0.950
c=3	0.239	37.299	33.878	3.134	0.962
c=4	0.458	71.709	59.658	3.116	0.948
c=5	0.658	103.662	81.380	3.058	0.926
c=6	2.068	324.962	183.285	3.038	0.918
c=7	1.443	233.649	149.451	2.819	0.849

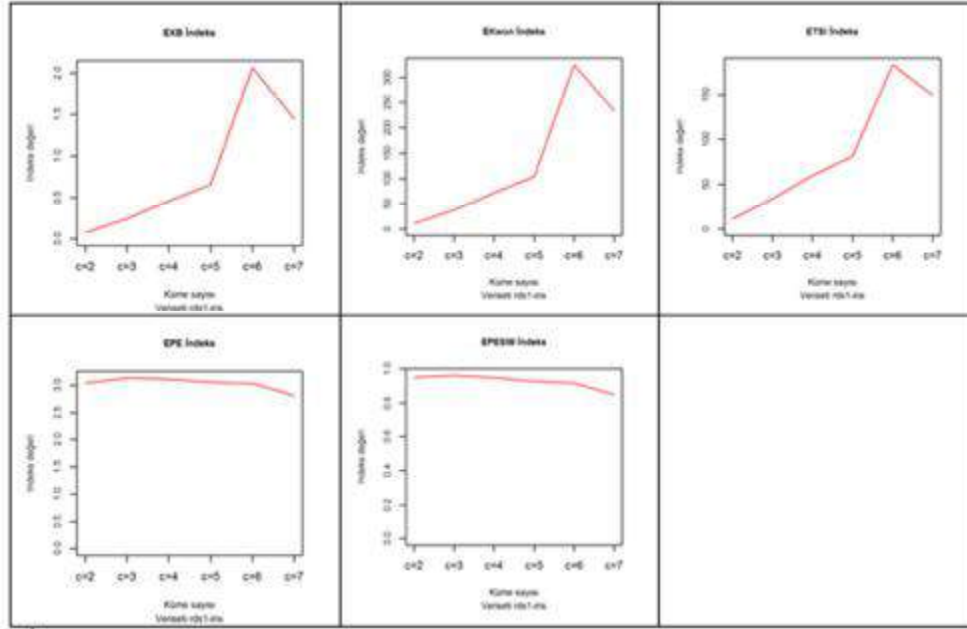
RDS1-Iris veri seti üzerinde uygulanan kümeleme algoritmasından alınan basarım sonuçlar Çizelge 4.123’de gösterilmektedir. Minimum iterasyon sayısı ve hesaplama suresi küme sayısı 2 (c=2)’de gözlenmektedir.

Çizelge 4.123. RDS1-Iris veri seti için OBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	14	0.090	86.076
$c=3$	43	0.560	49.971
$c=4$	43	0.850	61.785
$c=5$	213	6.230	49.609
$c=6$	53	2.170	50.124
$c=7$	51	2.600	23.372



Şekil 4.77. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS1-Iris)



Şekil 4.78. OBCO algoritması uzatılmış indeks değerleri (RDS1-Iris)

4.4.2.2. Gerçek Veri Seti 2 (RDS2-Quail)

RDS2-Quail veri setinden hesaplanan genelleştirilmiş bulanık indeks değerleri Çizelge 4.124'de gösterilmektedir. Ek 4.10 ve Ek 4.10 (Devamı)'da oluşan küme yapıları incelendiğinde gürültüler gözlenmektedir. Ayrıca her küme sayısı içerisindeki kümeler karmaşık yapılar oluşturmaktadır. Elde edilen değerler incelendiğinde genel tahmin 2 ($c=2$)'ye isabet etmektedir yalnızca V_{MPC} indeksi tahminini 4 ($c=4$) olarak yapmaktadır. Herhangi bir indeksin gerçek küme sayısına ($c=3$) isabet etmemesi ancak genel anlamda yapılan tahminlerin ona yakın olması durumundan dolayı algoritmanın çokta kötü bir basarım göstermediğinden bahsetmek mümkündür.

Çizelge 4.124. RDS2-Quail veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.088	8.156	8.405	0.097	0.889	0.772
c=3	0.128	12.583	13.090	0.150	0.872	0.721
c=4	0.168	17.641	18.459	0.153	0.891	0.726
c=5	0.130	15.016	15.826	0.167	0.887	0.706
c=6	0.254	30.771	32.326	0.194	0.875	0.639

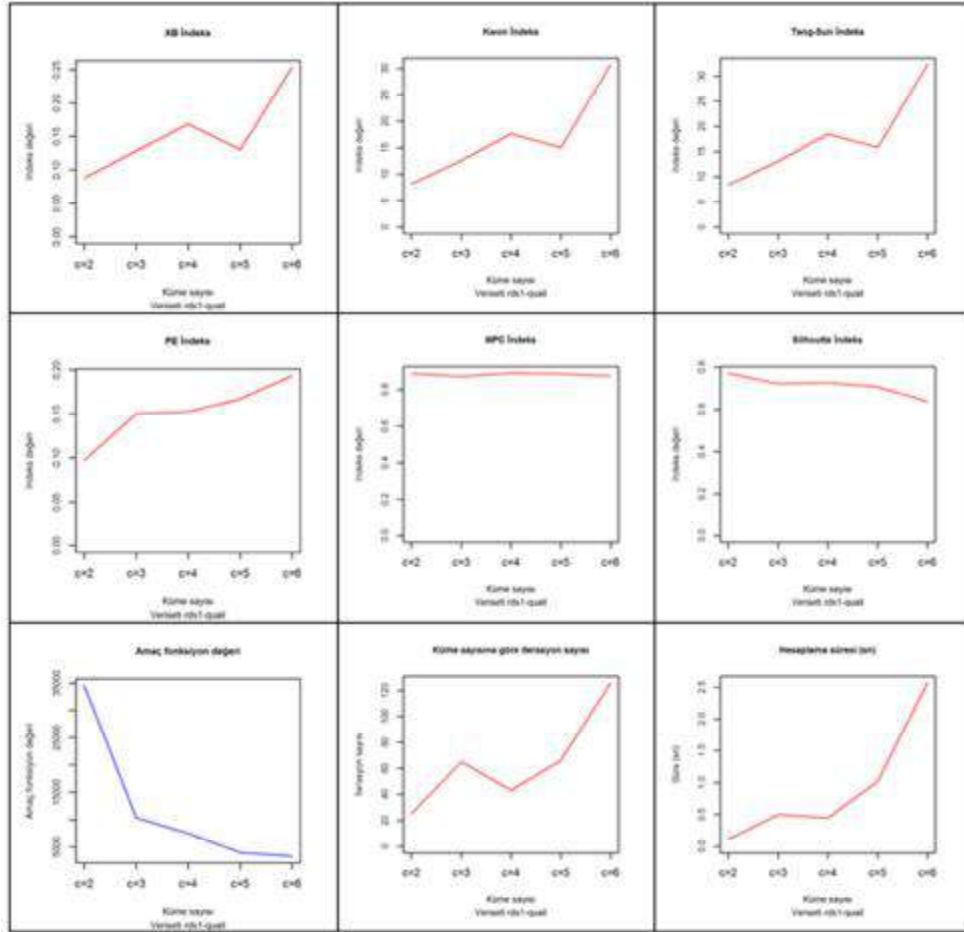
Çizelge 4.125. RDS2-Quail veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.111	10.269	10.518	2.289	0.755
c=3	0.180	17.208	17.714	2.115	0.657
c=4	0.242	24.269	25.084	1.876	0.572
c=5	0.195	20.857	21.664	1.651	0.500
c=6	0.386	42.652	44.195	1.553	0.469

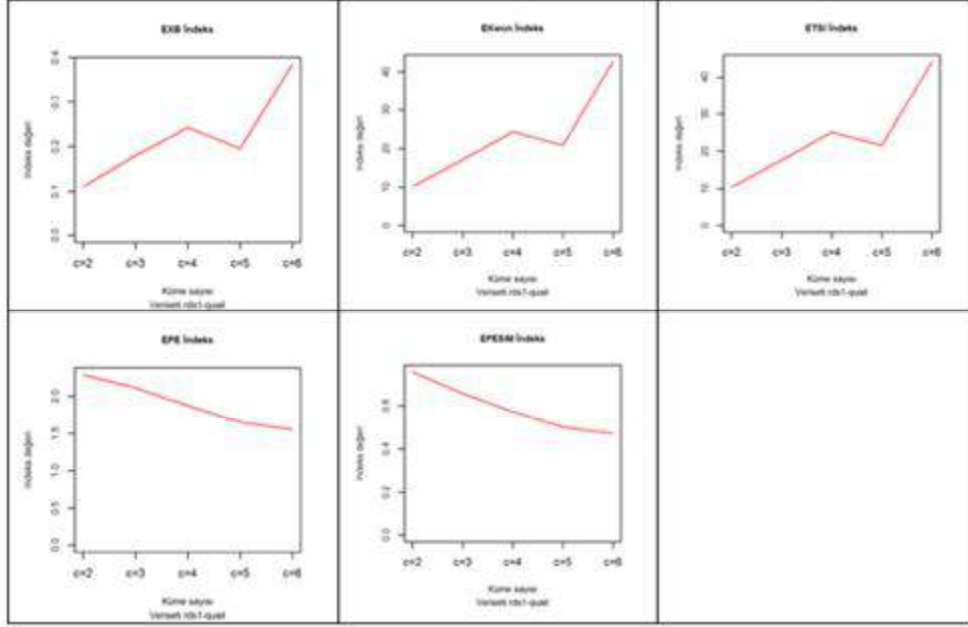
RDS2-Quail veri setinden hesaplanan uzatılmış indeks değerleri Çizelge 4.125'te gösterilmektedir. Ayrıca Çizelge 4.126 RDS2-Quail veri seti için OBCO algoritması başarımlı değerleri görülmektedir. İki Çizelge 'de (4.125 ve 4.126) incelenen sonuçlar genel anlamda küme sayısını 2 (c=2) olarak işaret etmektedirler.

Çizelge 4.126. RDS2-Quail veri seti için OBCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	25	0.110	34566.796
$c=3$	65	0.500	10348.926
$c=4$	43	0.450	7521.439
$c=5$	66	1.020	3927.341
$c=6$	126	2.570	3337.075



Şekil 4.79. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımları (RDS2-Quail)



Şekil 4.80. OBCO algoritması uzatılmış indeks değerleri (RDS2-Quail)

4.4.2.3. Gerçek Veri Seti 3 (RDS3-Wine)

RDS3-Wine veri setinden hesaplanan genelleştirilmiş bulanık indeks değerleri Çizelge 4.127’de gösterilmektedir. Elde edilen değerler incelendiğinde tüm indeks tahminleri 2 ($c=2$)’ye isabet etmektedirler. Genel anlamda yapılan tahminlerin gerçek küme sayısına ($c=3$) yakın olması durumundan dolayı beklenen optimal basarım olmasa da ona yakın bir basarım gösterdiğini söylemek mümkündür.

Çizelge 4.127. RDS3-Wine veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.068	12.354	12.604	0.086	0.899	0.812
c=3	0.183	34.083	34.798	0.136	0.889	0.732
c=4	0.117	22.908	23.593	0.155	0.882	0.720
c=5	0.146	29.192	29.998	0.167	0.885	0.696
c=6	0.185	38.814	39.987	0.168	0.889	0.709
c=7	0.106	26.946	28.303	0.163	0.896	0.739

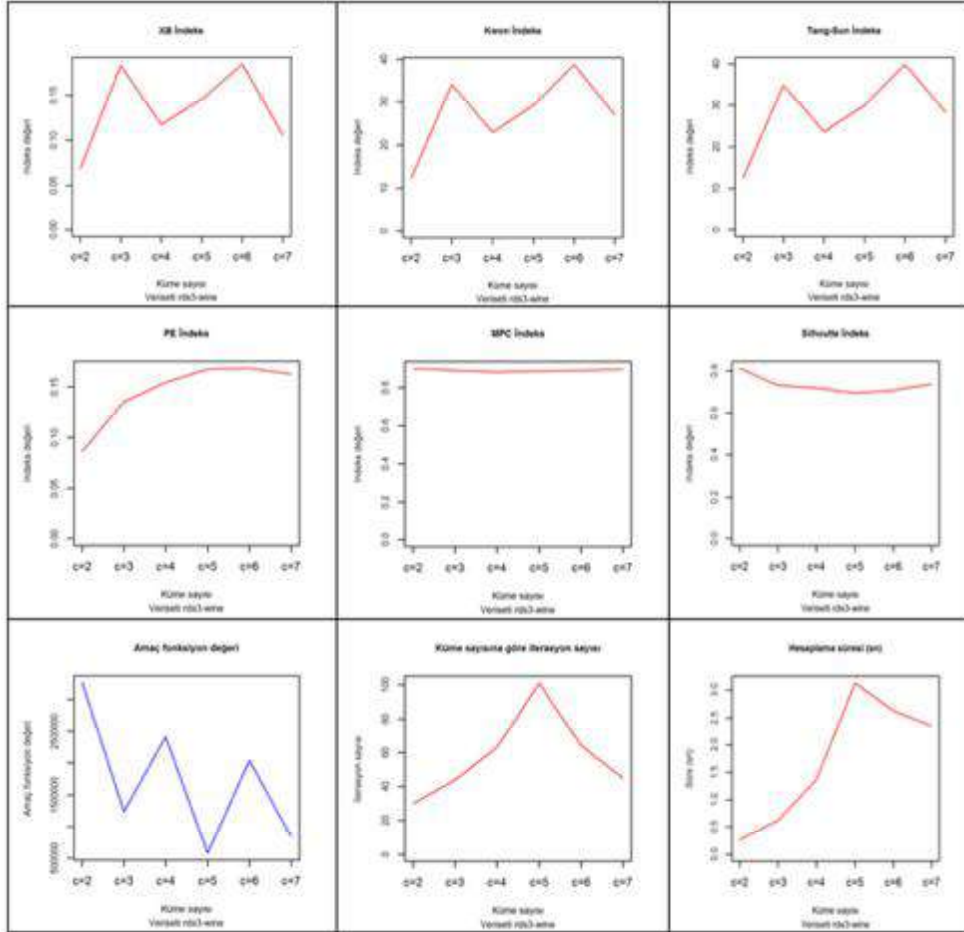
Çizelge 4.128. RDS3-Wine veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.089	16.032	16.282	3.985	1.263
c=3	0.256	46.994	47.709	3.639	1.114
c=4	0.163	31.126	31.811	2.992	0.906
c=5	0.212	41.043	41.849	2.811	0.848
c=6	0.269	53.766	54.940	2.460	0.741
c=7	0.147	34.373	35.730	1.939	0.583

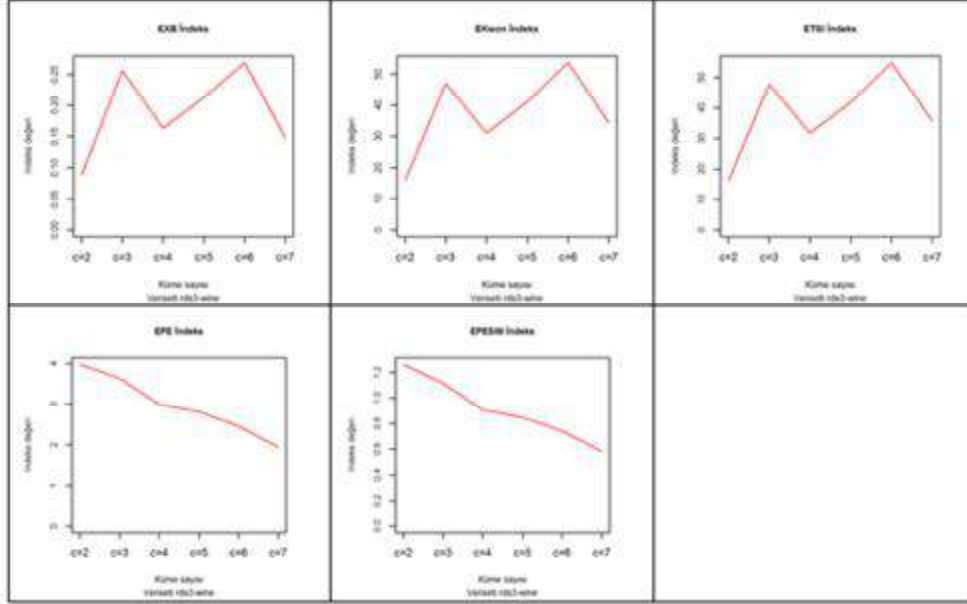
EPE ve EPESIM indeksleri tahminlerini Çizelge 4.128’de görüldüğü üzere 7 (c=7) olarak yapmaktadır. Diğer uzatılmış indeks değerleri ise tahminlerini genelleştirilmiş bulanık indeks değerlerinde olduğu gibi 2 (c=2) olarak göstermektedirler.

Çizelge 4.129. RDS3-Wine veri seti için OBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	30	0.260	3262665.511
c=3	44	0.630	1240842.030
c=4	63	1.370	2414380.480
c=5	101	3.130	581867.799
c=6	64	2.630	2037121.569
c=7	45	2.350	843958.744



Şekil 4.81. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS3-Wine)



Şekil 4.82. OBCO algoritması uzatılmış indeks değerleri (RDS3-Wine)

4.4.2.4. Gerçek Veri Seti 4 (RDS4-Glass)

RDS4-Glass gerçek veri setinden hesaplanan genelleştirilmiş bulanık indeks değerleri Çizelge 4.130'da gösterilmektedir. Değerler incelendiğinde tüm tahminler 2 ($c=2$)'ye isabet etmektedir Gerçek küme sayısı 6 ($c=6$) olması, herhangi bir indeksin bu kümeye isabet etmemesi ve son olarak yapılan tahminlerin gerçek küme sayısına yakın tahmin bile vermemesi durumu genel anlamda uygulanan algoritmanın iyi bir basarım göstermediğini ifade etmektedir.

Çizelge 4.130. RDS4-Glass veri setinde OBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	6.046	1.29e+02	676.243	0.460	0.390	0.372
c=3	122.635	2.62e+03	1388.145	0.981	0.120	0.122
c=4	42.804	9.21e+02	1140.569	0.927	0.297	0.205
c=5	1592.960	3.43e+04	1307.512	1.000	0.186	0.155
c=6	2439.159	5.26e+04	1320.656	1.000	0.159	-0.022
c=7	4822.692	1.04e+05	1338.805	1.000	0.140	-0.073

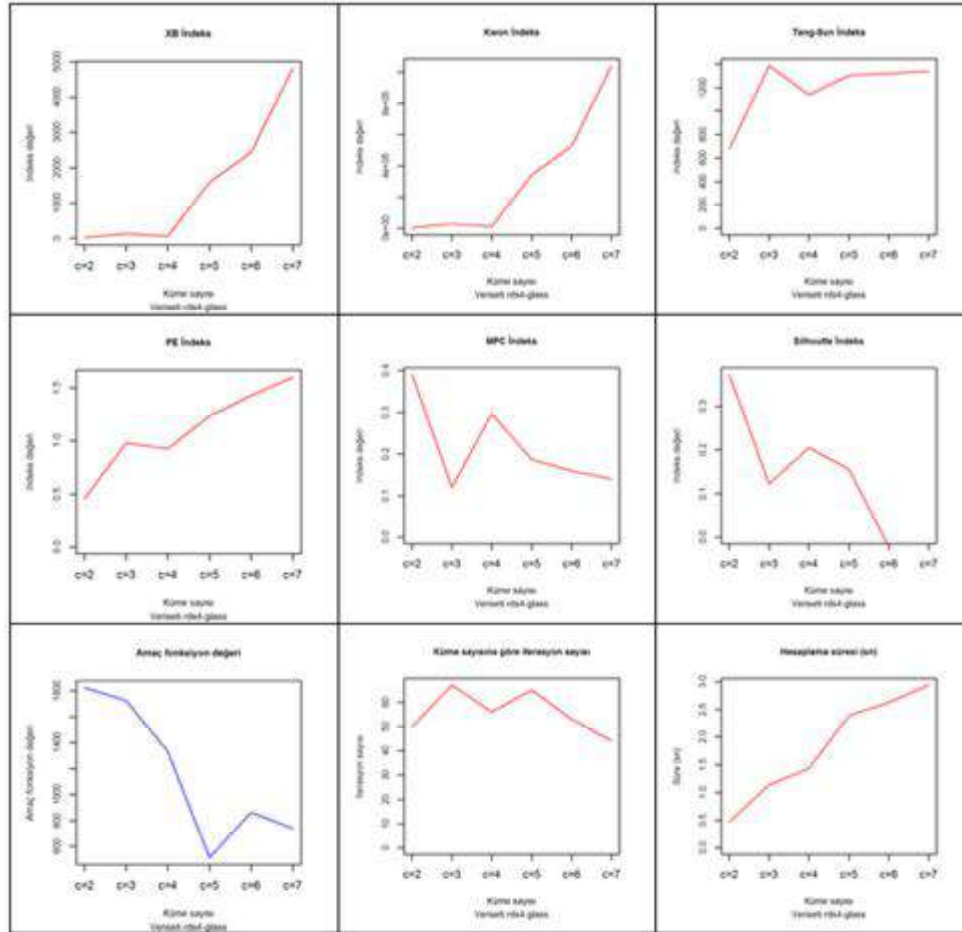
Çizelge 4.131’de diğer gerçek veri setlerinden alınan değerlerden biraz daha farklı bir sonuç ortaya çıkmaktadır. Iris, Quail ve Wine veri setlerinden alınan sonuçlarda EPE ve EPESIM indeksleri birbirine benzer yapılar oluştursa da buradaki değerlendirme aksi bir tablo ortaya koymaktadır. Burada gerçek küme sayısına (c=6) net tahmin yapılmasa da en yakın tahmini veren indeks EPESIM olarak görülmektedir.

Çizelge 4.131. RDS4-Glass veri setinde OBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

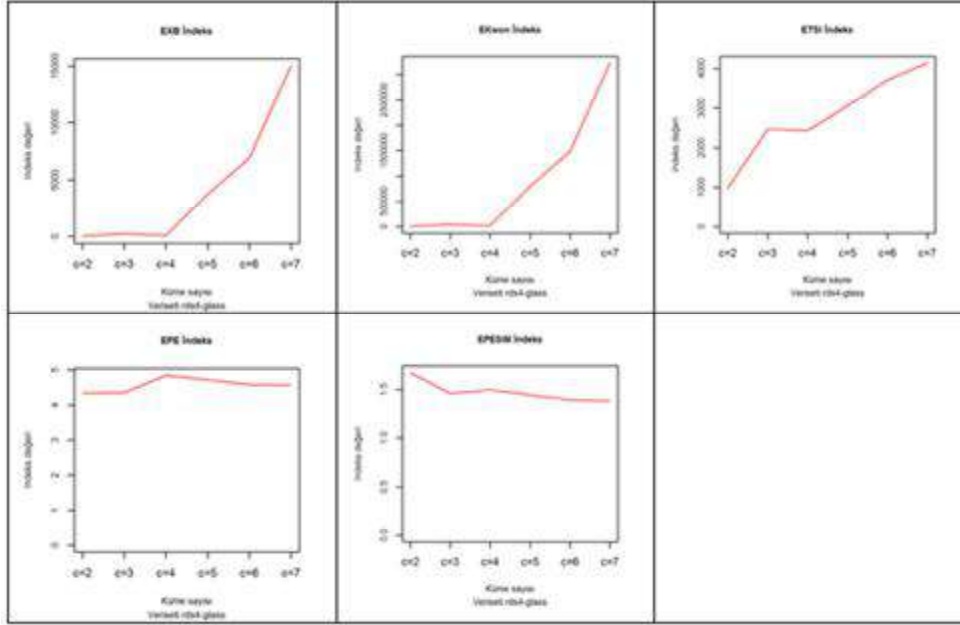
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	8.811	1.88e+02	985.412	4.347	1.675
c=3	218.801	4.68e+03	2476.577	4.358	1.460
c=4	91.192	1.95e+03	2419.126	4.844	1.499
c=5	3748.422	8.04e+04	3059.330	4.711	1.447
c=6	6911.207	1.48e+05	3716.265	4.579	1.397
c=7	15054.043	3.23e+05	4147.821	4.565	1.388

Çizelge 4.132. RDS4-Glass veri seti için OBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	50	0.470	1823.801
c=3	67	1.140	1724.951
c=4	56	1.430	1332.536
c=5	65	2.390	512.822
c=6	53	2.630	863.677
c=7	44	2.940	738.977



Şekil 4.83. OBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS4-Glass)



Şekil 4.84. OBCO algoritması uzatılmış indeks değerleri (RDS4-Glass)

4.5. İOCO Algoritması

4.5.1. Sentetik Veri Setleri

4.5.1.1. Sentetik Veri Seti 1 (SDSC)

Çizelge 4.133 incelendiğinde genelleştirilmiş bulanık indekslerinin değerleri verilmektedir. Ek 5.1 ve Ek 5.1 (Devamı)'deki küme yapılarına bakıldığında ortaya çıkan yapılar üzerinde küme sayısı 2 ($c=2$) ve 4 ($c=4$) dışında net bir ayırım (separation) durumu gözlenmemektedir. Ek 2.1'de ki OCO kümeleme algoritmasının küme sayısının değişimine göre oluşturduğu yapılarda kümelerin optimal ayrımları yaptıkları ancak burada itme teriminin de eklenmesiyle tamimiyle farklı sonuç ve yapılar ortaya çıkmaktadır. Bu durum Çizelge 4.133'deki değerler üzerindeki farklı değerlerde görülmektedir ki uygulanan kümeleme algoritmasının küme sayısı üzerinde istenilen hesaplamaları yapamamaktadır. Ancak hesaplanan küme sayılarından genel anlamda tahminler 3 ($c=3$) üzerine olmaktadır. Sonuç itibariyle genel indekslerin tahminin gerçek küme sayısına ($c=4$)

yakın tahminlerde bulunması ve çok uzak küme tahminlerinde bulunulmaması üzerinden algoritmanın bir basarımı olduğunu söylemek mümkündür.

Çizelge 4.133. SDSC veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

C	V _{XB}	V _{Kwon}	V _{TS}	V _{PE}	V _{MPC}	V _{Fsil}
c=2	0.361	289.031	241.253	0.609	0.160	0.470
c=3	0.135	108.111	99.620	0.794	0.319	0.614
c=4	1.000	0.000	3405.362	1.000	0.195	0.000
c=5	1.178	9.430	2281.121	1.000	0.001	0.000
c=6	1.915	1.532	4533.733	1.000	0.002	0.000
c=7	2.965	2.372	2046.632	1.000	0.002	0.000
c=8	7.898	6.319	1655.137	1.000	0.001	0.000

Çizelge 4.134'te küme sayısı değişimine göre uzatılmış indeks değerleri verilmektedir. İndeksler genel anlamda farklı küme sayılarını tahmin etmektedir.

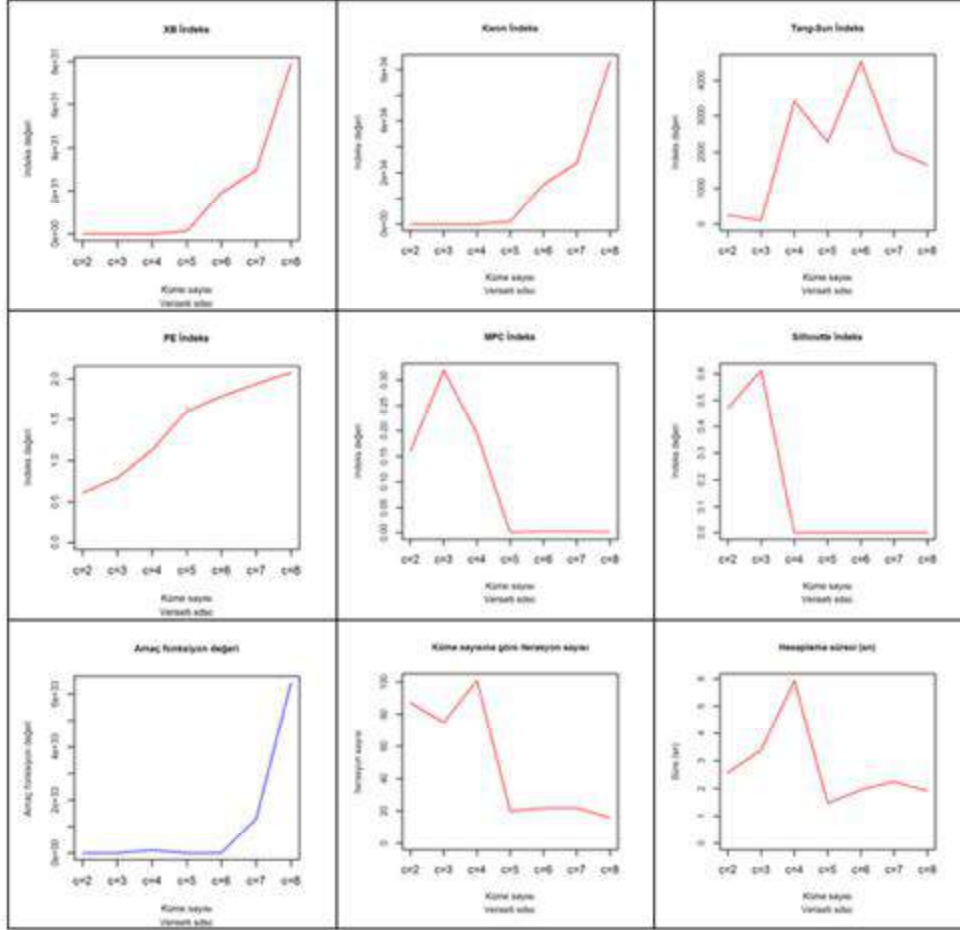
Çizelge 4.134. SDSC veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V _{EXB}	V _{EKwon}	V _{ETS}	V _{EPE}	V _{EPESIM}
c=2	0.578	462.825	386.193	18.948	5.934
c=3	0.199	159.893	147.245	15.330	4.648
c=4	1.000	1e+10	11687.716	10.036	3.038
c=5	4.104	3.283	7943.769	13.928	4.216
c=6	7.160	5.728	16946.511	9.936	3.005
c=7	1.321	1.057	9120.947	12.996	3.916
c=8	3.859	3.087	8086.870	12.880	3.877

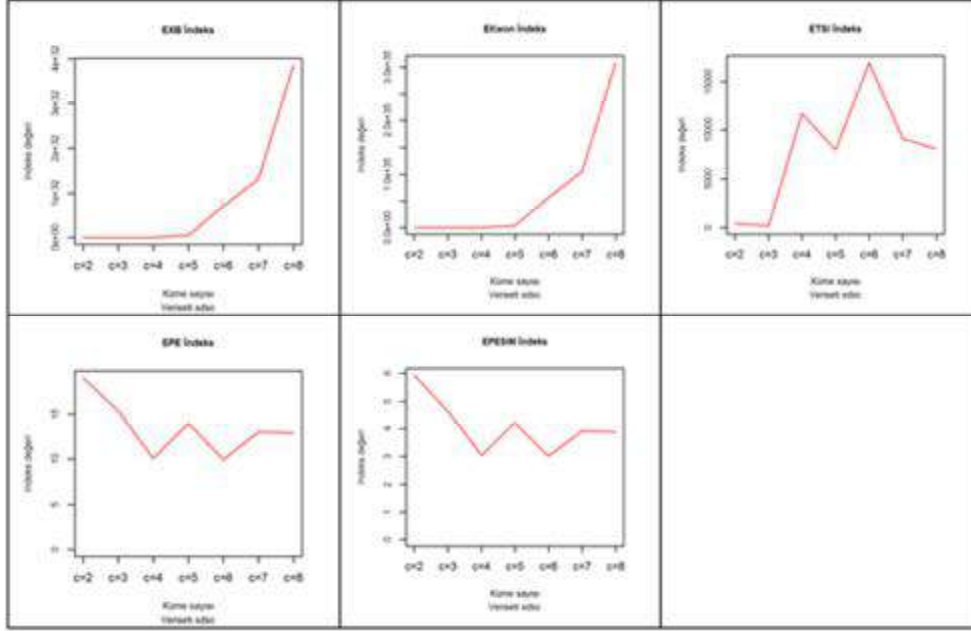
Çizelge 4.135 incelendiğinde uygulanan kümeleme algoritmasının başarımlarını sonuçları görülmektedir. Şekil 4.85 de iterasyon sayıları ve hesaplama süresi üzerinde değerler gerçek küme sayısına ulaştıktan sonra yüksek oranda bir azalma eğilimi göstermektedir. Başarımlar üzerinde minimum değerler farklı küme sayılarında; iterasyon sayısı 8 (c=8) ve hesaplama süresi 5 (c=5) olarak isabet etmektedir.

Çizelge 4.135. SDSC veri seti için İOCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	87	2.570	1719.666
c=3	75	3.380	1371.185
c=4	101	5.940	9.985
c=5	20	1.470	3.314
c=6	22	1.950	4.478
c=7	22	2.250	1.298
c=8	16	1.910	6.417



Şekil 4.85. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSC)



Şekil 4.86. İOCO algoritması uzatılmış indeks değerleri (SDSC)

4.5.1.2. Sentetik Veri Seti 2 (SDSCN)

Çizelge 4.136 incelendiğinde elde edilen U değerlerinden hesaplanan genelleştirilmiş indeks sonuçlarına göre uygun küme sayısını genel anlamda 4 ($c=4$) olarak göstermektedirler. V_{XB} küme sayısını 3 ($c=3$) olarak tahmin ederken, V_{Kwon} hem 5 ($c=5$) hem de 6 ($c=6$)'da minimum değer tahminlerinde bulunmaktadır. Son olarak V_{PE} indeksi tahminini 2 ($c=2$) olarak yapmaktadır Genel anlamda indekslerin veri seti üzerinde gerçek küme sayısına tam isabet ve diğer indekslerin de yakın küme sayısı tahminlerinde bulunması bu veri için uygulanan kümeleme algoritmasının iyi bir başarımlı gösterdiğini göstermektedir.

Çizelge 4.136. SDSCN veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	<i>V_{XB}</i>	<i>V_{Kwon}</i>	<i>V_{TS}</i>	<i>V_{PE}</i>	<i>V_{MPC}</i>	<i>V_{Fsil}</i>
c=2	0.499	409.175	322.218	0.629	0.124	0.458
c=3	0.143	117.854	108.217	0.826	0.286	0.595
c=4	0.063	52.070	50.306	0.865	0.407	0.778
c=5	1.000	0.000	4834.547	1.000	0.006	0.000
c=6	1.000	0.000	9746.588	1.000	0.002	0.000
c=7	1.076	8.825	3045.807	1.000	0.001	0.000
c=8	7.291	5.979	2917.909	1.000	0.000	0.000

Uzatılmış indeks değerlerinin sonuçları arasında bir belirsizlik söz konusudur. Çünkü tahminler farklı sayıda küme tahminlerini göstermektedirler. Burada EXB ve ETS indeksleri tahminlerini gerçek küme sayısı, yani 4 (c=4) olarak gösterirken EPE ve EPESIM indeksleri tahminlerini 6 (c=6) olarak yapmaktadır. Bu yüzden iyi bir başarımlı olup olmadığı konusunda uzatılmış indeksler üzerinden net bir yorum yapmak mümkün değildir.

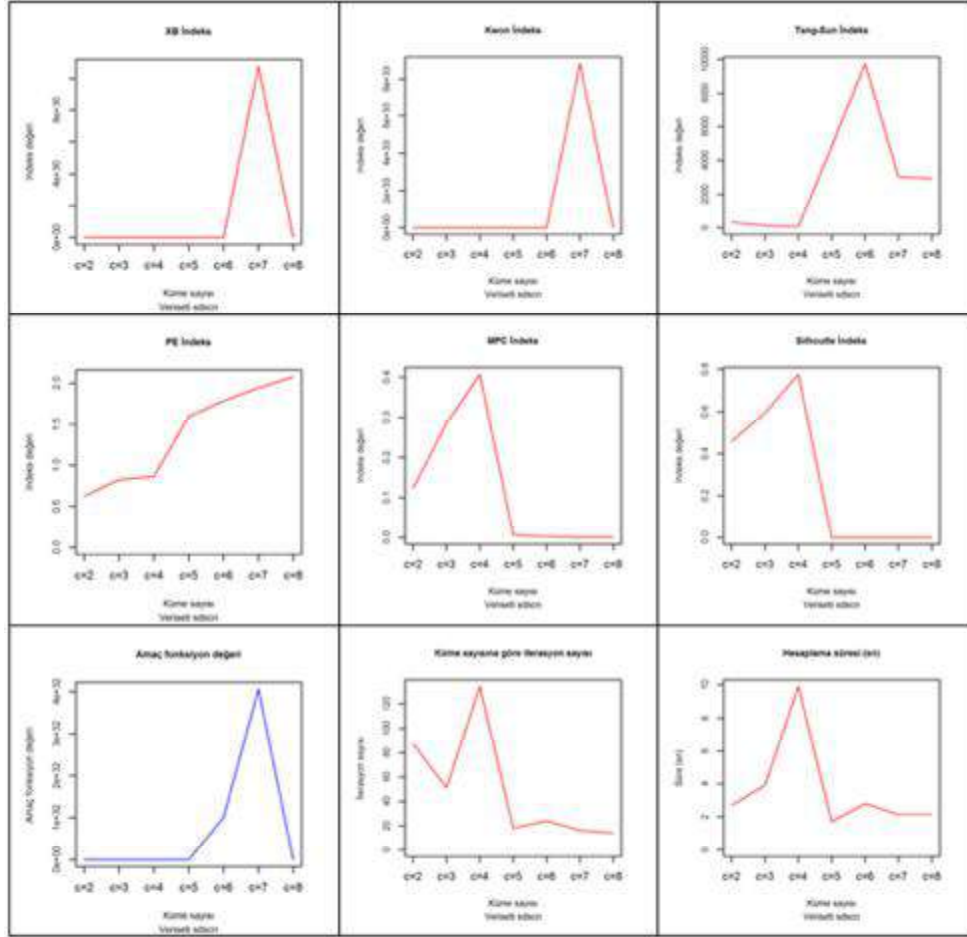
Çizelge 4.137. SDSCN veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	0.778	638.180	502.446	19.639	6.172
c=3	0.213	175.133	160.729	16.273	4.936
c=4	0.083	68.451	66.079	11.534	3.477
c=5	1.000	1e+10	15756.636	11.249	3.410
c=6	1.000	1e+10	33897.313	6.608	2.006
c=7	4.586	3.760	12978.700	11.847	3.572
c=8	3.400	2.788	13608.019	11.523	3.470

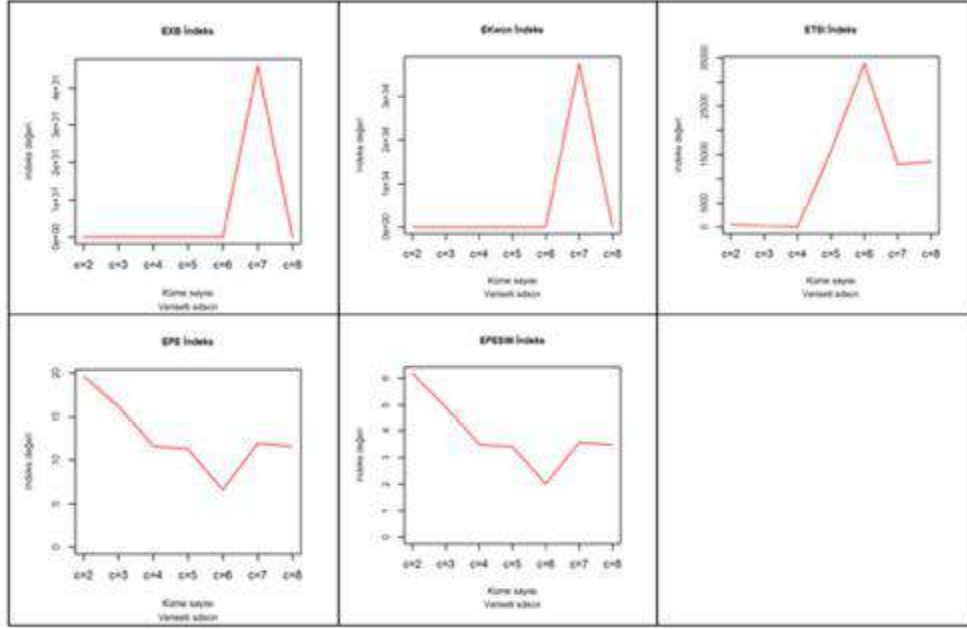
Çizelge 4.138’de İOCO algoritmasının veri seti üzerindeki başarımı gösterilmektedir. Minimum tahminler farklı küme sayılarını işaret etmektedir. Burada da SDSC (Şekil 4.85)’ye benzer bir gerçek küme sayısından (c=4) sonraki kırılma noktası göze çarpmaktadır ki hem iterasyon sayısı hem de hesaplama süresi işaret edilen küme sayılarında çok daha kısa surede ve çok daha az iterasyon ile hesaplanabilmektedir.

Çizelge 4.138. SDSCN veri seti için İOCO algoritması başarımı

<i>C</i>	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	87	2.700	1801.547
c=3	51	3.920	1507.937
c=4	134	9.900	996.099
c=5	18	1.730	1.167
c=6	24	2.800	1.014
c=7	16	2.140	4.056
c=8	14	2.140	3.941



Şekil 4.87. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSCN)



Şekil 4.88. İOCO algoritması uzatılmış indeks değerleri (SDSCN)

4.5.1.3. Sentetik Veri Seti 3 (SDSE)

Elips şekilli gürültüsüz sentetik veri setinden elde edilen genelleştirilmiş bulanık indeks değerlerine göre genel tahmin Çizelge 4.139’da da görüldüğü üzere 4 (c=4)’e isabet etmektedir. Burada V_{PE} ve V_{Kwon} indeks tahminlerinin dışında genel tahminlerin gerçek küme sayısını (c=4) göstermesi uygulanan kümeleme algoritmasının iyi bir başarımlı gösterdiğini ifade etmektedir.

Çizelge 4.139. SDSE veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.586	468.886	355.512	0.634	0.115	0.517
c=3	0.196	156.973	140.161	0.890	0.220	0.597
c=4	0.072	58.459	56.171	0.938	0.346	0.619
c=5	4.129	3.303	4071.647	1.000	0.003	0.000
c=6	1.782	1.425	2108.626	1.000	0.001	0.000
c=7	4.176	3.341	1801.839	1.000	0.003	0.000
c=8	1.000	0.000	1528.830	1.000	0.020	0.000

Çizelge 4.140'da hesaplanan uzatılmış indeks değerlerine göre optimal küme sayısının V_{EPE} , V_{EPESIM} ve V_{EKwon} indeksleri ile 5 (c=5), V_{EXB} ve V_{ETS} indeksleri tarafından ise 4 (c=4) olarak isabet etmiştir.

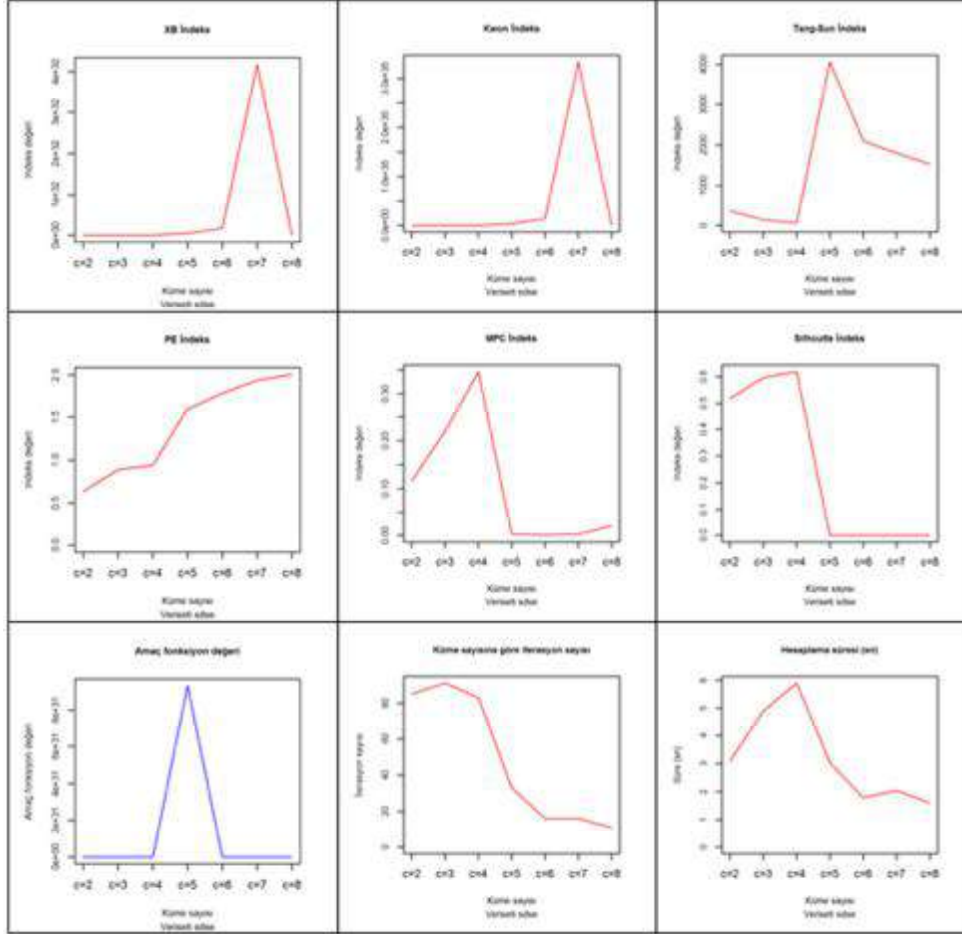
Çizelge 4.140. SDSE veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.868	694.673	526.614	19.086	6.003
c=3	0.307	246.367	219.882	17.038	5.174
c=4	0.112	90.442	86.803	12.834	3.871
c=5	1.280	1.024	12624.220	12.276	3.720
c=6	7.206	5.765	8527.473	14.526	4.383
c=7	1.906	1.525	8224.918	15.123	4.554
c=8	1.000	1e+10	10304.920	13.669	4.110

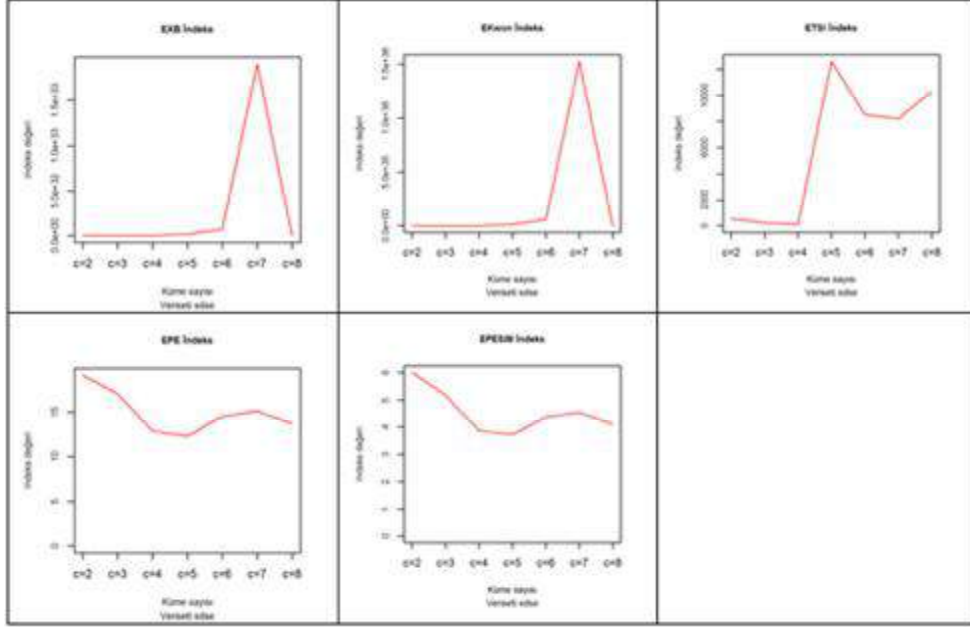
Çizelge 4.141’de uygulanan kümeleme algoritmasının başarımları gösterilmektedir. Başarımlardaki sonuçlara bakıldığında küme sayısı 8 ($c=8$)’de iterasyon sayısı ve hesaplama süresini minimum değerlerde yani gerçek küme sayısına ($c=4$) göre daha iyi bir başarımları sergilemektedir.

Çizelge 4.141. SDSE veri seti için İOCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	85	3.080	1670.201
c=3	91	4.900	1591.426
c=4	83	5.900	1243.898
c=5	33	3.040	9.311
c=6	16	1.780	1.403
c=7	16	2.030	1.368
c=8	11	1.600	3.170



Şekil 4.89. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSE)



Şekil 4.90. İOCO algoritması uzatılmış indeks değerleri (SDSE)

4.5.1.4. Sentetik Veri Seti 4 (SDSEN)

Elips şekilli gürültülü sentetik veri setinden elde edilen genelleştirilmiş bulanık indeks değerlerine göre genel tahmin Çizelge 4.142’de gösterilmektedir. Elips şekilli gürültüsüz veri setinde olduğu gibi tahminler 4 ($c=4$)’e isabet etmektedir. Burada V_{PE} ve V_{Kwon} indeks tahminlerinin dışında genel tahminlerin gerçek küme sayısını ($c=4$) göstermesi uygulanan kümeleme algoritmasının iyi bir başarımla gösterdiğini ifade etmektedir.

Çizelge 4.142. SDSEN veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.568	466.000	355.849	0.633	0.116	0.495
c=3	0.152	124.821	113.971	0.844	0.268	0.614
c=4	0.062	51.119	49.338	0.882	0.393	0.707
c=5	1.253	1.028	10136.904	1.000	0.003	0.000
c=6	9.526	7.811	9243.656	1.000	0.003	0.000
c=7	1.000	0.000	1758.215	1.000	0.003	0.000
c=8	4.842	3.970	4656.031	1.000	0.001	0.000

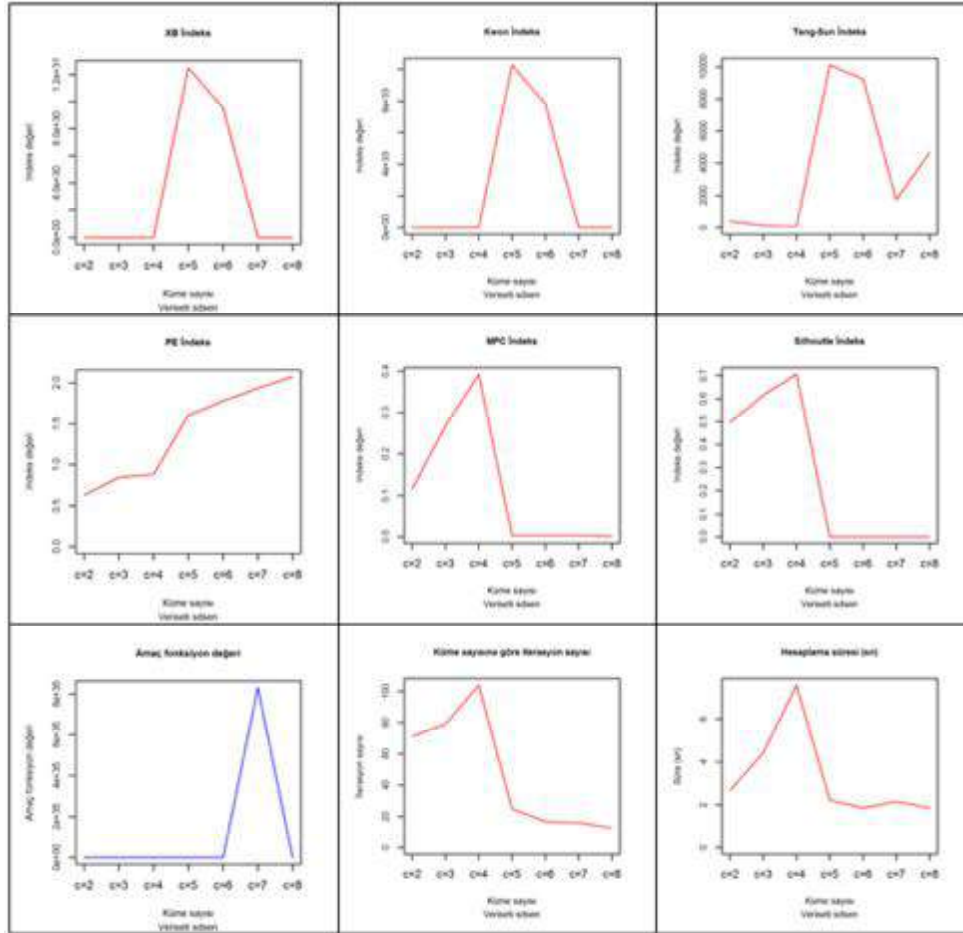
Çizelge 4.140'da hesaplanan uzatılmış indeks değerlerine göre optimal küme sayısının V_{EPE} , V_{EPESIM} ve V_{EKwon} indeksleri ile 5 ($c=5$), V_{EXB} ve V_{ETS} indeksleri tarafından ise 4 ($c=4$) olarak isabet etmiştir. Bu durumda V_{EXB} ve V_{ETS} dışındaki indeksler bu veri seti üzerinde etkin değildirler.

Çizelge 4.143. SDSEN veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

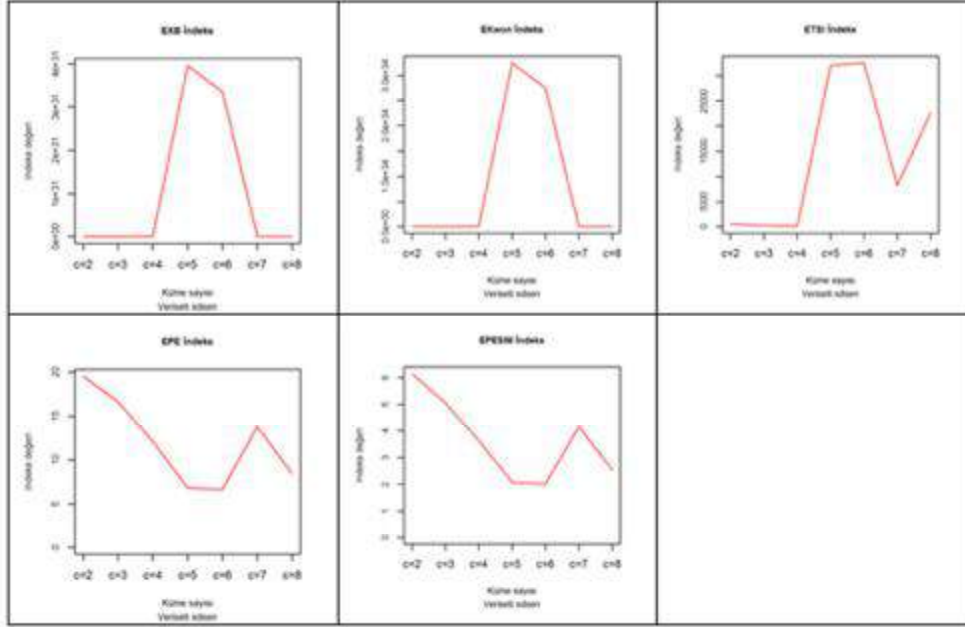
<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.844	692.427	528.662	19.552	6.148
c=3	0.228	187.699	171.300	16.672	5.058
c=4	0.087	71.892	69.320	12.054	3.634
c=5	3.970	3.255	32100.879	6.813	2.079
c=6	3.361	2.756	32614.811	6.633	2.013
c=7	1.000	1e+10	8281.637	13.790	4.154
c=8	2.349	1.926	22592.785	8.411	2.536

Çizelge 4.144. SDCEN veri seti için İOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	71	2.670	1741.729
c=3	79	4.410	1520.201
c=4	104	7.570	1096.868
c=5	25	2.220	2.028
c=6	17	1.860	1.338
c=7	16	2.140	8.307
c=8	13	1.860	4.92e+14



Şekil 4.91. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDCEN)



Şekil 4.92. İOCO algoritması uzatılmış indeks değerleri (SDSEN)

4.5.1.5. Sentetik Veri Seti 5 (SDSR)

Çizelge 4.145’de kare şekilli gürültüsüz sentetik veri setinden elde edilen indeks değerlerine göre genel tahminler 4 ($c=4$) olarak tahmin edilmektedir. Burada V_{PE} ve V_{Kwon} indeks tahminlerini farklı kümeler üzerinde göstermektedir ancak genel tahminlerin gerçek küme sayısını ($c=4$) göstermesi uygulanan kümeleme algoritmasının iyi bir başarımla gösterdiğini ifade etmektedir.

Çizelge 4.145. SDSR veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.394	315.661	259.610	0.613	0.152	0.480
c=3	0.133	106.918	98.584	0.816	0.297	0.605
c=4	0.064	51.681	49.945	0.871	0.402	0.780
c=5	7.785	6.228	3838.505	1.000	0.003	0.000
c=6	1.000	0.000	1649.657	1.000	0.001	0.000
c=7	2.786	2.229	2550.541	1.000	0.001	0.000
c=8	7.295	5.836	1798.456	1.000	0.001	0.000

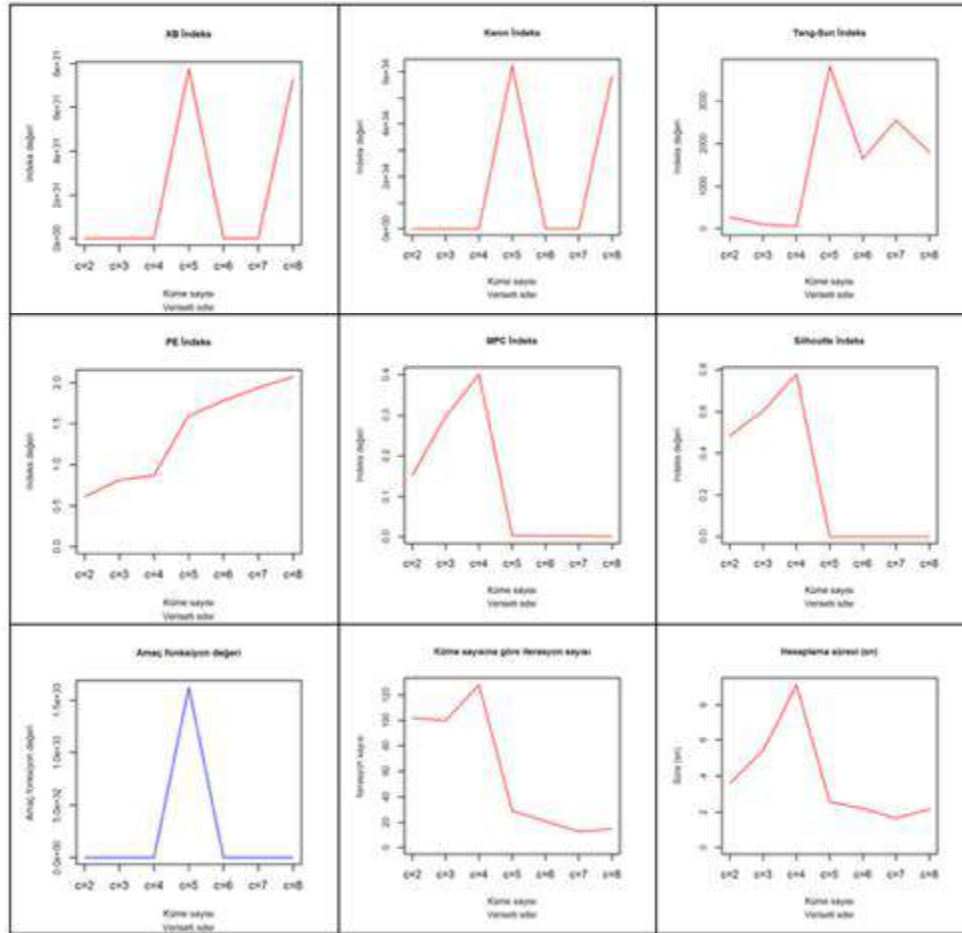
Çizelge 4.146. SDSR veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.629	503.520	413.990	18.969	5.940
c=3	0.201	160.854	148.227	15.864	4.812
c=4	0.084	67.384	65.069	11.316	3.411
c=5	2.554	2.043	12592.538	11.582	3.511
c=6	1.000	1e+10	6866.161	15.094	4.552
c=7	1.193	9.546	10922.098	12.519	3.773
c=8	3.599	2.879	8873.065	12.794	3.851

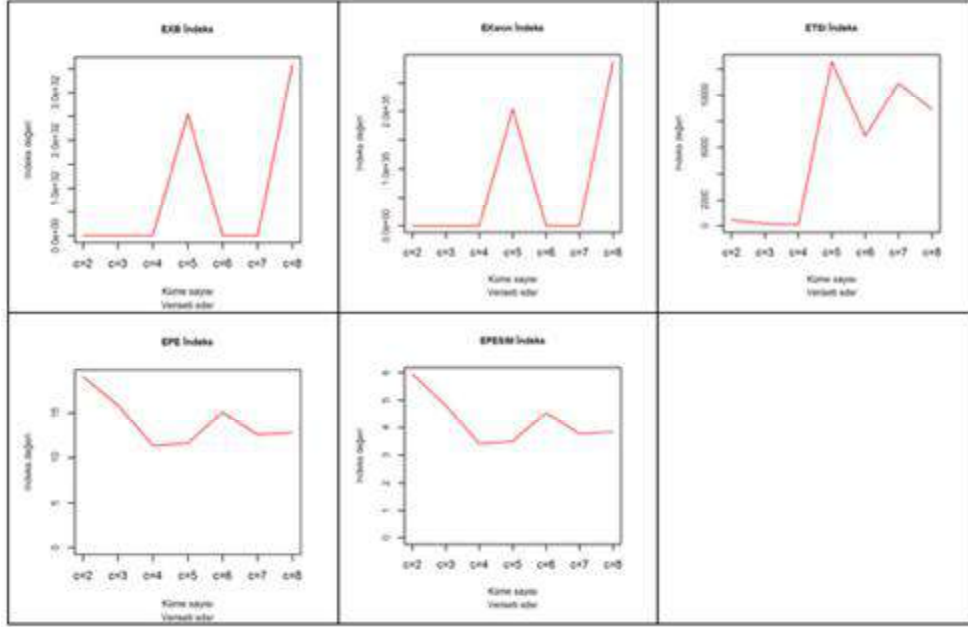
Çizelge 4.146'da hesaplanan uzatılmış indeks değerlerine göre optimal küme sayısını V_{EKwon} indeksi dışında 4 (c=4) olarak tahmin etmesi uygulanan uzatılmış indekslerin veri seti üzerinde etkinliğini ortaya koymaktadır. Ayrıca İOCO algoritması başarımını gösteren Çizelge 4.147'de işlemler 7 (c=7)'de minimum olarak hesaplanmaktadır.

Çizelge 4.147. SDSR veri seti için İOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	102	3.610	1711.491
c=3	100	5.410	1440.294
c=4	128	9.120	979.269
c=5	29	2.560	1.622
c=6	21	2.210	3.622
c=7	13	1.670	1.488
c=8	15	2.170	1.536



Şekil 4.93. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR)



Şekil 4.94. İOCO algoritması uzatılmış indeks değerleri (SDSR)

4.5.1.6. Sentetik Veri Seti 6 (SDSRN)

Çizelge 4.148’de elde edilen indeks değerleri incelendiğinde genel tahmin 3 ($c=3$)’e isabet ettiği gözlenmektedir. V_{Kwon} 3 farklı küme sayısı üzerinde minimum değeri göstermektedir Herhangi bir indeks tarafından gerçek küme sayısına ($c=4$) isabetli tahminde bulunamaması ve istenilen yakın tahminlerin az olması durumu itibariyle uygulanan algoritmanın kümelerin tespitinde iyi bir başarıyı gösterememektedir.

Çizelge 4.148. SDSRN veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.542	444.834	343.834	0.631	0.121	0.542
c=3	0.157	129.138	117.673	0.828	0.285	0.603
c=4	1.762	1.445	1850.205	1.000	0.000	0.000
c=5	5.055	4.145	3321.625	1.000	0.003	0.000
c=6	1.000	0.000	1714.313	1.000	0.002	0.000
c=7	1.000	0.000	3336.672	1.000	0.003	0.000
c=8	1.000	0.000	1971.858	1.000	0.000	0.000

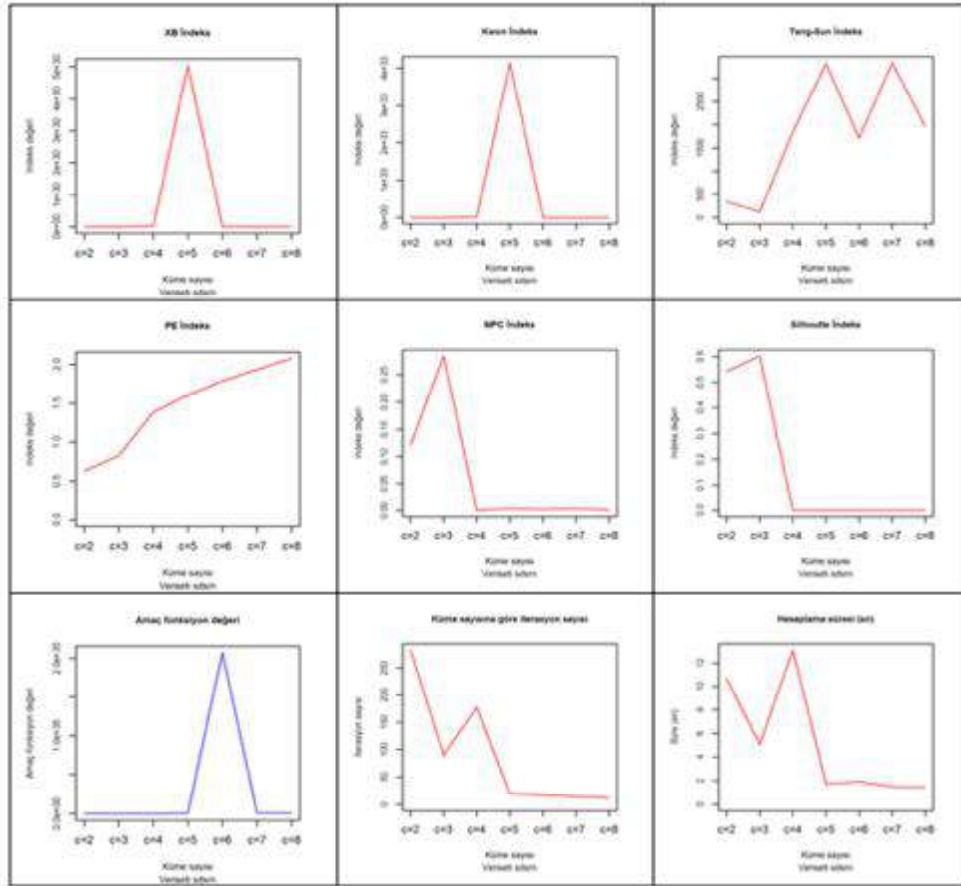
Çizelge 4.149. SDSRN veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.835	685.280	529.582	19.648	6.160
c=3	0.238	195.517	178.064	16.292	4.942
c=4	5.674	4.653	5955.697	16.606	5.046
c=5	1.684	1.381	11066.493	12.790	3.873
c=6	1.000	1e+10	7098.972	15.589	4.700
c=7	1.000	1e+10	13988.287	11.348	3.420
c=8	1.000	1e+10	9694.025	12.986	3.909

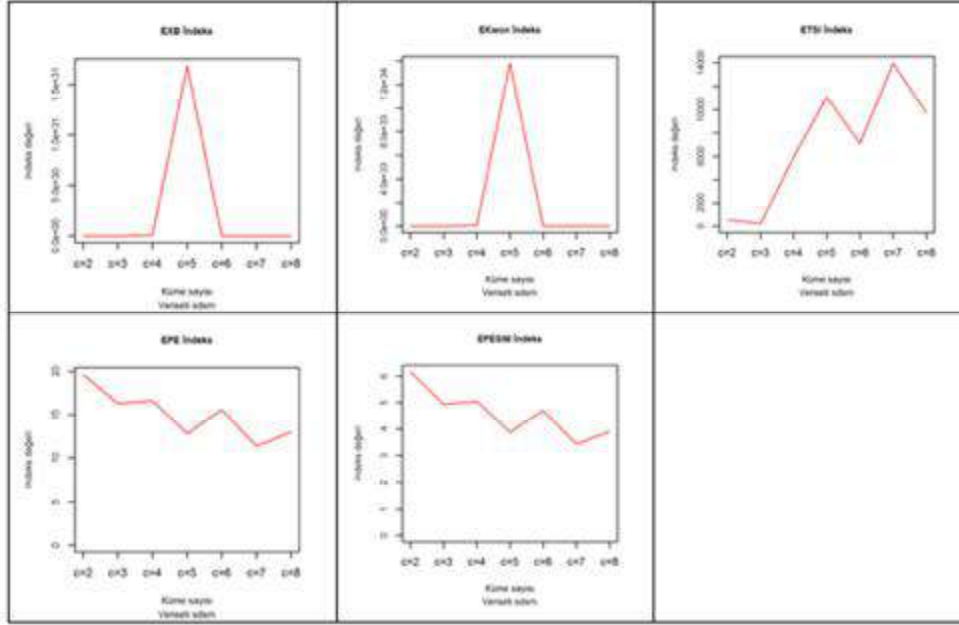
Çizelge 4.149'da hesaplanan uzatılmış indeks değerlerine göre optimal küme sayısının V_{EKwon} indeksi dışında 3 (c=3) ve 7 (c=7) olarak tahmin etmesi yani hiçbir indeks tarafından gerçek küme sayısının (c=4) işaret edilmemesi uygulanan uzatılmış indekslerin veri seti üzerinde etkinliğinin zayıf olduğunu göstermektedir. İOCO algoritması başarıımı daha kısa süre ve daha az iterasyon sayısı ile hesaplayan küme sayısını 7 (c=7) olarak Çizelge 4.150'de göstermektedir.

Çizelge 4.150. SDSRN veri seti için İOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	283	10.680	1777.612
c=3	90	5.040	1497.662
c=4	177	13.120	6.291
c=5	19	1.720	1.397
c=6	17	1.920	2.076
c=7	14	1.500	7.301
c=8	12	1.450	8.112



Şekil 4.95. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN)



Şekil 4.96. İOCO algoritması uzatılmış indeks değerleri (SDSRN)

4.5.1.7. Sentetik Veri Seti 7 (SDSX)

Çizelge 4.151'de elde edilen indeks değerleri incelendiğinde en etkin genelleştirilmiş bulanık indeksin V_{XB} olduğunu gerçek küme sayısına ($c=3$) yapmış olduğu net tahminden dolayı söylemek mümkündür. Genel olarak indekslerdeki çoğunluğun tahminlerinde 4 ($c=4$)'ü göstermesi kümeye yakınlığından dolayı algoritmanın kümelerin tespitinde iyi bir başarıyı gösterdiği anlaşılmaktadır.

Çizelge 4.151. SDSX veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	<i>V_{XB}</i>	<i>V_{Kwon}</i>	<i>V_{TS}</i>	<i>V_{PE}</i>	<i>V_{MPC}</i>	<i>V_{Fsil}</i>
c=2	2.644	2120.839	903.676	0.680	0.026	0.462
c=3	0.962	771.567	511.373	1.000	0.049	0.472
c=4	0.529	426.406	327.765	1.000	0.166	0.546
c=5	1.000	0.000	4977.732	1.000	0.002	0.000
c=6	1.000	0.000	2013.168	1.000	0.001	0.000
c=7	4.128	3.310	1716.683	1.000	0.002	0.000
c=8	1.000	0.000	1901.421	1.000	0.002	0.000

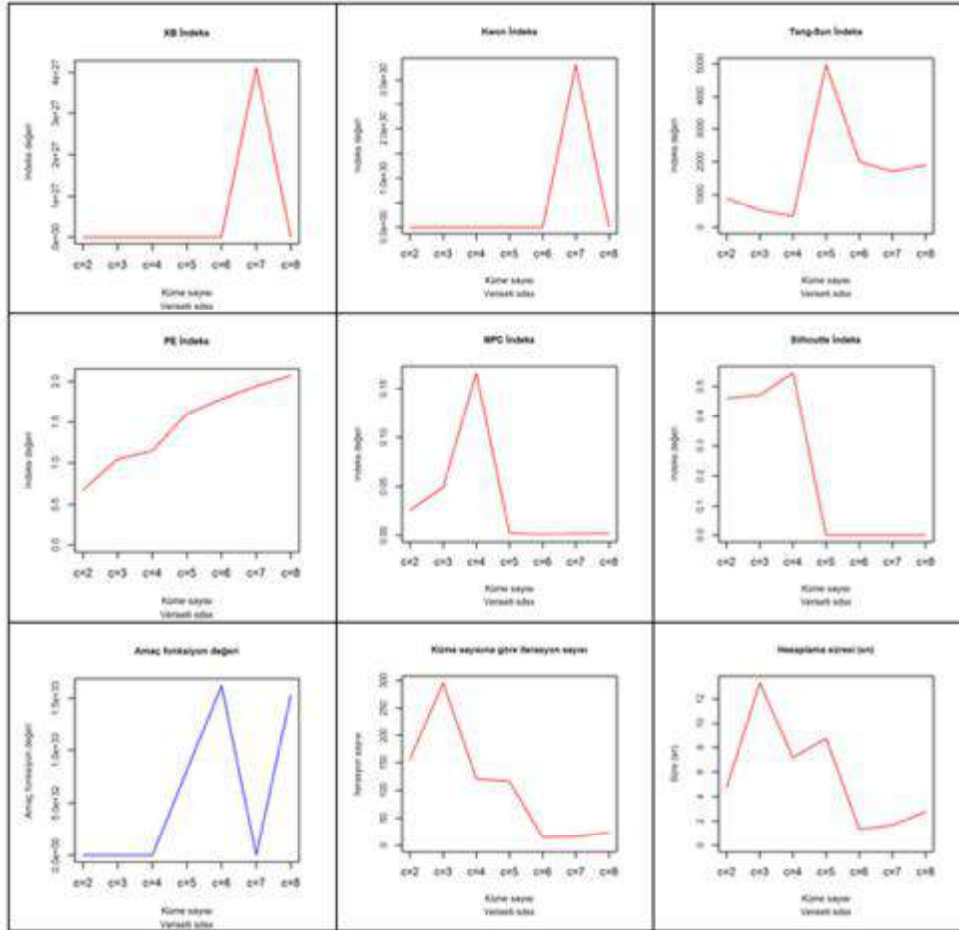
Çizelge 4.152. SDSX veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	4.177	3350.058	1427.377	17.559	5.645
c=3	1.920	1540.299	1020.728	18.094	5.533
c=4	1.114	895.747	687.956	16.255	4.910
c=5	1.000	1e+10	15433.094	12.287	3.723
c=6	1.000	1e+10	8107.744	15.857	4.782
c=7	1.934	1.551	8043.771	15.227	4.586
c=8	1.000	1e+10	9631.885	14.963	4.502

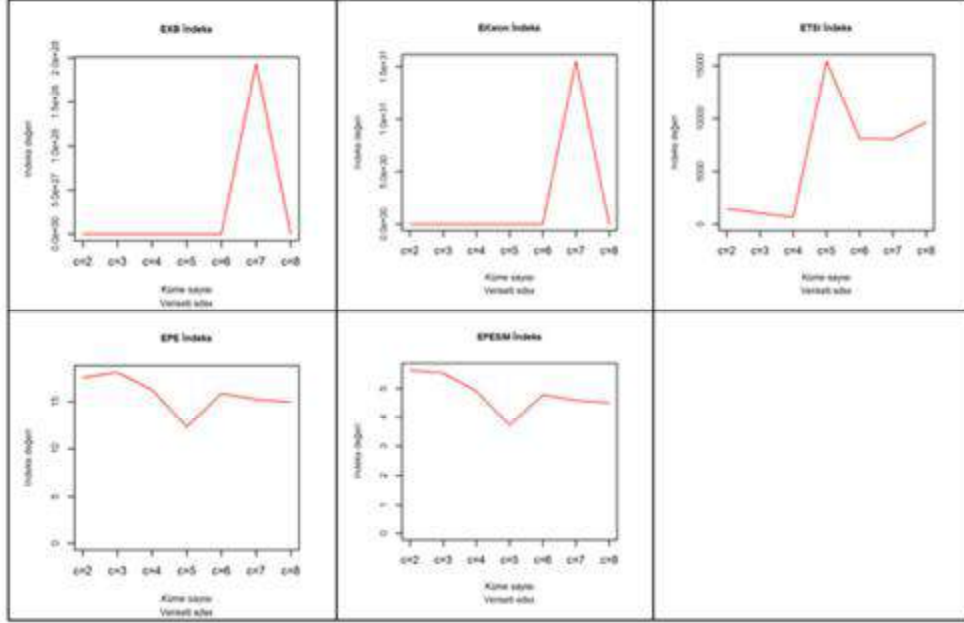
Çizelge 4.152’de uzatılmış indeks değerlerinin sonuçlarına göre genel tahmin 4 (c=4) olarak yapılmaktadır. Herhangi bir indeksin gerçek küme sayısını (c=3) tahmin göstermesi durumu uzatılmış indekslerin etkinliklerinin veri seti üzerinde çok iyi olmadığı anlamına gelmektedir.

Çizelge 4.153. SDSX veri seti için İOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	157	4.710	1732.564
c=3	296	13.310	1963.882
c=4	121	7.160	1766.792
c=5	116	8.710	8.112
c=6	15	1.330	1.622
c=7	16	1.670	1.828
c=8	23	2.750	1.527



Şekil 4.97. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX)



Şekil 4.98. İOCO algoritması uzatılmış indeks değerleri (SDSX)

4.5.1.8. Sentetik Veri Seti 8 (SDSXXN)

Çizelge 4.154 incelendiğinde yalnızca Fsil bulanık indeksinin gerçek kümeye (c=3) isabet ettiği gözlenmektedir. Kwon indeksinin 3 farklı kümede minimum değere hatta “0” değerini göstermesi durumundan bu veri seti için etkin bir indeks olmadığını söylemek mümkündür. Ek 5.8 ve Ek 5.8 (Devamı) incelendiğinde oluşan küme yapılarındaki gürültüler genel olarak sağ iki uç arasında yer almaktadır.

Çizelge 4.154. SDSXN veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	2.615	2149.602	923.921	0.680	0.027	0.468
c=3	0.829	681.449	473.978	1.000	0.056	0.494
c=4	0.269	222.356	191.900	1.000	0.159	0.474
c=5	1.000	0.000	3106.205	1.000	0.002	0.000
c=6	6.811	5.598	4140.683	1.000	0.002	0.000
c=7	1.000	0.000	6677.125	1.000	0.002	0.000
c=8	1.000	0.000	2026.285	1.000	0.001	0.000

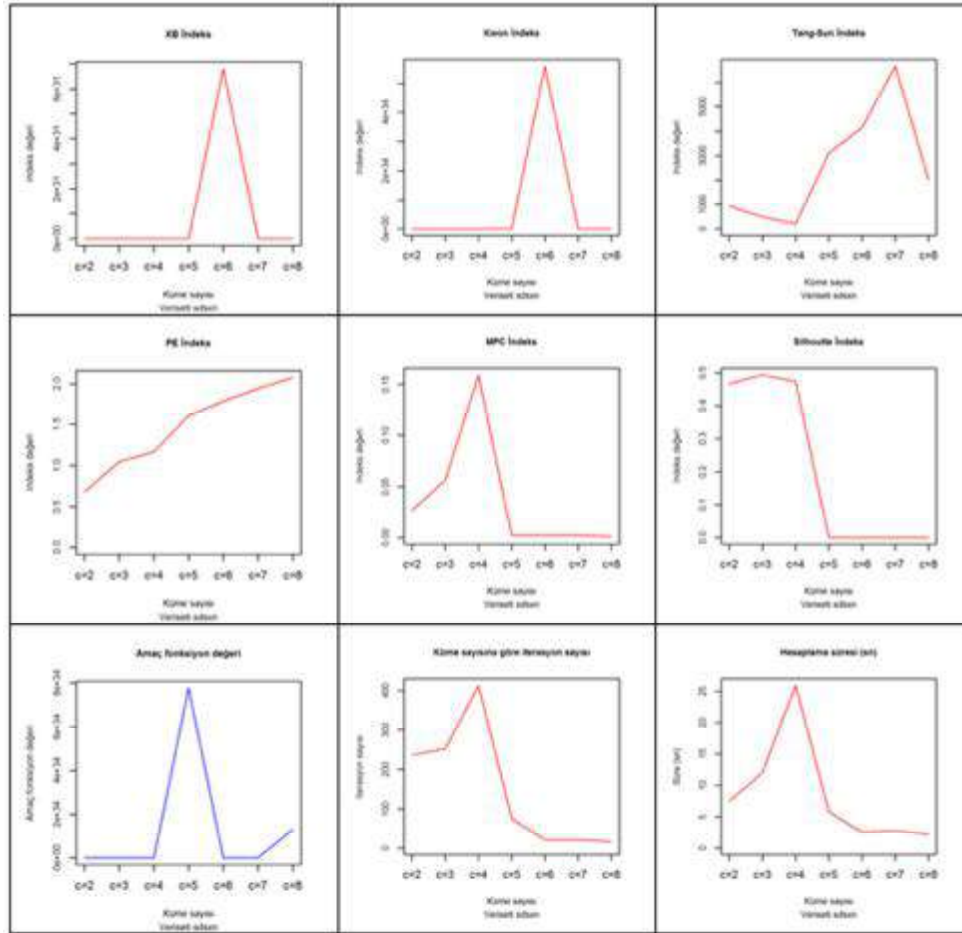
Çizelge 4.155. SDSXN veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	4.156	3.41e+02	1468.417	18.084	5.800
c=3	1.570	1.29e+02	897.542	18.567	5.672
c=4	0.632	5.20e+01	448.854	17.265	5.214
c=5	1.000	1e+10	9972.680	15.765	4.768
c=6	2.319	0.01e+01	14102.570	13.487	4.071
c=7	1.000	1e+10	24333.252	9.663	2.916
c=8	1.000	1e+10	10358.192	15.185	4.569

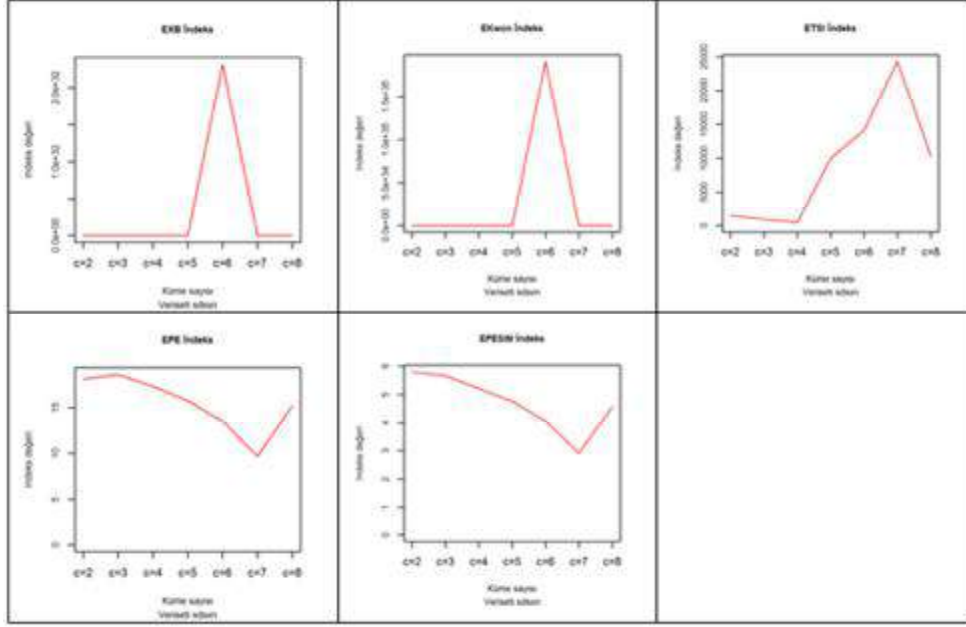
Çizelge 4.155 konkav şekilli gürültülü veri setinden elde edilen uzatılmış indeks değerleri incelendiğinde sonuçların gürültüsüz uzatılmış indeks değerlerinden çok farklı sonuçlar verdiği görülmektedir. Çizelge 4.156'daki başarımların değerleri ise minimum hesaplamayı 8 (c=8)'de göstermiştir.

Çizelge 4.156. SDSXN veri seti için İOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	237	7.440	1798.842
c=3	252	12.030	1927.741
c=4	414	25.950	2011.876
c=5	76	5.850	7.788
c=6	21	2.530	1.861
c=7	21	2.760	5.557
c=8	16	2.250	1.298



Şekil 4.99. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSXN)



Şekil 4.100. İOCO algoritması uzatılmış indeks değerleri (SDSXN)

4.5.2. Gerçek Veri Setleri

4.5.2.1. Gerçek Veri Seti 1 (RDS1-Iris)

RDS1-Iris veri setinde ikisi birbiriyle çakışan 3 doğal küme bulunduğundan herhangi bir kümeleme algoritmasının küme sayısını 2 veya 3 küme bulması başarılı bir kümeleme olduğu anlamına gelecektir. Çizelge 4.157'de V_{Kwon} dışındaki tüm indeks değerlerine göre küme sayısının 2 olduğu önerilmiştir. Bu durum İOCO algoritmasının başarılı olduğunu göstermektedir.

Çizelge 4.157. RDS1-Iris veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.049	7.635	7.681	0.273	0.703	0.847
c=3	2.623	3.934	10360.595	1.000	0.036	0.000
c=4	6.074	9.111	718.799	1.000	0.015	0.000
c=5	6.928	1.039	860.870	1.000	0.006	0.000
c=6	1.575	2.363	882.476	1.000	0.003	0.000
c=7	2.501	3.754	835.797	1.000	0.017	0.000

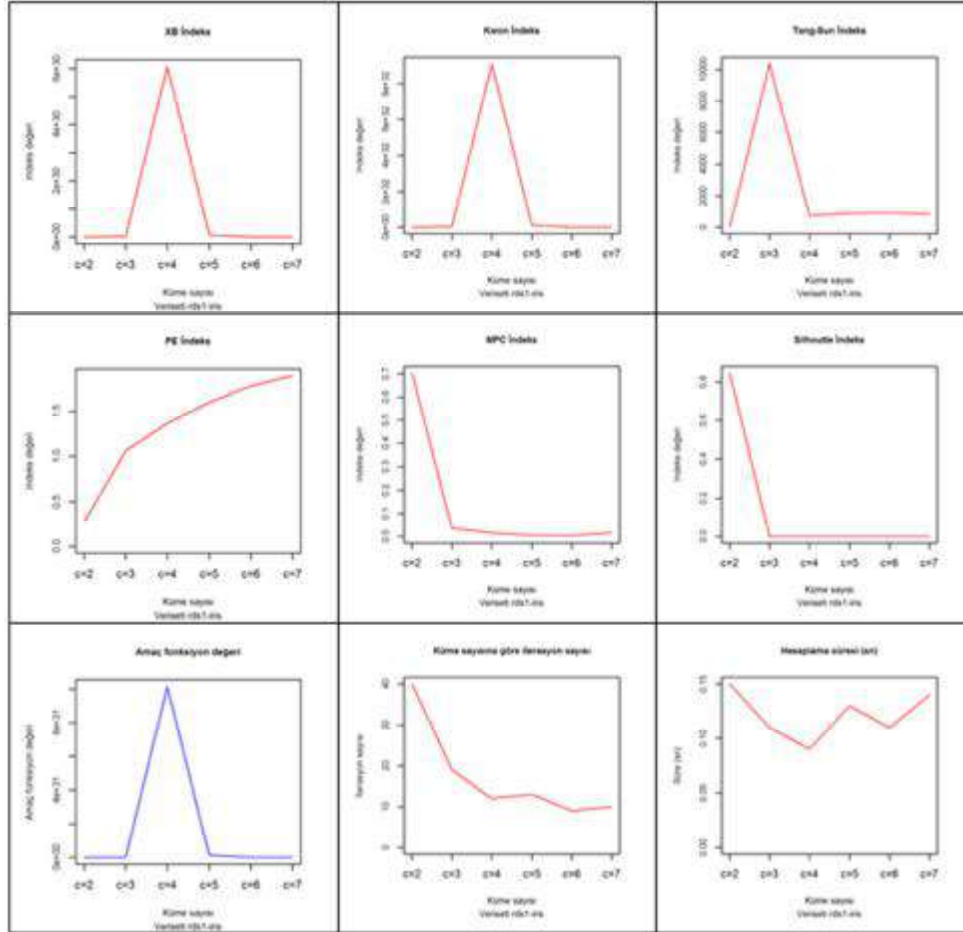
Çizelge 4.158'de görülen uzatılmış indeks değerlerinde, V_{EPE} ve V_{EPESIM} indeks değerleri uygun küme sayısını 3 (c=3) olarak uygun görürken V_{EXB} ve V_{ETS} indeksleri 2 (c=2) olarak tahmin etmişlerdir. Veri seti üzerinde etkinlik gösteremeyen V_{EKwon} indeksi dışındaki tüm indeks değerlerinin gerçek küme sayısını (c=2,c=3) isabetli bir şekilde tahmin ettiği gözlenmektedir.

Çizelge 4.158. RDS1-Iris veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

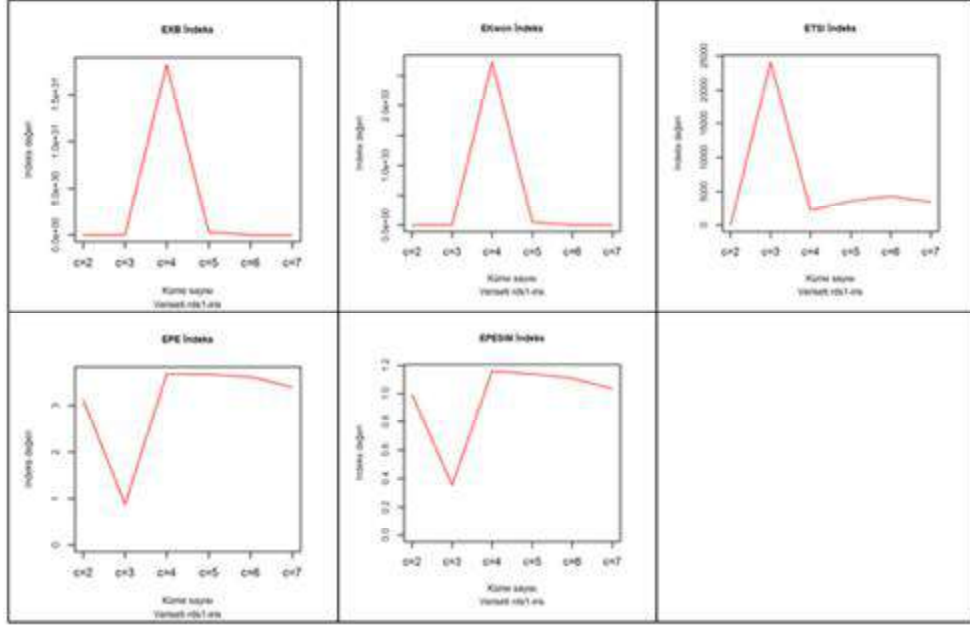
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.059	9.160	9.167	3.124	0.990
c=3	6.108	9.163	24127.29	0.887	0.352
c=4	1.822	2.733	2156.446	3.690	1.159
c=5	2.882	4.323	3580.913	3.683	1.139
c=6	7.720	1.158	4324.283	3.621	1.110
c=7	1.024	1.536	3421.060	3.402	1.034

Çizelge 4.159. RDS1-Iris veri seti için İOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	40	0.150	229.683
c=3	19	0.110	2.814
c=4	12	0.090	1.014
c=5	13	0.130	1.207
c=6	9	0.110	3.293
c=7	10	0.140	2.719



Şekil 4.101. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS1-Iris)



Şekil 4.102. İOCO algoritması uzatılmış indeks değerleri (RDS1-Iris)

4.5.2.2. Gerçek Veri Seti 2 (RDS2-Quail)

RDS2-Quail gerçek veri setinde genelleştirilmiş bulanık indeks değerleri. Çizelge 4.160'da verilmektedir. V_{XB} , V_{Kwon} ve V_{TS} indeks değerlerine göre küme sayısının 6 ($c=6$) olduğu önerilmiştir. Genel anlamda indeks sonuçları değerlendirildiğinde İOCO algoritmasının başarılı olamadığı görülmektedir.

Çizelge 4.160. RDS2-Quail veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
$c=2$	117.586	10582.961	9700.666	0.689	0.009	-0.249
$c=3$	15.566	1435.377	1420.021	0.776	0.297	0.293
$c=4$	20.079	1861.317	1824.215	1.018	0.259	0.249
$c=5$	12.000	1131.901	1127.094	1.103	0.258	0.156
$c=6$	7.400	721.472	723.061	1.248	0.234	0.251

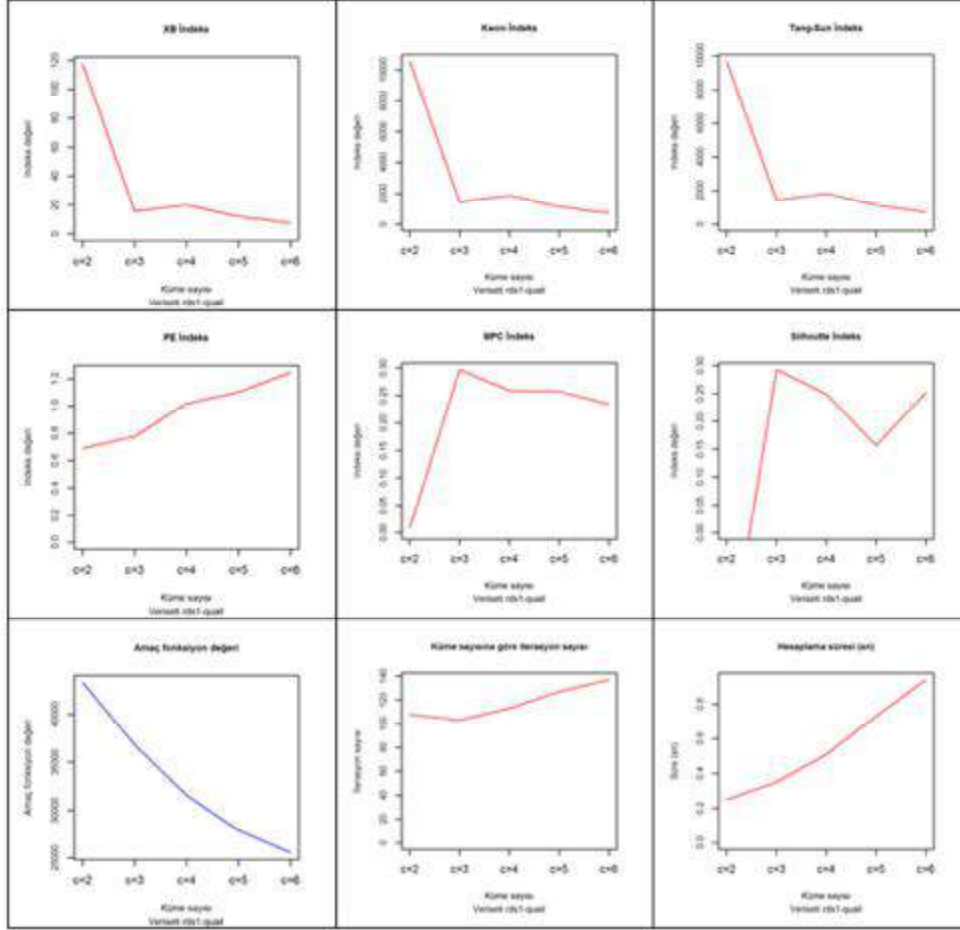
Çizelge 4.161. RDS2-Quail veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	213.081	19177.569	17578.561	2.483	1.095
c=3	25.133	2296.345	2261.694	2.247	0.719
c=4	40.881	3733.569	3641.493	2.124	0.662
c=5	23.741	2188.602	2167.383	1.848	0.565
c=6	20.287	1881.364	1867.890	1.786	0.542

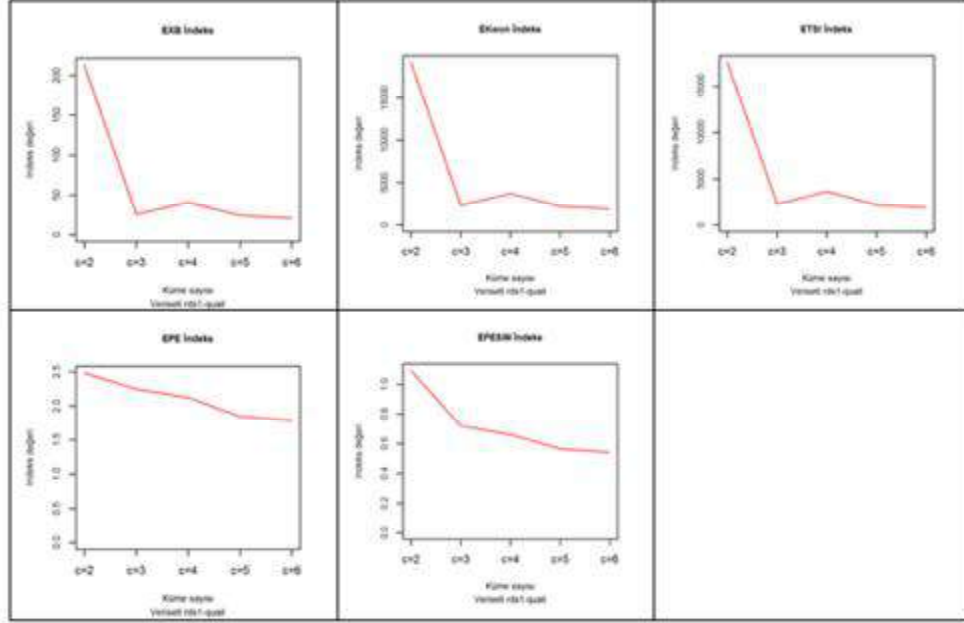
Çizelge 4.161’de görülen uzatılmış indeks değerlerinde tüm indeks değerleri uygun küme sayısını 6 (c=6) olarak uygun görmektedir. Gerçek küme sayısından (c=3) uzak bir tahminde bulunan indeks değerleri veri seti üzerinde etkin olmadıkları gözlemlenmektedir. Ancak indekslerin yani sıra uygulanan algoritmanın başarımları sonuçları gerçek küme sayısına yakın minimum iterasyon sayısı ve hesaplama sürelerini (Çizelge 4.162) önermektedir.

Çizelge 4.162. RDS2-Quail veri seti için İOCO algoritması başarımları

<i>C</i>	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	108	0.250	43358.642
c=3	103	0.350	36818.440
c=4	113	0.510	31568.288
c=5	127	0.730	27874.058
c=6	137	0.940	25548.287



Şekil 4.103. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail)



Şekil 4.104. İOCO algoritması uzatılmış indeks değerleri (RDS2-Quail)

4.5.2.3. Gerçek Veri Seti 3 (RDS3-Wine)

RDS3-Wine gerçek veri setinde geliştirilmiş bulanık indeks değerleri Çizelge 4.163'de verilmektedir. V_{XB} , V_{Kwon} , V_{Fsil} ve V_{TS} indeks değerlerine göre küme sayısının 5 ($c=5$) olduğu önerilmiştir. Bu durumda uygulanan kümeleme algoritması iyi bir başarımlı gösterememektedir.

Çizelge 4.163. RDS3-Wine veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	345.120	6.14e+03	61278.796	0.674	0.038	-0.424
c=3	127.725	2.27e+03	22718.831	1.000	0.070	-0.302
c=4	1658.069	3.00e+04	294295.463	1.000	0.244	0.010
c=5	40.459	7.38e+02	7430.867	1.000	0.241	0.117
c=6	1288.816	2.35e+04	232260.330	1.000	0.228	-0.186
c=7	476.209	9.65e+03	96969.314	1.000	0.278	0.025

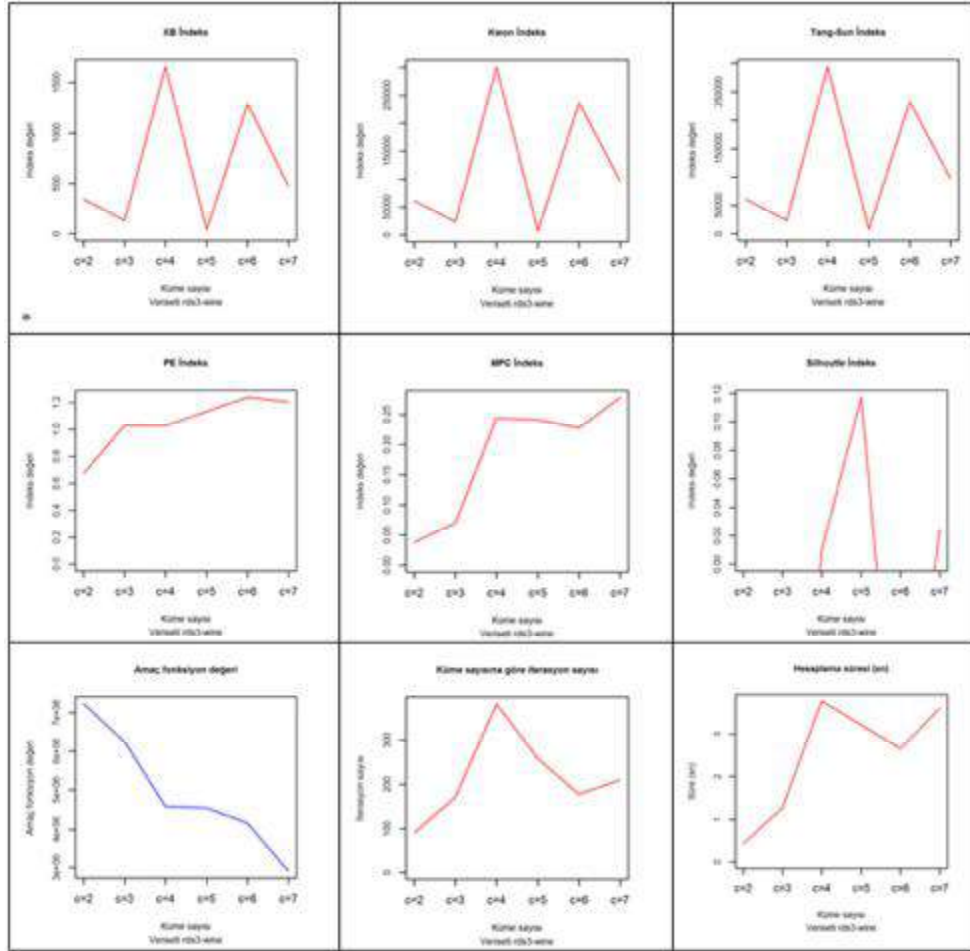
Çizelge 4.164. RDS3-Wine veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	584.039	1.03e+04	1.03e+04	3.974	1.515
c=3	249.668	4.44e+03	4.44e+03	3.884	1.262
c=4	8421.186	1.50e+05	1.46e+05	3.440	1.058
c=5	80.939	1.45e+03	1.46e+03	3.206	0.972
c=6	2521.431	4.55e+04	4.47e+04	2.907	0.878
c=7	1159.825	2.18e+04	2.16e+04	2.268	0.683

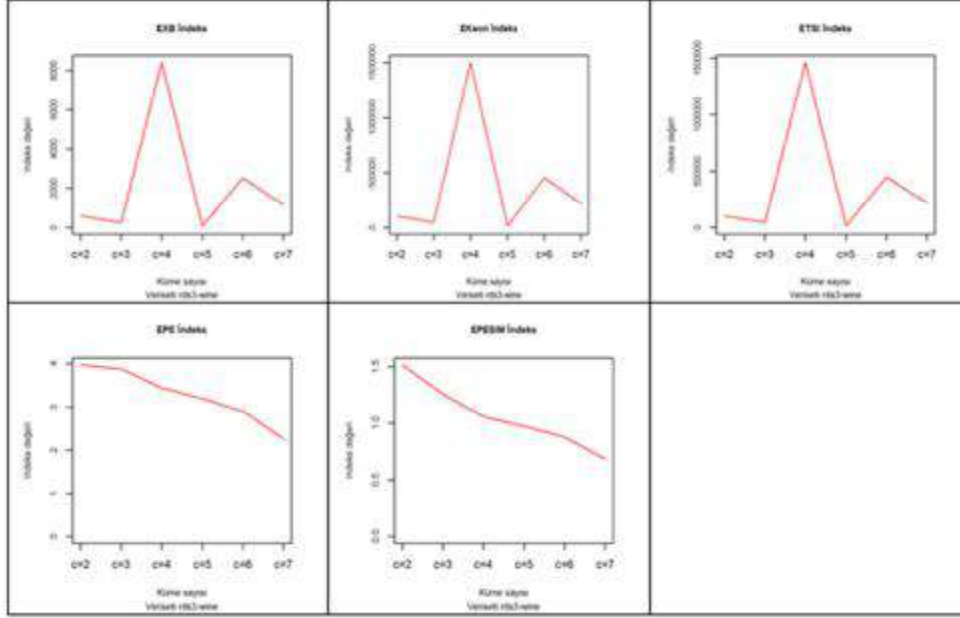
Çizelge 4.164’de görülen uzatılmış indeks değerlerinde tüm indeks değerleri uygun küme sayısını 5 (c=5) olarak uygun görmektedir. Burada V_{EPE} ve V_{EPESIM} indeks değerleri gerçek küme sayısına (c=3) daha da uzak bir tahminde (c=7) bulunarak veri seti üzerinde etkin olamadıkları göstermektedirler. Uygulanan algoritmanın başarımlarını değerleri incelendiğinde (Çizelge 4.165) gerçek küme sayısına yakın minimum iterasyon sayısı ve hesaplama sürelerini önermektedir.

Çizelge 4.165. RDS3-Wine veri seti için İOCO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	92	0.450	7238468.672
c=3	171	1.280	6237334.990
c=4	383	3.780	4579942.138
c=5	258	3.220	4548249.758
c=6	177	2.650	4163854.916
c=7	210	3.620	2906556.684



Şekil 4.105. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS3-Wine)



Şekil 4.106. İOCO algoritması uzatılmış indeks değerleri (RDS3-Wine)

4.5.2.4. Gerçek Veri Seti 4 (RDS4-Glass)

RDS4-Glass gerçek veri setinde genelleştirilmiş bulanık indeks değerleri Çizelge 4.166'da verilmektedir. V_{XB} indeks değeri gerçek küme sayısına ($c=6$) tam isabet tahmin yapardıktan veri seti üzerinde en etkin indeks olarak göze çarpmaktadır. V_{XB} indeksinin yanı sıra V_{PE} ve V_{MPC} indeksleri de yakın tahminlerde bulunarak küme sayısını 5 ($c=5$) olarak önerilmiştir. Ancak genel anlamda çoğu indekslerin gerçek küme sayısını işaret etmemesi ve V_{Fsil} 'den net bir hesaplama bile alınamaması durumu uygulanan kümeleme algoritmasının iyi bir başarıyı göstermediğini ifade etmektedir.

Çizelge 4.166. RDS4-Glass veri setinde İOCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	6.508	1392.906	803.923	0.626	0.128	0.373
c=3	9.267	1.984	1996.449	1.000	0.086	0.000
c=4	5.807	1.251	1610.273	1.000	0.111	0.000
c=5	2.972	6.585	6819.548	0.212	0.905	0.000
c=6	2.303	4.936	1884.869	1.000	0.054	0.000
c=7	3.358	7.200	1948.075	1.000	0.038	0.000

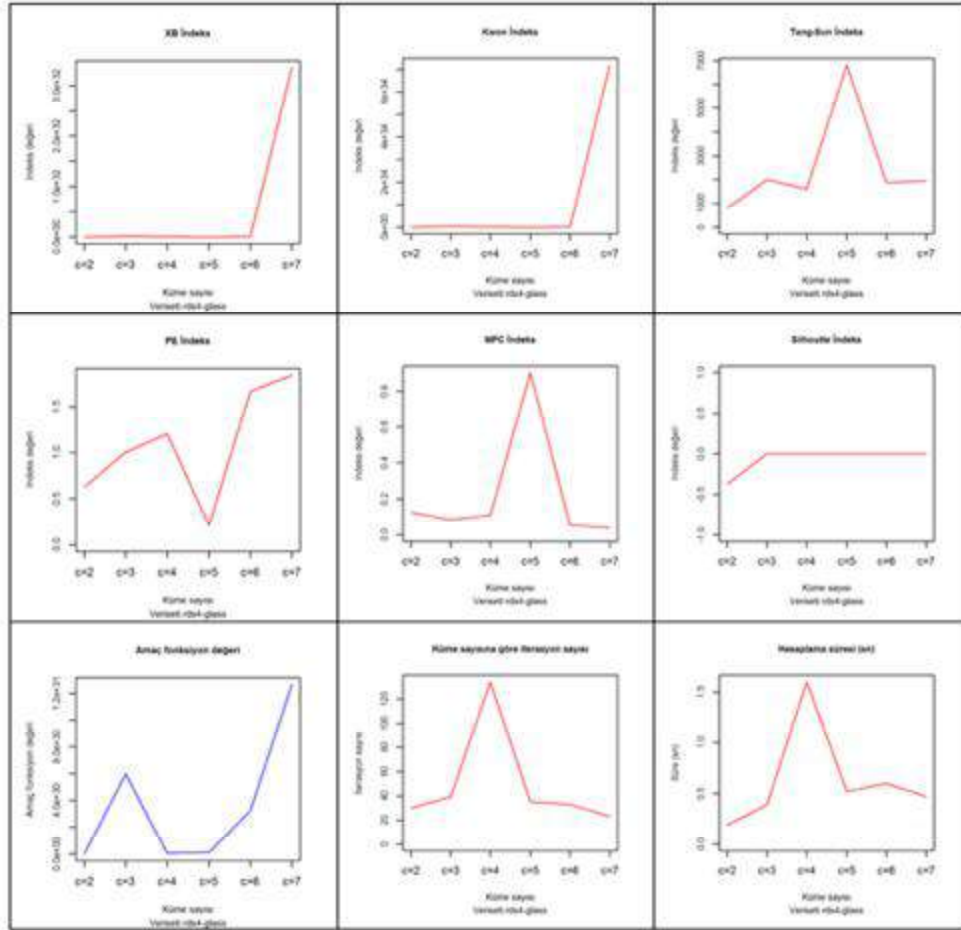
Çizelge 4.167. RDS4-Glass veri setinde İOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	8.376	1792.621	1034.580	3.979	1.499
c=3	1.970	4.217	4242.581	4.324	1.385
c=4	1.255	2.694	3461.847	4.921	1.511
c=5	7.166	1.535	157762.5	1.444	0.440
c=6	9.130	1.954	7461.182	6.073	1.838
c=7	1.469	3.145	8508.797	6.083	1.838

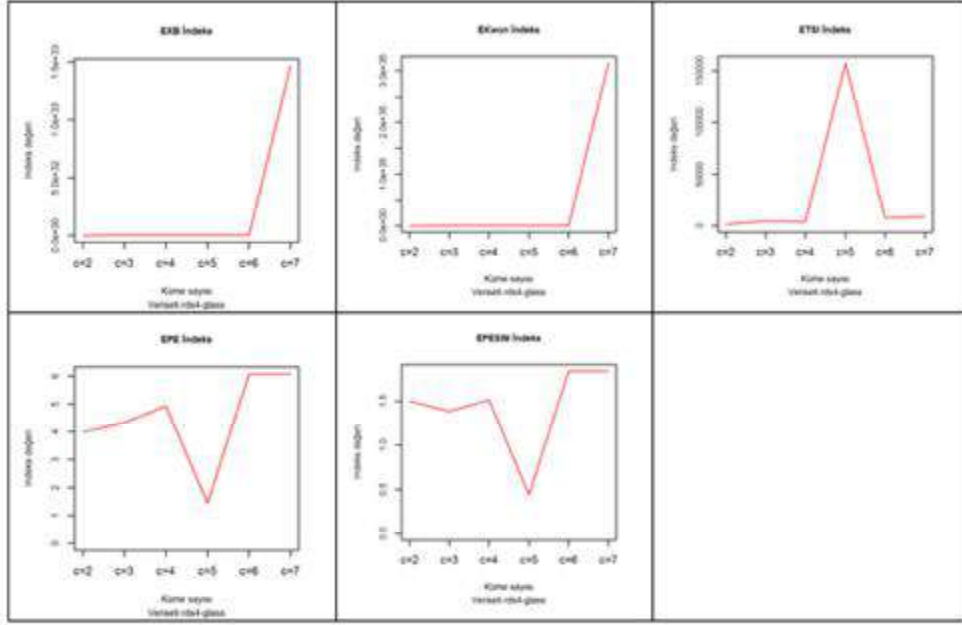
Çizelge 4.167’de görülen uzatılmış indeks değerlerinin hiçbirinde gerçek küme (c=6) sayısının isabetine rastlanmamaktadır. V_{EPE} , V_{EKwon} ve V_{EPESIM} indeks değerleri küme sayısını 5 olarak önermiştir. Bu öneri diğer indekslere göre en yakın öneri olarak görülmektedir. İterasyon sayısı ve hesaplama süreleri bakımından minimum öneriler için küme sayısı Çizelge 4.168’de 2 (c=2) olarak gösterilmektedir.

Çizelge 4.168. RDS4-Glass veri seti için İOCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	30	0.190	1120.077
c=3	39	0.390	5.965
c=4	134	1.600	4.939
c=5	35	0.520	9.739
c=6	33	0.600	3.231
c=7	23	0.470	1.266



Şekil 4.107. İOCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS4-Glass)



Şekil 4.108. İOCO algoritması uzatılmış indeks değerleri (RDS4-Glass)

4.6. UOBCO Algoritması

4.6.1. Sentetik Veri Setleri

4.6.1.1. Sentetik Veri Seti 1 (SDSC)

Çizelge 4.169 incelendiğinde daire şekilli gürültüsüz veri setine uygulanan yönetimsiz olabilirlikli bulanık ortalamalar (UOBCO) kümeleme algoritmasıyla elde edilen değerleri genelleştirilmiş bulanık indeksler ile tahminleri gösterilmektedir. Birçok kümeleme algoritmasında sıkça karşılaşılan ve uygulamalarda sorun oluşturan gürültülü noktalara (noisy points) karşı iyi bir basarım gösteren bu kümeleme algoritması veri seti (SDSC) üzerinde de tüm indeksler tarafından gerçek küme sayısına ($c=4$) isabet etmesiyle bu durumu sonuçlar üzerinden de desteklemektedir. Ek 6.1 ve Ek 6.1 (Devamı) küme yapılarına da bakıldığında kümeler içerisindeki veri noktaları arasındaki mesafelerin birbirine yakın, aynı zamanda örtüşen küme yapıları oluşturmaması

kümeleme algoritmasının veri seti üzerinde iyi bir basarım gösterdiği anlamına gelmektedir.

Çizelge 4.169. SDSC veri setinde UOBACO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.267	213.891	187.897	0.237	0.731	0.579
c=3	0.132	106.124	97.749	0.242	0.804	0.616
c=4	0.075	60.532	56.475	0.128	0.929	0.799
c=5	0.679	549.148	335.842	0.224	0.864	0.703
c=6	0.720	581.884	361.073	0.270	0.851	0.694
c=7	0.994	803.822	455.864	0.314	0.836	0.677
c=8	3.290	2658.203	808.709	0.372	0.815	0.641

Çizelge 4.170'de küme sayısının değişimine göre elde edilen uzatılmış indeks değerleri gözlemlenmektedir. Tahminler genel anlamda 4 (c=4) şeklinde olmaktadır ki buda gerçek küme sayısına isabet etmektedir. EPE ve EPESIM indeksleri ise tahminlerini 8 (c=8) olarak işaret etmektedir ki bu durum bu iki indeks (EPE ve EPESIM) veri seti üzerinde etkin indeksler olmadığını göstermektedir.

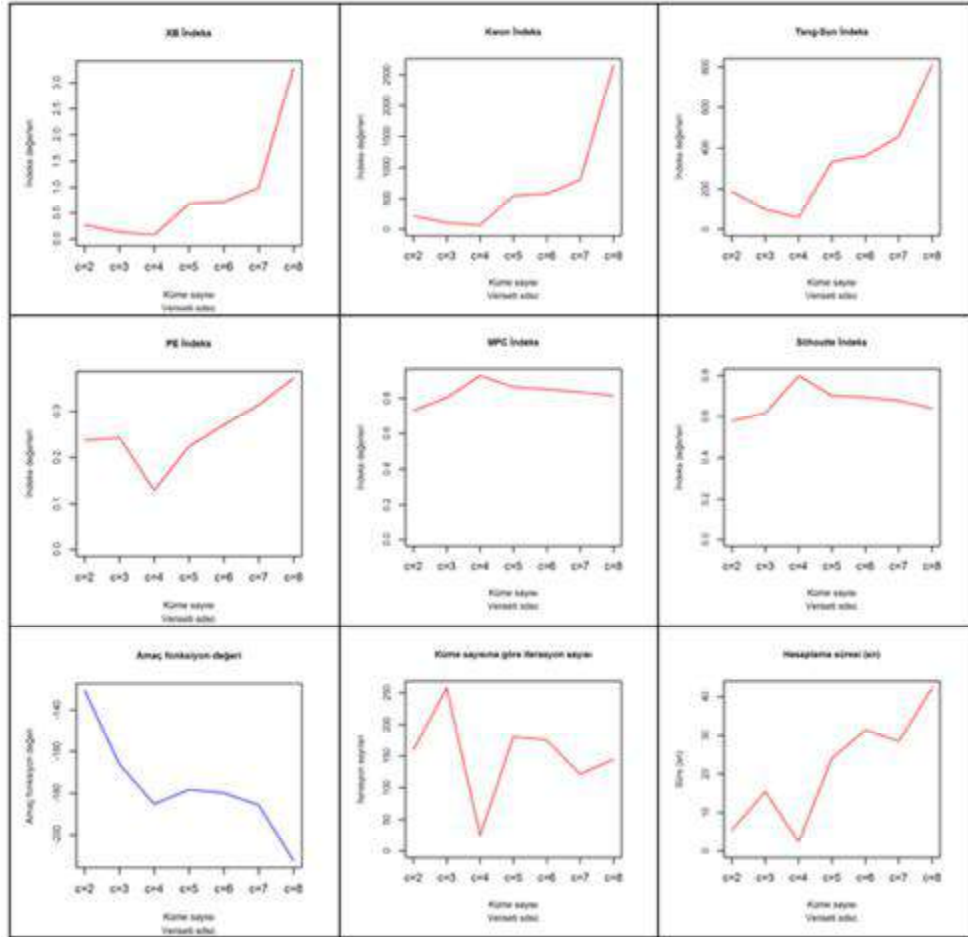
Çizelge 4.170. SDSC veri setinde UOBACO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.283	226.834	199.253	8.931	2.719
c=3	0.149	119.351	109.912	5.761	1.736
c=4	0.097	78.435	73.128	4.860	1.459
c=5	0.926	746.355	456.142	4.785	1.438
c=6	1.009	813.118	504.262	4.722	1.419
c=7	1.423	1146.690	649.981	4.614	1.386
c=8	4.813	3876.435	1178.814	4.548	1.366

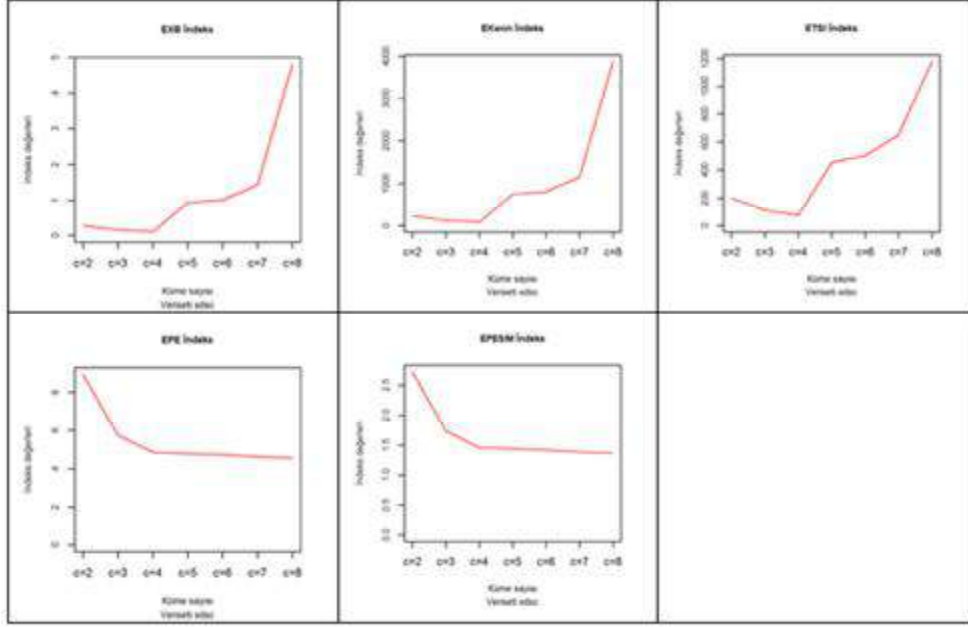
Çizelge 4.171'e bakıldığında uygulanan kümeleme algoritmasının basarım değerleri görülmektedir. Daha önceden geliştirilen OBCO algoritmasında BCO'nun çalıştırması sonucu hesaplanan eta vektörünün (η_i) hesaplanmasına ihtiyaç duyulmaması ve bu nedenle hesaplama süresinin kısılması avantajı da veri setine uygulanan UOBCO kümeleme algoritmasında sonuçlar üzerindeki etkisini göstermektedir. Diğer küme sayısı tahminlerinden çok daha kısa surede hesaplama yapan algoritmada minimum hesaplamayı 4 ($c=4$) olarak göstermektedir. Minimum iterasyon sayısında 4 ($c=4$) olarak gözlenen bu veri setinde, küme sayıları gerçek küme sayısından ($c=4$) uzaklaştıkça iterasyon sayılarında büyük oranda artışlara neden olmaktadır.

Çizelge 4.171. SDSC veri seti için UOBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	160	5.470	-130.661
c=3	258	15.430	-166.196
c=4	25	2.360	-185.220
c=5	181	24.090	-178.199
c=6	175	31.420	-179.833
c=7	121	28.380	-185.654
c=8	145	42.370	-212.540



Şekil 4.109. UOBDO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSC)



Şekil 4.110. UOBCO algoritması uzatılmış indeks değerleri (SDSC)

4.6.1.2. Sentetik Veri Seti 2 (SDSCN)

Daire şekilli gürültüsüz sentetik veri setinden elde edilen genelleştirilmiş indeks tahminleri Çizelge 4.172’de gösterilmektedir. İndeksler tahminlerinde aynı küme sayısını ($c=4$) yani gerçek küme sayısını ($c=4$) işaret etmektedirler. Bu durumda uygulanan algoritmanın veri seti üzerindeki iyi başarımı açıkça görülmektedir.

Çizelge 4.172. SDSCN veri setinde UOBOCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	<i>V_{XB}</i>	<i>V_{Kwon}</i>	<i>V_{TS}</i>	<i>V_{PE}</i>	<i>V_{MPC}</i>	<i>V_{Fsil}</i>
c=2	0.273	223.934	196.240	0.236	0.733	0.577
c=3	0.135	111.048	102.420	0.263	0.787	0.595
c=4	0.076	62.964	58.906	0.155	0.910	0.778
c=5	0.618	511.645	331.035	0.241	0.856	0.690
c=6	0.762	631.083	391.485	0.297	0.835	0.662
c=7	1.342	1110.801	562.434	0.363	0.807	0.630
c=8	2.923	2418.578	822.709	0.426	0.783	0.604

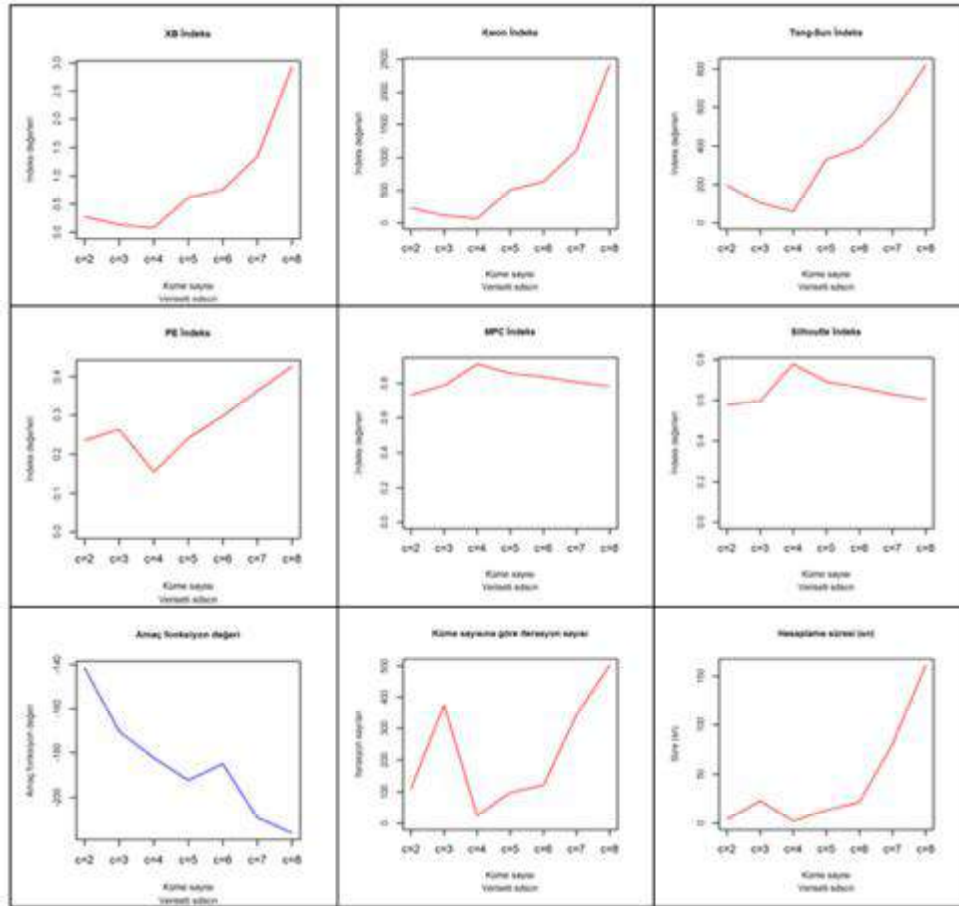
Çizelge 4.173. SDSCN veri setinde UOBOCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	0.290	237.968	208.525	9.123	2.780
c=3	0.151	124.071	114.411	6.052	1.824
c=4	0.097	79.841	74.653	5.015	1.506
c=5	0.818	675.341	436.699	4.881	1.467
c=6	1.035	855.067	530.166	4.778	1.436
c=7	1.863	1538.255	778.528	4.663	1.401
c=8	4.139	3416.112	1161.592	4.568	1.373

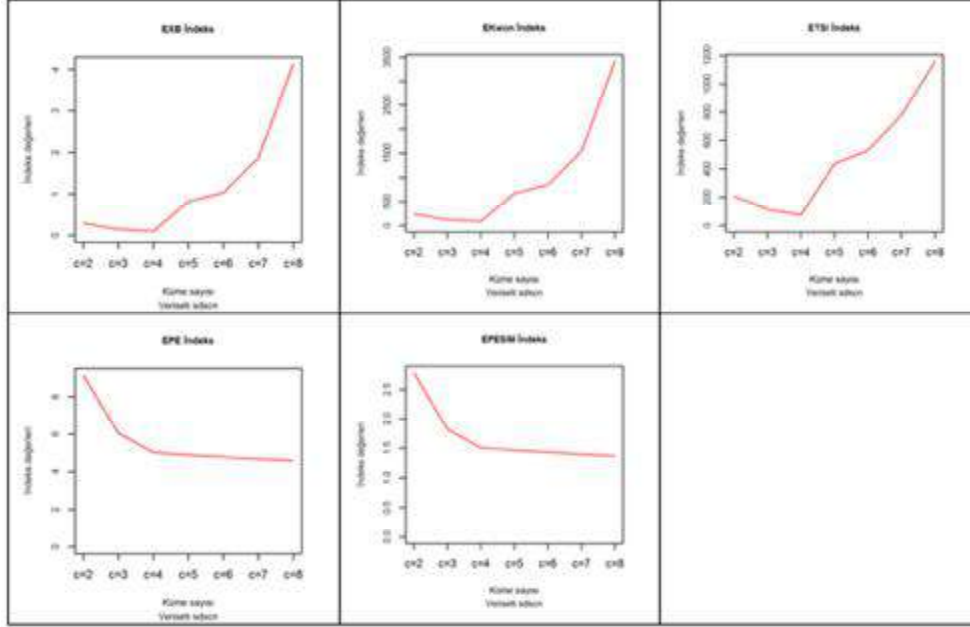
Çizelge 4.173'te uzatılmış indeks değerlerinin tahminleri verilmektedir. Tahminler SDSC veri setinde gözlenen sonuçlar benzer bir tablo (Çizelge 4.170) oluşturmaktadır. Veri setine uygulanan algoritmanın genel basarımı ise 4.174'te gösterilmektedir ki minimum hesaplamalar gerçek küme sayısını (c=4) işaret etmektedir.

Çizelge 4.174. SDSCN veri seti için UOBKO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	109	3.780	-141.526
c=3	374	22.880	-170.337
c=4	24	2.300	-182.205
c=5	98	13.460	-192.253
c=6	120	21.770	-184.986
c=7	343	80.550	-209.305
c=8	502	160.980	-215.938



Şekil 4.111. UOBKO algoritması küme geçerlilik indeksleri ve kümeleme başarımı(SDSCN)



Şekil 4.112. UOBCO algoritması uzatılmış indeks değerleri (SDSCN)

4.6.1.3. Sentetik Veri Seti 3 (SDSE)

Elips şekilli gürültüsüz sentetik veri setinden elde edilen genelleştirilmiş bulanık indeks değerlerine göre genel tahmin Çizelge 4.175’de gösterilmektedir. Burada tüm indeks tahminlerinin gerçek küme sayısını ($c=4$) göstermesi uygulanan kümeleme algoritmasının iyi bir basarım gösterdiğini ifade etmektedir.

Çizelge 4.175. SDSE veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	<i>V_{XB}</i>	<i>V_{Kwon}</i>	<i>V_{TS}</i>	<i>V_{PE}</i>	<i>V_{MPC}</i>	<i>V_{Fsil}</i>
c=2	0.249	199.153	175.377	0.266	0.667	0.521
c=3	0.156	125.363	114.758	0.272	0.787	0.598
c=4	0.100	80.919	75.235	0.241	0.842	0.686
c=5	0.316	254.561	206.092	0.305	0.813	0.632
c=6	0.245	198.171	166.929	0.319	0.815	0.622
c=7	0.450	363.456	274.921	0.379	0.788	0.591
c=8	0.795	643.178	419.037	0.427	0.770	0.559

Elips şekilli gürültüsüz sentetik veri setinden elde edilen uzatılmış indeks değerlerine göre genel tahmin EPESIM ve EPE dışında Çizelge 4.176'de gösterildiği gibi 4 (c=4)'tur.

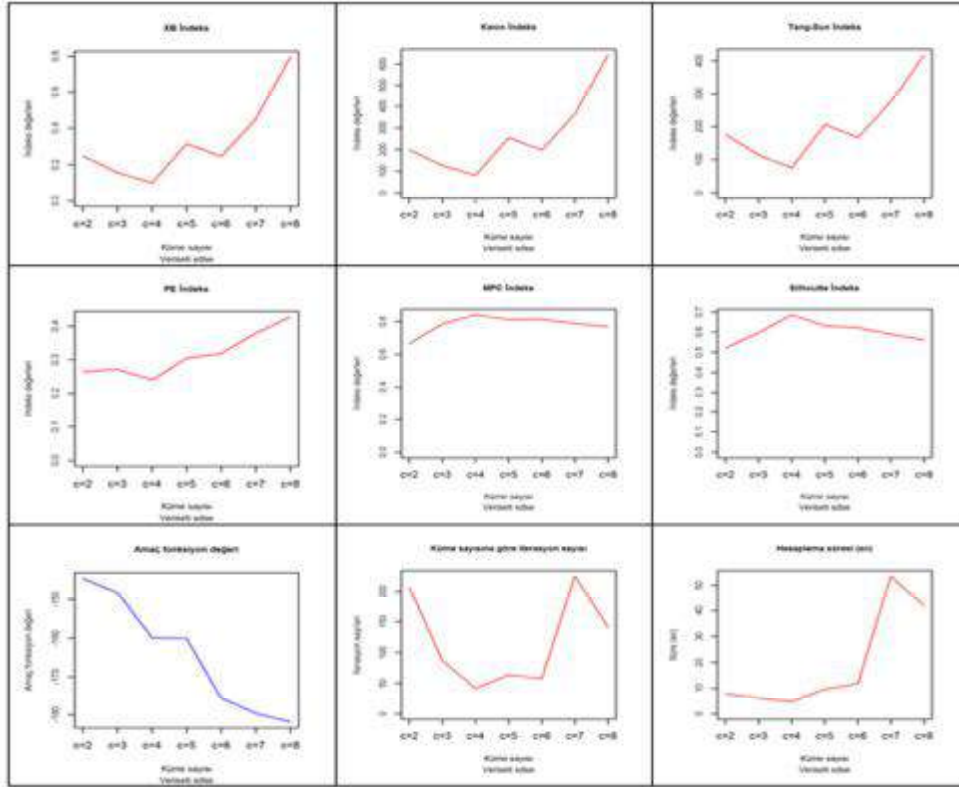
Çizelge 4.176. SDSE veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	0.263	210.989	185.787	7.935	2.439
c=3	0.173	138.966	127.192	6.466	1.949
c=4	0.121	97.003	90.157	5.350	1.608
c=5	0.393	316.344	256.015	4.994	1.501
c=6	0.318	256.201	215.714	4.539	1.363
c=7	0.596	480.794	363.524	4.348	1.306
c=8	1.072	864.855	563.239	4.168	1.252

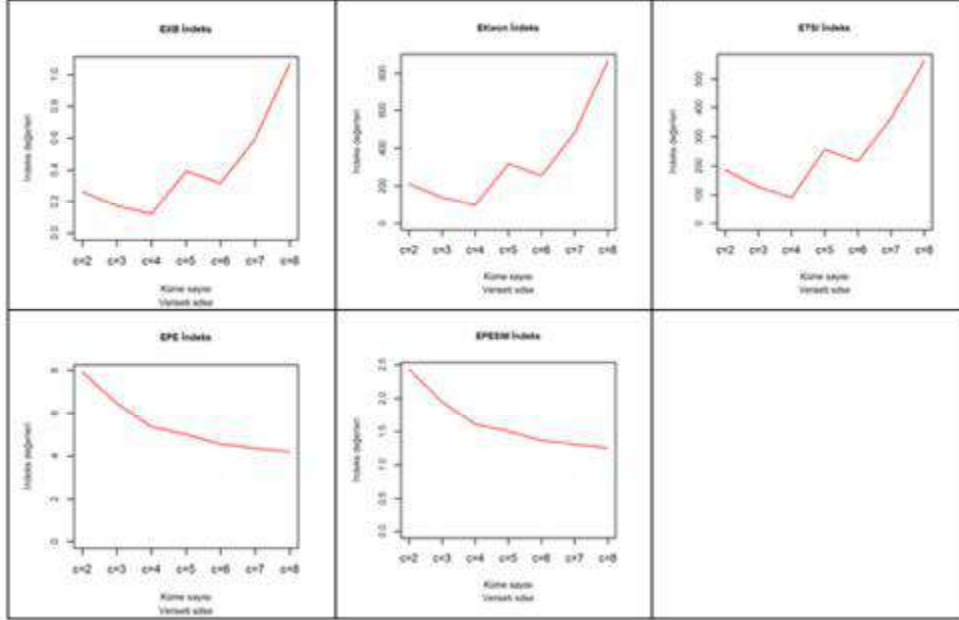
UOBCO algoritmasının veri seti üzerindeki basarimi Şekil 4.113'te iterasyon sayısı ile hesaplama süresi arasında pozitif ve benzer eğilimler gözlenmektedir.

Çizelge 4.177. SDSE veri seti için UOBDO algoritması başarımları

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	206	8.000	-144.348
$c=3$	87	6.110	-148.139
$c=4$	42	4.630	-159.957
$c=5$	64	9.670	-160.128
$c=6$	58	11.810	-175.634
$c=7$	225	53.640	-179.655
$c=8$	141	41.800	-181.866



Şekil 4.113. UOBDO algoritması küme geçerlilik indeksleri ve kümeleme başarımları (SDSE)



Şekil 4.114. UOBEO algoritması uzatılmış indeks değerleri (SDSE)

4.6.1.4. Sentetik Veri Seti 4 (SDSEN)

Elips şekilli gürültülü sentetik veri setinden elde edilen geliştirilmiş bulanık indeks değerlerine göre genel tahmin Çizelge 4.178’de gösterilmektedir. Burada tüm indeks tahminlerini gürültüsüz veri setinde (SDSE) olduğu gibi gerçek küme sayısını ($c=4$) göstermesi uygulanan kümeleme algoritmasının iyi bir basarım gösterdiğini ifade etmektedir.

Çizelge 4.178. SDSEN veri setinde UOBACO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	<i>V_{XB}</i>	<i>V_{Kwon}</i>	<i>V_{TS}</i>	<i>V_{PE}</i>	<i>V_{MPC}</i>	<i>V_{Fsil}</i>
c=2	0.259	212.719	186.635	0.254	0.689	0.534
c=3	0.140	115.281	106.142	0.255	0.800	0.618
c=4	0.085	70.288	65.553	0.194	0.878	0.742
c=5	0.288	238.604	191.131	0.248	0.854	0.691
c=6	0.696	575.927	370.470	0.322	0.820	0.668
c=7	0.678	562.524	359.486	0.333	0.817	0.648
c=8	1.151	955.729	506.396	0.383	0.797	0.627

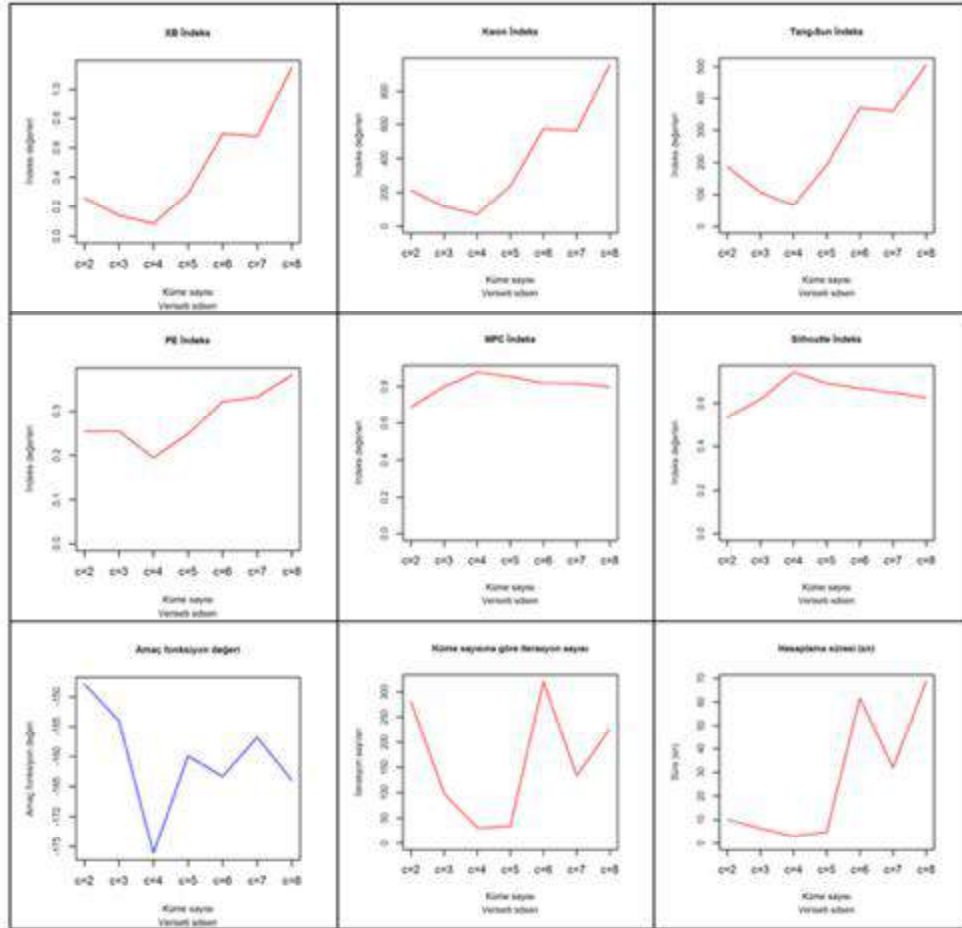
Elips şekilli gürültülü sentetik veri setinden elde edilen uzatılmış indeks değerlerine göre EPESIM ve EPE bu veri seti (SDSEN) üzerinde de farklı tahminde bulunarak etkin bir indeks olmadıklarını göstermektedir (Çizelge 4.179). Uzatılmış indekslerin genel tahmini (c=4) yine gerçek küme sayısına (c=4) isabet etmektedir.

Çizelge 4.179. SDSEN veri setinde UOBACO algoritması ile küme sayısına göre uzatılmış indeks değerleri

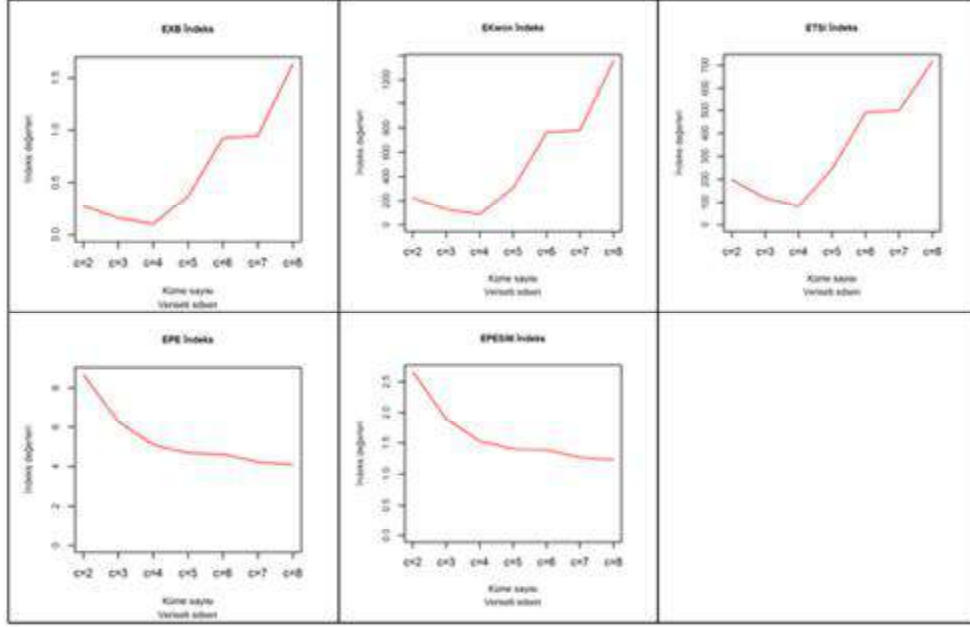
<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	0.276	226.418	198.640	8.685	2.664
c=3	0.156	128.479	118.274	6.278	1.892
c=4	0.106	87.186	81.274	5.103	1.533
c=5	0.372	307.535	246.217	4.686	1.408
c=6	0.922	761.553	489.646	4.617	1.387
c=7	0.942	779.615	497.940	4.231	1.271
c=8	1.637	1354.078	717.090	4.090	1.228

Çizelge 4.180. SENSEN veri seti için UOBKO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	281	10.120	-147.885
$c=3$	99	6.300	-153.953
$c=4$	30	2.900	-176.057
$c=5$	34	4.860	-159.923
$c=6$	321	61.660	-163.372
$c=7$	133	32.000	-156.835
$c=8$	226	68.760	-163.985



Şekil 4.115. UOBKO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SENSEN)



Şekil 4.116. UOBCO algoritması uzatılmış indeks değerleri (SDSEN)

4.6.1.5. Sentetik Veri Seti 5 (SDSR)

Sentetik veri setinden (SDSR) hesaplanan geliştirilmiş bulanık indeks değerleri Çizelge 4.181’de gösterilmektedir. Elde edilen değerler incelendiğinde tüm indeks tahminleri 4 ($c=4$)’u göstermektedir. Genel anlamda yapılan tahminlerin gerçek küme sayısına ($c=4$) tam isabet etmesinden dolayı algoritmanın iyi bir basarım gösterdiği anlaşılmaktadır.

Çizelge 4.181. SDSR veri setinde UOBKO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	<i>V_{XB}</i>	<i>V_{Kwon}</i>	<i>V_{TS}</i>	<i>V_{PE}</i>	<i>V_{MPC}</i>	<i>V_{Fsil}</i>
c=2	0.261	208.946	183.729	0.220	0.755	0.594
c=3	0.137	110.064	101.404	0.268	0.782	0.594
c=4	0.082	66.511	61.917	0.153	0.912	0.780
c=5	0.574	463.334	307.551	0.239	0.857	0.687
c=6	0.494	399.667	278.364	0.270	0.844	0.660
c=7	0.813	657.868	395.618	0.319	0.826	0.658
c=8	6.357	5133.29	984.257	0.437	0.778	0.610

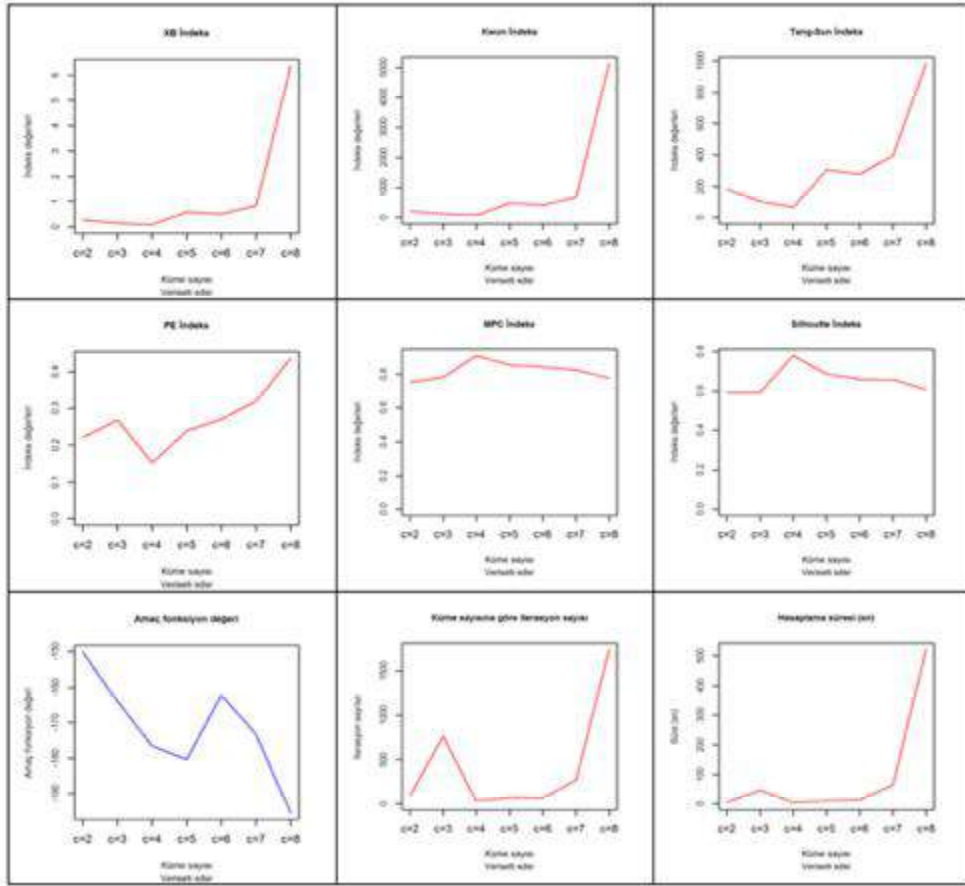
Çizelge 4.182. SDSR veri setinde UOBKO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	0.279	223.080	196.143	8.699	2.645
c=3	0.153	122.744	113.068	5.961	1.797
c=4	0.104	84.061	78.211	4.918	1.477
c=5	0.759	611.627	405.749	4.767	1.432
c=6	0.681	549.427	382.431	4.517	1.357
c=7	1.157	933.263	560.909	4.441	1.334
c=8	9.049	7286.498	1396.568	4.483	1.347

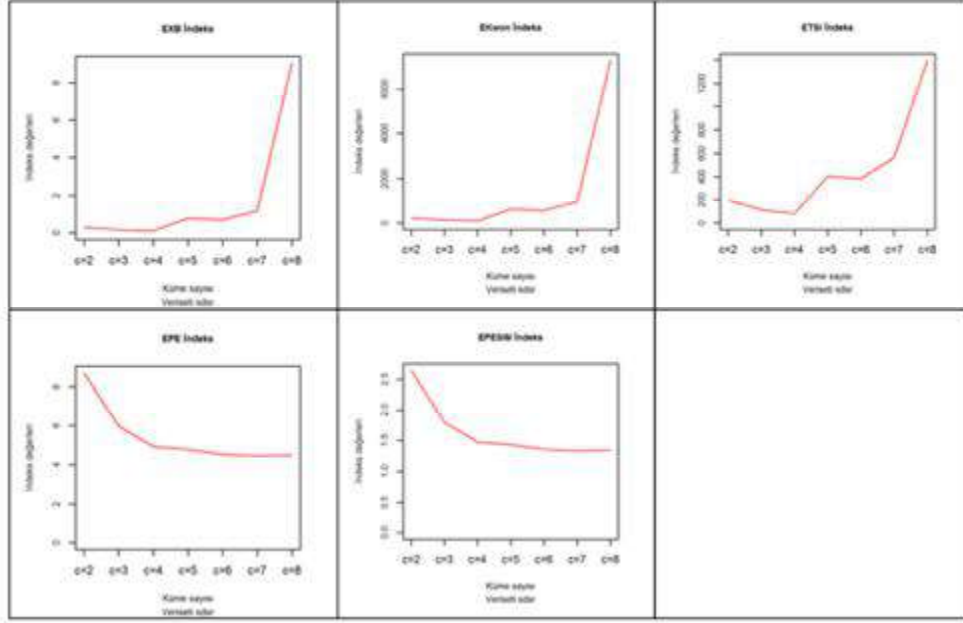
Çizelge 4.182’de uzatılmış indeks değerlerinin tahminleri ve Çizelge 4.183’de veri setine uygulanan algoritmanın basarım sonuçları gösteriyor ki tahminler (c=4) genel anlamda gerçek küme sayısına (c=4) isabet etmektedir.

Çizelge 4.183. SDSR veri seti için UOBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	92	3.440	-150.125
c=3	764	47.360	-164.047
c=4	26	2.540	-176.650
c=5	70	9.480	-180.325
c=6	66	12.300	-162.311
c=7	270	64.830	-173.419
c=8	1741	521.560	-195.530



Şekil 4.117. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSR)



Şekil 4.118. UOBCO algoritması uzatılmış indeks değerleri (SDSR)

4.6.1.6. Sentetik Veri Seti 6 (SDSRN)

Çizelge 4.184'deki değerler ve Şekil 4.119'daki eğilimler gösteriyor ki indekslerin optimal değerleri gerçek küme sayısını net bir şekilde tahmin etmektedirler. Bu durum uygulanan algoritmanın sağlam bir basarım gösterdiği anlamına gelmektedir.

Çizelge 4.184. SDSRN veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.268	219.903	192.852	0.223	0.750	0.589
c=3	0.155	127.154	116.149	0.264	0.787	0.592
c=4	0.081	66.871	62.333	0.162	0.904	0.773
c=5	0.449	371.834	265.482	0.238	0.857	0.690
c=6	1.036	858.003	467.477	0.303	0.829	0.649
c=7	0.861	714.532	421.237	0.334	0.817	0.636
c=8	1.305	1082.403	549.126	0.376	0.803	0.629

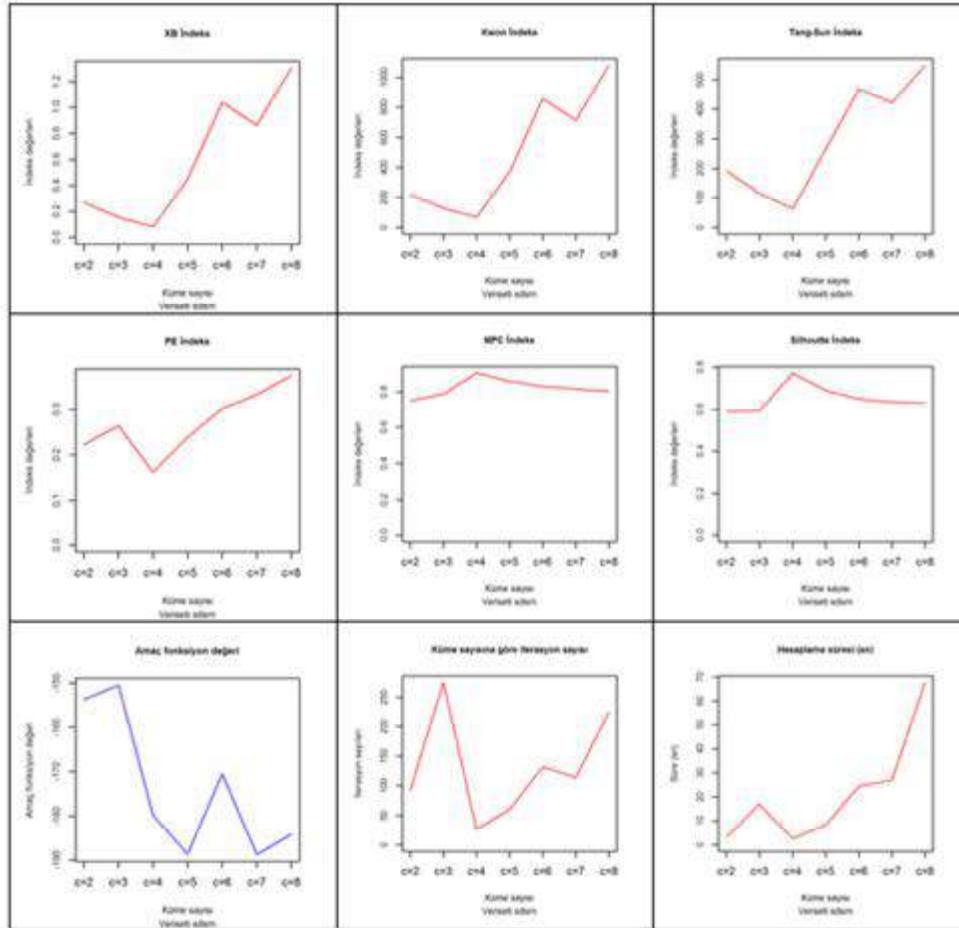
Uygulanan kümeleme algoritmasıyla elde edilen Çizelge ve Şekillerde şüana kadar yapılan işlemlerin hepsinde olduğu gibi bu veri setinde de uzatılmış indeks değerleri EPESIM ve EPE dışında aynı küme sayısı ($c=4$) tahmininde bulunmaktadır (Çizelge 4.185). Aslında bu durum indeks değerlerinin tahminlerinin uygulandığı veri seti şekillerinden etkilenmediği anlaşılmaktadır.

Çizelge 4.185. SDSRN veri setinde UOBACO algoritması ile küme sayısına göre uzatılmış indeks değerleri

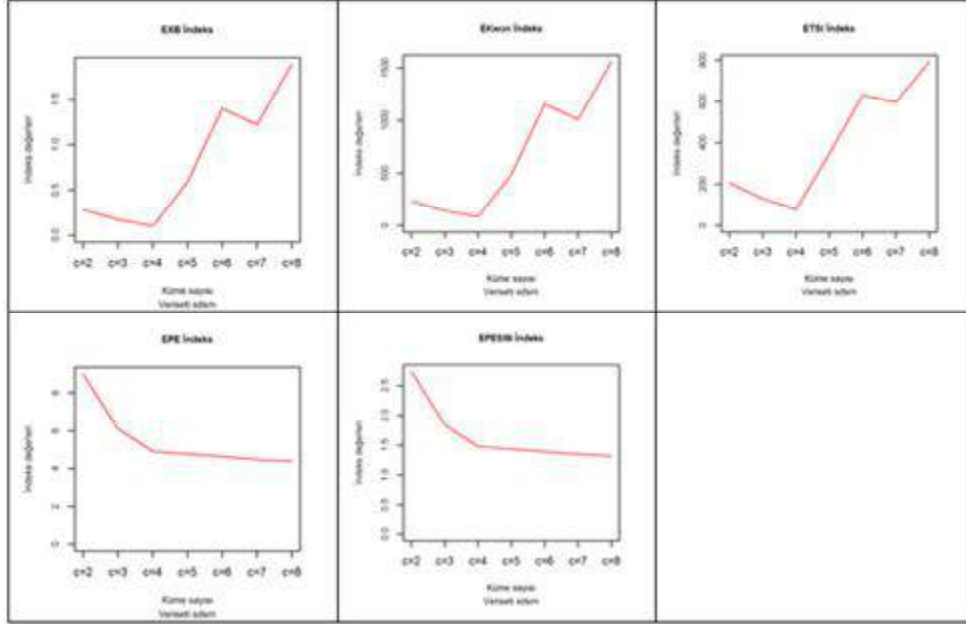
C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
$c=2$	0.286	234.835	205.933	8.991	2.738
$c=3$	0.173	142.000	129.689	6.108	1.841
$c=4$	0.102	84.08	78.331	4.913	1.476
$c=5$	0.591	488.179	348.354	4.764	1.431
$c=6$	1.403	1158.501	630.879	4.626	1.390
$c=7$	1.221	1009.725	594.925	4.479	1.345
$c=8$	1.890	1562.057	792.070	4.395	1.320

Çizelge 4.186. SDSRN veri seti için UOBACO algoritması başarıımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	93	3.240	-153.673
$c=3$	275	16.920	-150.413
$c=4$	26	2.590	-179.785
$c=5$	61	8.250	-188.723
$c=6$	132	24.360	-170.444
$c=7$	114	27.140	-188.744
$c=8$	225	67.690	-184.208



Şekil 4.119. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSRN)



Şekil 4.120. UOBCO algoritması uzatılmış indeks değerleri (SDSRN)

4.6.1.7. Sentetik Veri Seti 7 (SDSX)

Konkav şekilli gürültüsüz veri setinden elde edilen bulanık genelleştirilmiş indeks değerlerinin tahminleri Çizelge 4.187’de gösterilmektedir. İndeks değerlerinin tahmini optimal küme sayısının V_{PE} indeksi dışında 5 olarak önerildiği görülmektedir. Algoritmanın veri seti üzerinde oluşturduğu küme yapıları Ek 6.7 ve Ek 6.7 (Devamı)’da gösteriyor ki oluşan küme yapılarındaki veri noktalarında kopmalar yani noktalar arası mesafenin artması bu da bir gürültü durumu ortaya koyduğu gözlenmektedir. Şekil 4.121’de indekslerin oluşturduğu eğilimlerde tahminlerin optimal değerleri gerçek küme sayısını göstermediği bu durumda uygulanan algoritmanın bu veri seti üzerinde iyi bir başarı gösteremediği anlaşılmaktadır.

Çizelge 4.187. SDSX veri setinde UOBKO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	<i>V_{XB}</i>	<i>V_{Kwon}</i>	<i>V_{TS}</i>	<i>V_{PE}</i>	<i>V_{MPC}</i>	<i>V_{Fsil}</i>
c=2	0.338	271.478	231.873	0.294	0.643	0.491
c=3	0.185	148.578	135.627	0.385	0.679	0.488
c=4	0.156	125.967	115.599	0.401	0.723	0.547
c=5	0.114	92.208	85.640	0.316	0.807	0.655
c=6	0.149	121.142	109.845	0.349	0.797	0.645
c=7	0.274	222.642	188.535	0.389	0.785	0.614
c=8	0.292	237.185	196.574	0.392	0.789	0.593

Çizelge 4.188’de elde edilen uzatılmış indeks tahminleri gösterilmektedir. Tahminler genel anlamda bulanık indeks tahminlerine benzer küme sayıları tahmininde bulunarak 5 (c=5)’i göstermektedirler. Aynı zamanda uygulanması istenilen yeni EPE ve EPESIM indeks tahminleri de gerçek küme sayısından (c=3) uzak bir tahmin yaparak 8 (c=8)’i işaret etmektedir.

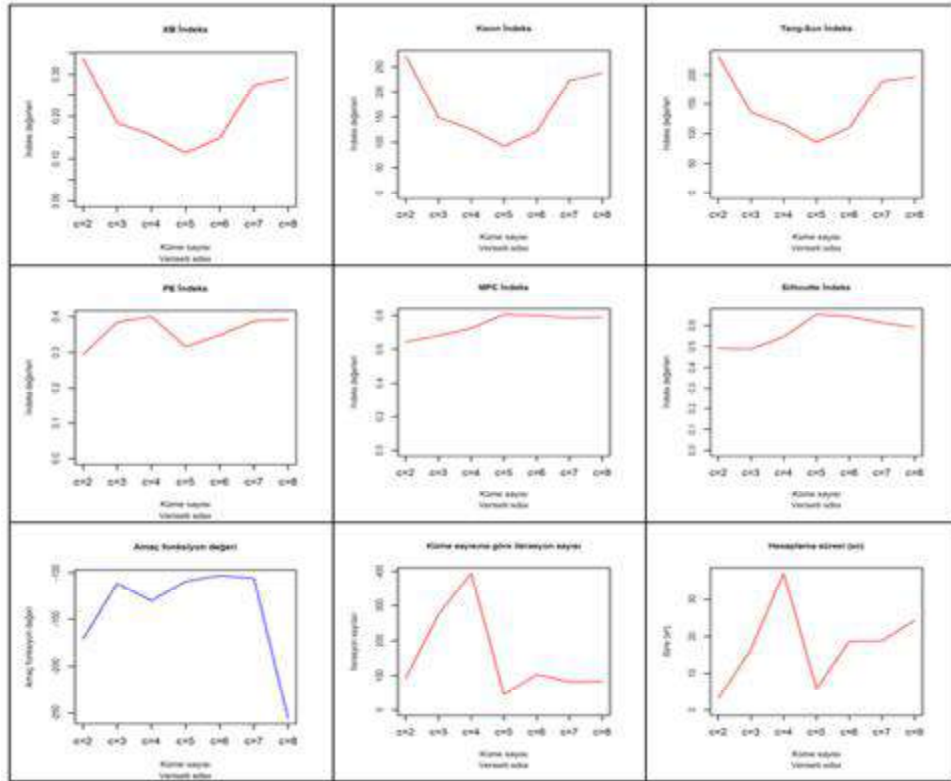
Çizelge 4.188. SDSX veri setinde UOBKO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	<i>V_{EXB}</i>	<i>V_{EKwon}</i>	<i>V_{ETS}</i>	<i>V_{EPE}</i>	<i>V_{EPESIM}</i>
c=2	0.363	291.728	249.153	10.392	3.215
c=3	0.201	161.948	147.817	7.388	2.234
c=4	0.176	141.673	129.988	5.724	1.723
c=5	0.137	110.342	102.445	4.614	1.386
c=6	0.185	149.716	135.696	4.267	1.281
c=7	0.345	279.238	236.362	3.939	1.183
c=8	0.391	316.678	262.323	3.931	1.180

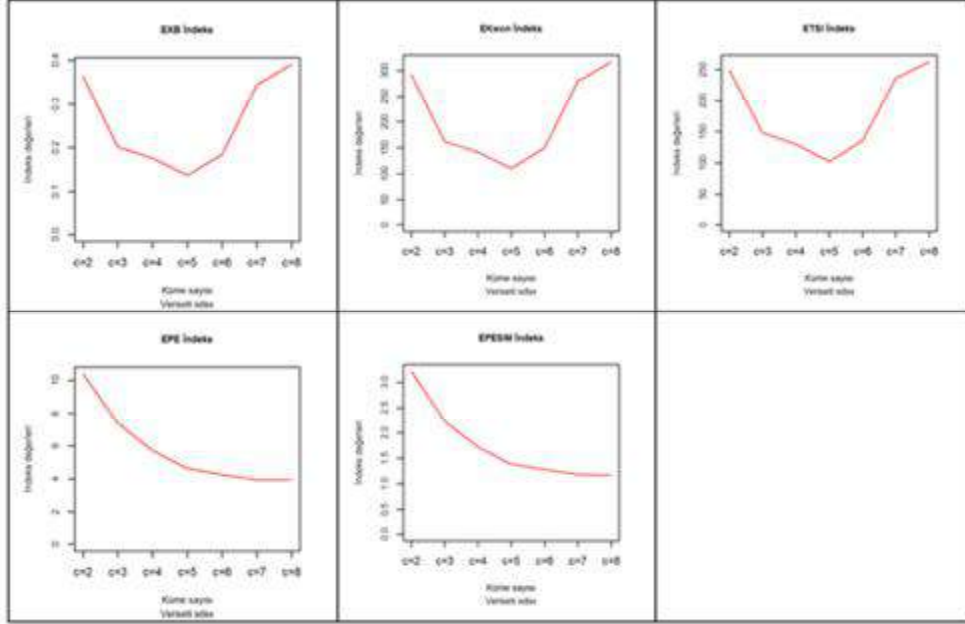
Çizelge 4.189’da uygulanan kümeleme algoritmasının basarım değerlerine göre minimum iterasyon işlemi 5 ($c=5$) olarak işaret edilmektedir. Farklı bir durum olarak hesaplama süresi en kısa süre olarak 2 ($c=2$) yapıldığı gözlenmektedir.

Çizelge 4.189. SDSX veri seti için UOBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	93	3.150	-171.140
c=3	275	16.590	-112.061
c=4	394	37.210	-130.740
c=5	45	5.890	-109.443
c=6	103	18.690	-103.314
c=7	82	18.790	-106.612
c=8	83	24.280	-255.218



Şekil 4.121. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (SDSX)



Şekil 4.122. UOBCO algoritması uzatılmış indeks değerleri (SDSX)

4.6.1.8. Sentetik Veri Seti 8 (SDSXXN)

Çizelge 4.190'da elde edilen indeks tahminlerine bakıldığında uygulamaların veri setindeki gürültüden etkilendikleri anlaşılmaktadır. Sonuçların göstermiş olduğu küme sayısı genel anlamda 6 ($c=6$) olarak ifade etmektedirler. Ayrıca MPC geliştirilmiş bulanık indeksi tahmininde hem 6 ($c=6$) hem de 7 ($c=7$) olarak göstermektedir.

Çizelge 4.190. SDSXN veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.331	272.163	233.274	0.297	0.637	0.486
c=3	0.180	147.937	135.169	0.385	0.681	0.508
c=4	0.140	115.972	107.438	0.414	0.710	0.539
c=5	0.149	123.053	112.395	0.367	0.771	0.611
c=6	0.129	107.162	98.888	0.383	0.777	0.610
c=7	0.282	234.896	196.284	0.400	0.777	0.588
c=8	0.248	206.913	177.114	0.429	0.768	0.561

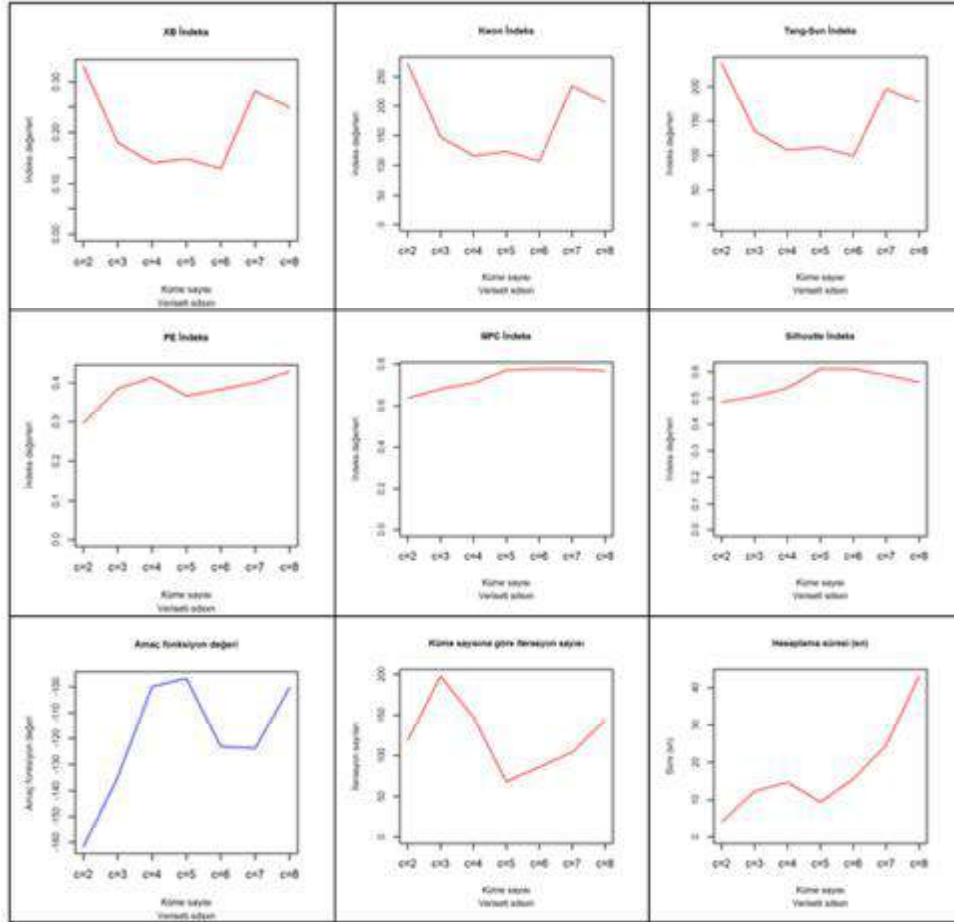
Çizelge 4.191. SDSXN veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.355	292.171	250.408	10.482	3.242
c=3	0.196	161.795	147.816	7.743	2.342
c=4	0.158	130.144	120.546	5.933	1.786
c=5	0.175	145.177	132.563	4.910	1.475
c=6	0.157	129.886	119.815	4.324	1.299
c=7	0.362	300.484	250.985	4.238	1.273
c=8	0.327	271.264	232.095	4.039	1.213

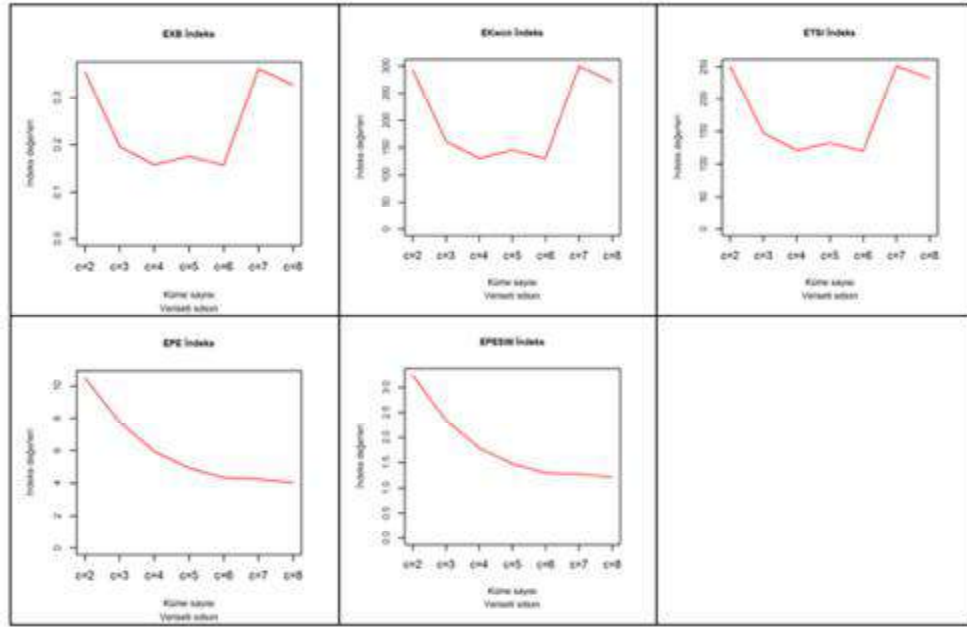
Çizelge 4.191'deki indeks değerleri de genel tahminlerini 6 (c=6) olarak göstermektedir. Çizelge 4.192'de elde edilen başarımların değerlerinde iterasyon sayı değeri en yüksek değer gerçek küme sayısını işaret etmektedir ki bu istenmeyen durumdur.

Çizelge 4.192. SDSXN veri seti için UOBKO algoritması başarıımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	118	4.150	-161.726
c=3	198	12.340	-134.991
c=4	147	14.630	-99.909
c=5	68	9.440	-96.798
c=6	86	15.560	-123.127
c=7	104	24.490	-123.821
c=8	143	43.140	-100.308



Şekil 4.123. UOBKO algoritması küme geçerlilik indeksleri ve kümeleme başarıımı (SDSXN)



Şekil 4.124. UOBKO algoritması uzatılmış indeks değerleri (SDSXN)

4.6.2. Gerçek Veri Setleri

4.6.2.1. Gerçek Veri Seti 1 (RDS1-Iris)

RDS1-Iris gerçek veri setine uygulanan karma kümeleme algoritması ile elde edilen bulanık indeks değerlerinin sonuçları Çizelge 4.193'de gösterilmektedir. Değerlerin isabetli tahmininde ($c=2$) bulunması UOBKO algoritmasının başarılı olduğunu göstermektedir.

Çizelge 4.193. RDS1-Iris veri setinde UOBKO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.057	8.848	8.824	0.057	0.938	0.850
c=3	0.169	26.954	24.648	0.154	0.869	0.732
c=4	0.325	52.302	42.844	0.225	0.840	0.671
c=5	1.414	227.638	114.582	0.349	0.769	0.604
c=6	1.211	203.821	108.750	0.440	0.713	0.494
c=7	1.325	230.479	113.911	0.447	0.728	0.518

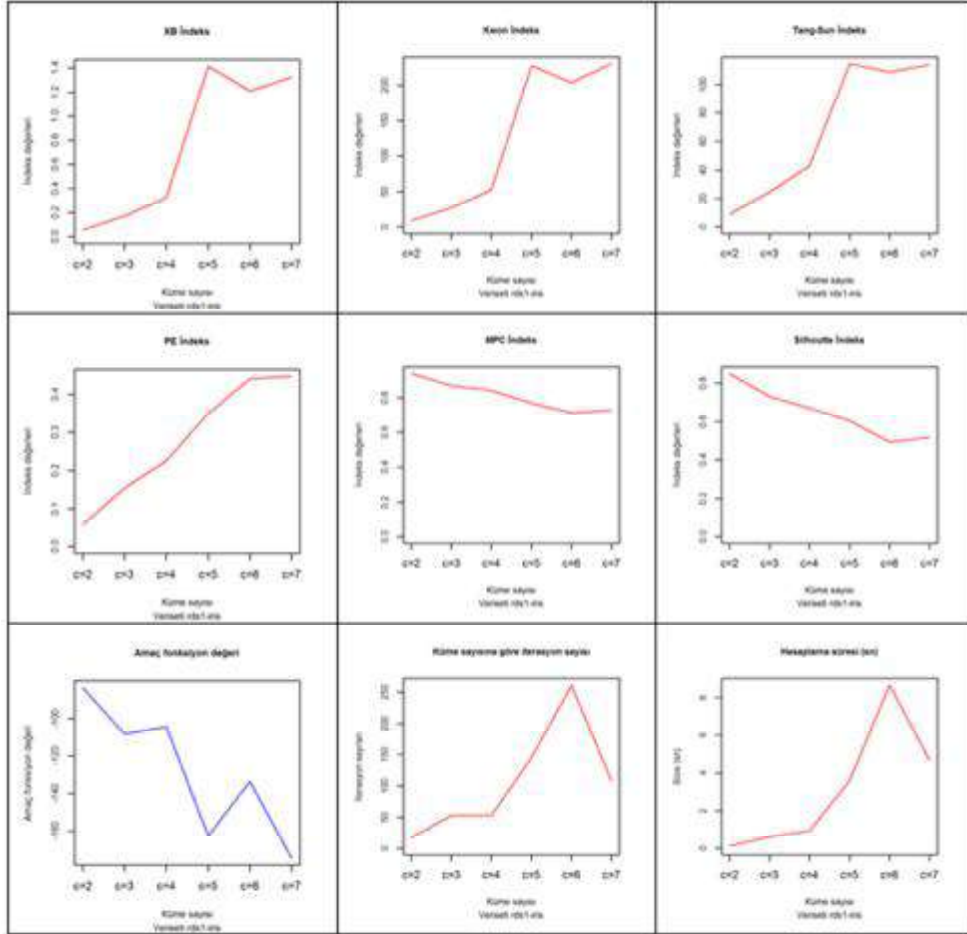
Çizelge 4.194. RDS1-Iris veri setinde UOBACO algoritması ile küme sayısına göre uzatılmış indeks değerleri

<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.069	10.535	10.460	1.640	0.500
c=3	0.249	38.846	35.218	1.842	0.565
c=4	0.561	87.688	71.202	1.918	0.584
c=5	2.798	435.253	217.324	2.012	0.611
c=6	2.713	429.032	226.357	1.839	0.556
c=7	3.413	543.572	265.188	1.801	0.543

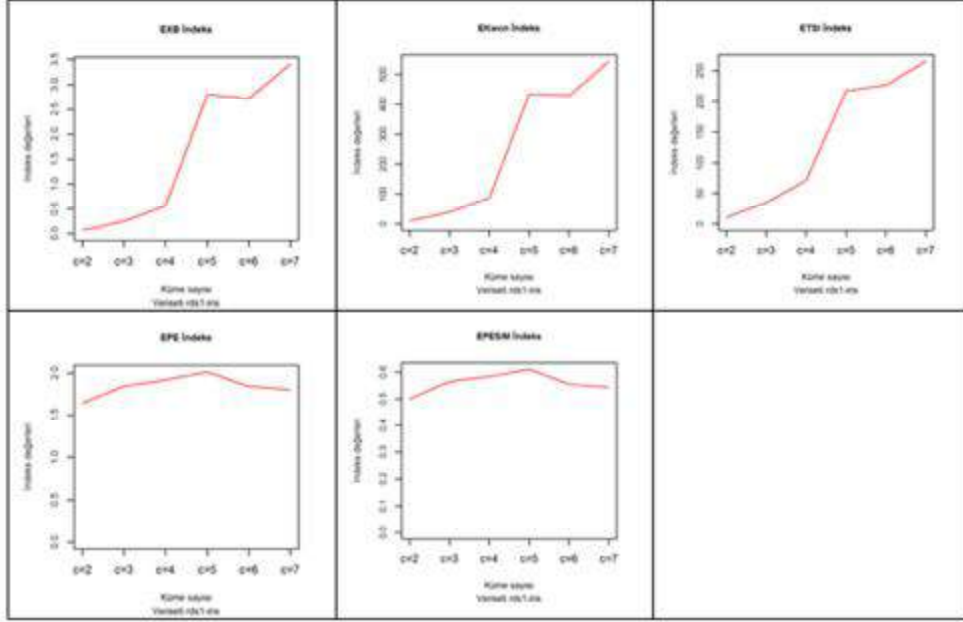
RDS1-Iris gerçek veri setine uygulanan karma kümeleme algoritmasının başarımı Çizelge 4.195’de gösterilmektedir. Aynı zamanda uzatılmış indeks değerlerinin isabetli tahminleri (c=2) Çizelge 4.194’te açıkça görülmektedir.

Çizelge 4.195. RDS1-Iris veri seti için UOBACO algoritması başarımı

<i>C</i>	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	17	0.110	-83.659
c=3	53	0.610	-108.210
c=4	53	0.940	-104.425
c=5	145	3.610	-162.486
c=6	261	8.660	-133.655
c=7	109	4.670	-174.601



Şekil 4.125. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS1-Iris)



Şekil 4. 126. UOBCO algoritması uzatılmış indeks değerleri (RDS1-Iris)

4.6.2.2. Gerçek Veri Seti 2 (RDS2-Quail)

RDS2-Quail gerçek veri setinde indeksler tarafından küme sayısı 3 ($c=3$) olarak isabet etmesi beklenmektedir. Çizelge 4.196'da V_{MPC} dışındaki tüm indeks değerlerine göre küme sayısının 2 olduğu önerilmiştir. UOBCO algoritmasının bu duruma çok yakın bir başarı gösterdiği yani genel anlamda uygulanan algoritmasının başarılı olduğundan bahsetmek mümkündür.

Çizelge 4.196. RDS2-Quail veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.083	7.754	8.003	0.098	0.888	0.772
c=3	0.103	10.221	10.692	0.142	0.880	0.749
c=4	0.197	20.711	21.695	0.149	0.896	0.728
c=5	0.153	17.684	18.650	0.167	0.888	0.706
c=6	0.326	38.638	40.446	0.208	0.870	0.630

Çizelge 4.197’de uzatılmış indeks değerlerinin tahminlerine göre genel tahmin 2 (c=2) olarak işaret edilmektedir. EPE ve EPESIM indeks değerleri RDS1-Iris’te göstermiş olduğu isabetli tahmini bu gerçek veri seti üzerinde göstermediği ve bu veri seti için etkin bir indeks olmadığını göstermektedir.

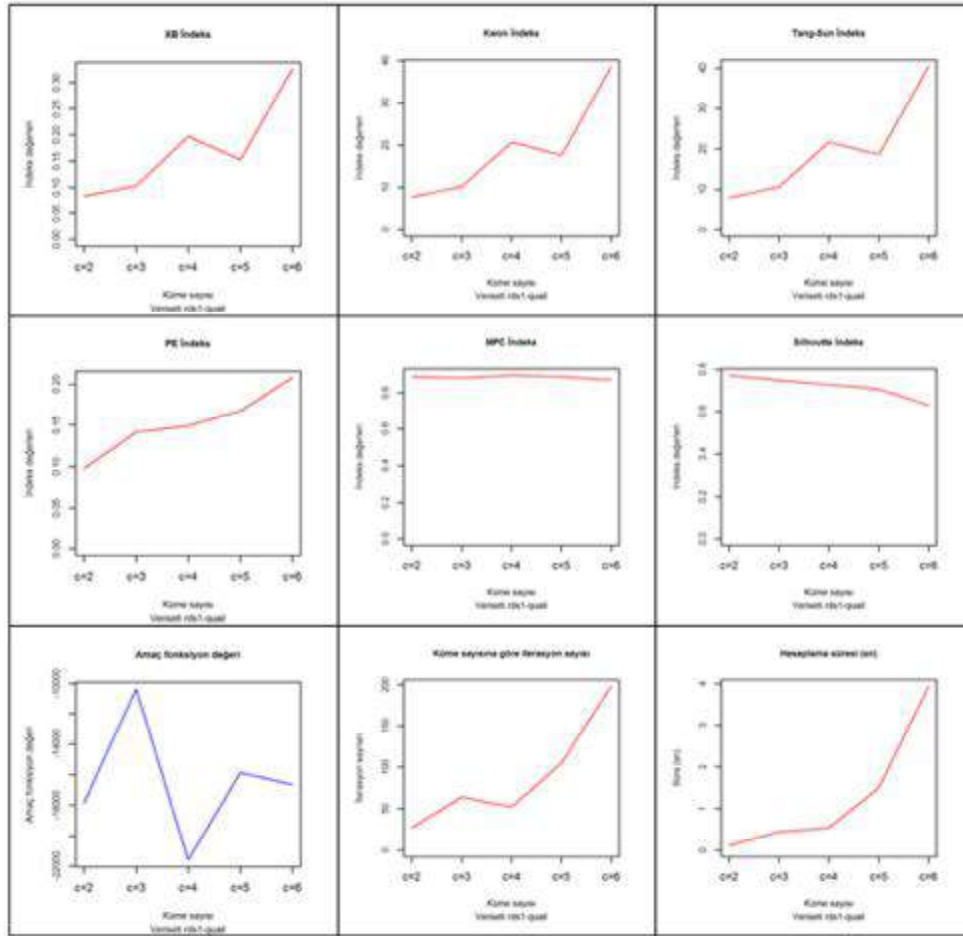
Çizelge 4.197. RDS2-Quail veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.095	8.789	9.037	1.243	0.392
c=3	0.136	13.215	13.685	1.123	0.345
c=4	0.307	30.609	31.588	1.060	0.323
c=5	0.289	29.901	30.860	1.010	0.306
c=6	0.733	75.309	77.072	1.030	0.312

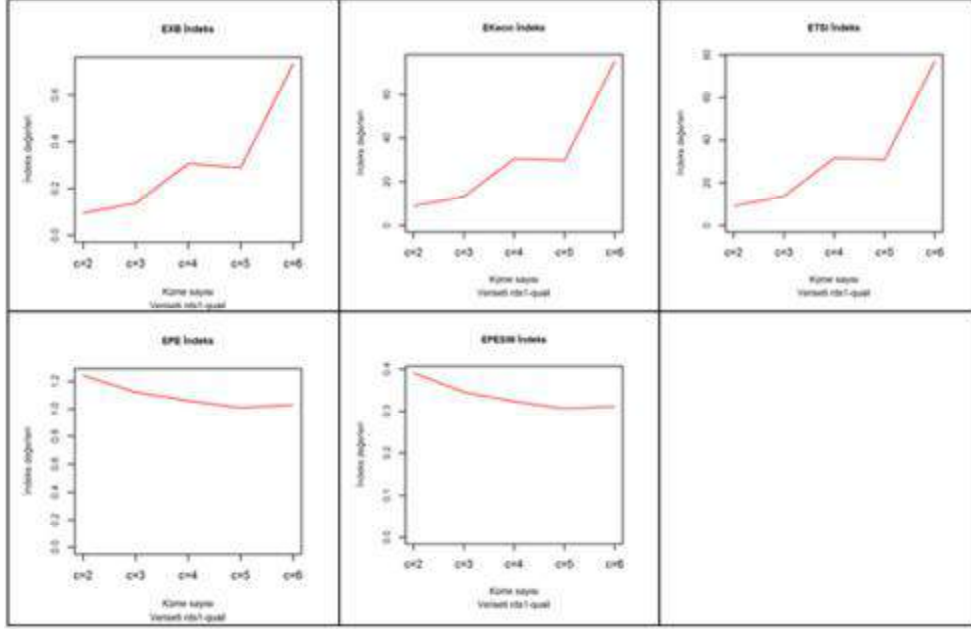
İndekslerin yani sıra uygulanan algoritmanın basarım sonuçları gerçek küme sayısına (c=3) yakın minimum iterasyon sayısı ve hesaplama sürelerini (Çizelge 4.198) 2 (c=2) olarak göstermektedirler.

Çizelge 4.198. RDS2-Quail veri seti için UOBCO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	27	0.110	-17838.615
$c=3$	64	0.440	-10365.054
$c=4$	52	0.540	-21583.740
$c=5$	105	1.500	-15857.471
$c=6$	198	3.940	-16629.209



Şekil 4. 127. UOBCO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS2-Quail)



Şekil 4.128. UOBOC algoritması uzatılmış indeks değerleri (RDS2-Quail)

4.6.2.3. Gerçek Veri Seti 3 (RDS3-Wine)

Çizelge 4.199 incelendiğinde Wine gerçek veri setinden elde edilen indeks tahminleri gösterilmektedir. İndeksler tahminlerini 2 ($c=2$) olarak işaret etmektedir. Gerçek küme sayısı ($c=3$) indekslerin tahminleri tarafından işaret edilmese de tahminlerin yakın olması durumu algoritmanın iyi bir basarım gösterdiği anlaşılmaktadır.

Çizelge 4.199. RDS3-Wine veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri

<i>C</i>	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.062	11.321	11.571	0.085	0.900	0.819
c=3	0.196	36.417	37.210	0.138	0.888	0.732
c=4	0.136	26.625	27.423	0.154	0.883	0.721
c=5	0.366	73.668	75.801	0.201	0.861	0.643
c=6	0.352	72.075	73.952	0.200	0.871	0.656
c=7	1.139	233.874	239.062	0.203	0.880	0.629

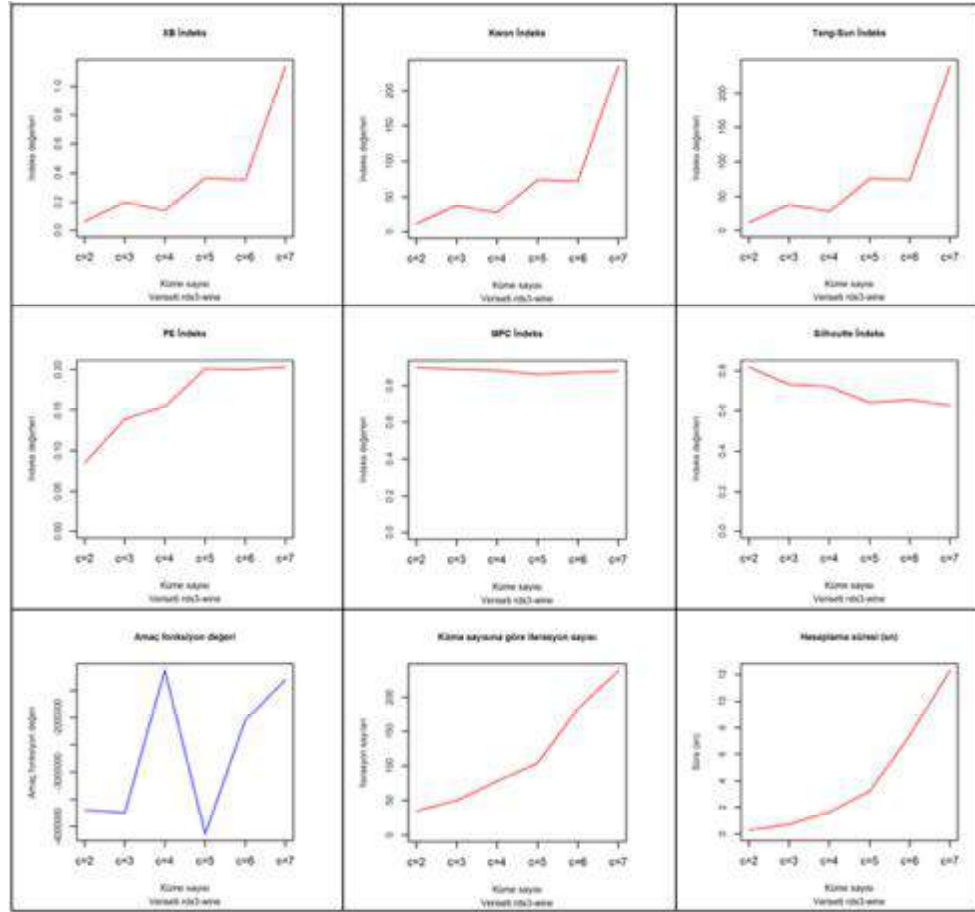
Çizelge 4.200’de gösterilen uzatılmış indekslerin tahminleri genel anlamda küme sayısını 2 (c=2) bulmaktadır. EPE ve EPESIM uzatılmış indeksleri tahminlerinde 4 (c=4)’u işaret ederek Quail veri setine göre gerçek kümeye daha yakın bir tahminde bulunmaktadır.

Çizelge 4.200. RDS3-Wine veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

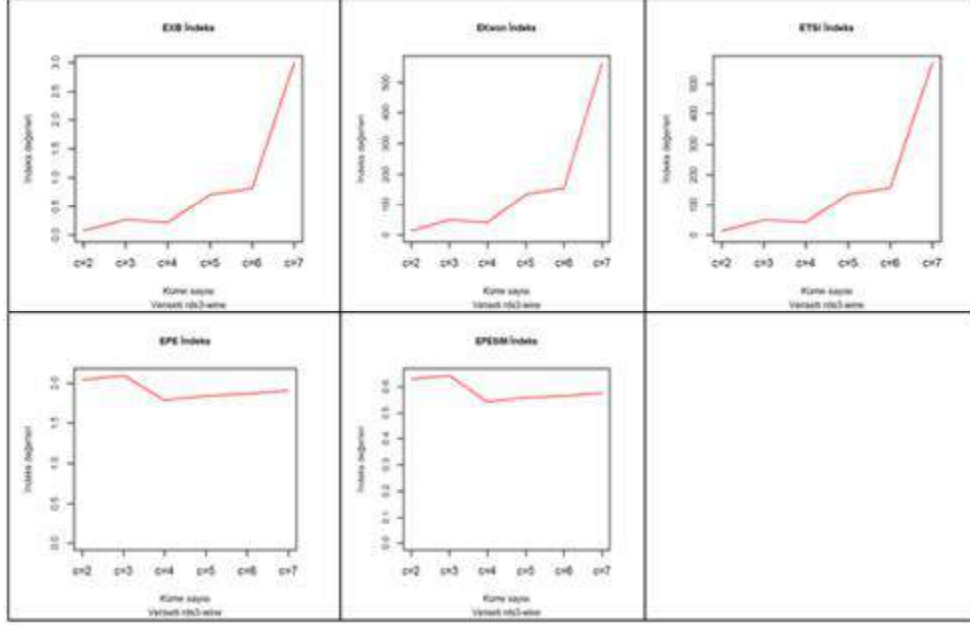
<i>C</i>	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.073	13.181	13.431	2.042	0.629
c=3	0.265	48.715	49.508	2.100	0.643
c=4	0.218	41.151	41.949	1.793	0.543
c=5	0.709	134.752	136.884	1.848	0.559
c=6	0.822	155.630	157.506	1.876	0.566
c=7	2.996	564.531	569.702	1.913	0.577

Çizelge 4.201. RDS3-Wine veri seti için UOBDO algoritması başarımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
$c=2$	35	0.270	-3689982.366
$c=3$	50	0.700	-3744166.185
$c=4$	78	1.620	-1136252.964
$c=5$	104	3.220	-4131479.141
$c=6$	183	7.390	-2051369.882
$c=7$	238	12.320	-1304830.336



Şekil 4. 129. UOBDO algoritması küme geçerlilik indeksleri ve kümeleme başarımı (RDS3-Wine)



Şekil 4.130. UOBCO algoritması uzatılmış indeks değerleri (RDS3-Wine)

4.6.2.4. Gerçek Veri Seti 4 (RDS4-Glass)

RDS4-Glass gerçek veri setine uygulanan karma kümeleme algoritması ile elde edilen bulanık indeks değerlerinin sonuçları Çizelge 4.202’de gösterilmektedir. Gerçek küme sayısı 6 ($c=6$) olan bu veri setinde değerlerin PE indeksinin tahmini ($c=2$) dışında tahminlerinde 3 ($c=3$)’u göstermesi UOBCO algoritmasının bu veri seti üzerinde iyi bir basarım gösterdiği söylenememektedir.

Çizelge 4.202. RDS4-Glass veri setinde UOBCO algoritması ile küme sayısına göre indeks değerleri

C	V_{XB}	V_{Kwon}	V_{TS}	V_{PE}	V_{MPC}	V_{Fsil}
c=2	0.199	42.812	41.574	0.142	0.829	0.688
c=3	0.122	26.532	26.191	0.169	0.864	0.745
c=4	3.151	691.274	436.832	0.493	0.613	0.412
c=5	1.30e+14	2.88e+16	1189.385	1.007	0.274	0.000
c=6	12.531	2862.584	878.550	0.628	0.569	0.318
c=7	9.929	2285.397	856.210	0.619	0.594	0.367

Çizelge 4.203'de uzatılmış indeks değerlerinin tahminlerine göre tüm tahminler 3 (c=3) olarak işaret edilmektedir. Yani kullanılan uzatılmış hiçbir indeks bu veri seti için etkin indeksler olmadığını göstermektedir.

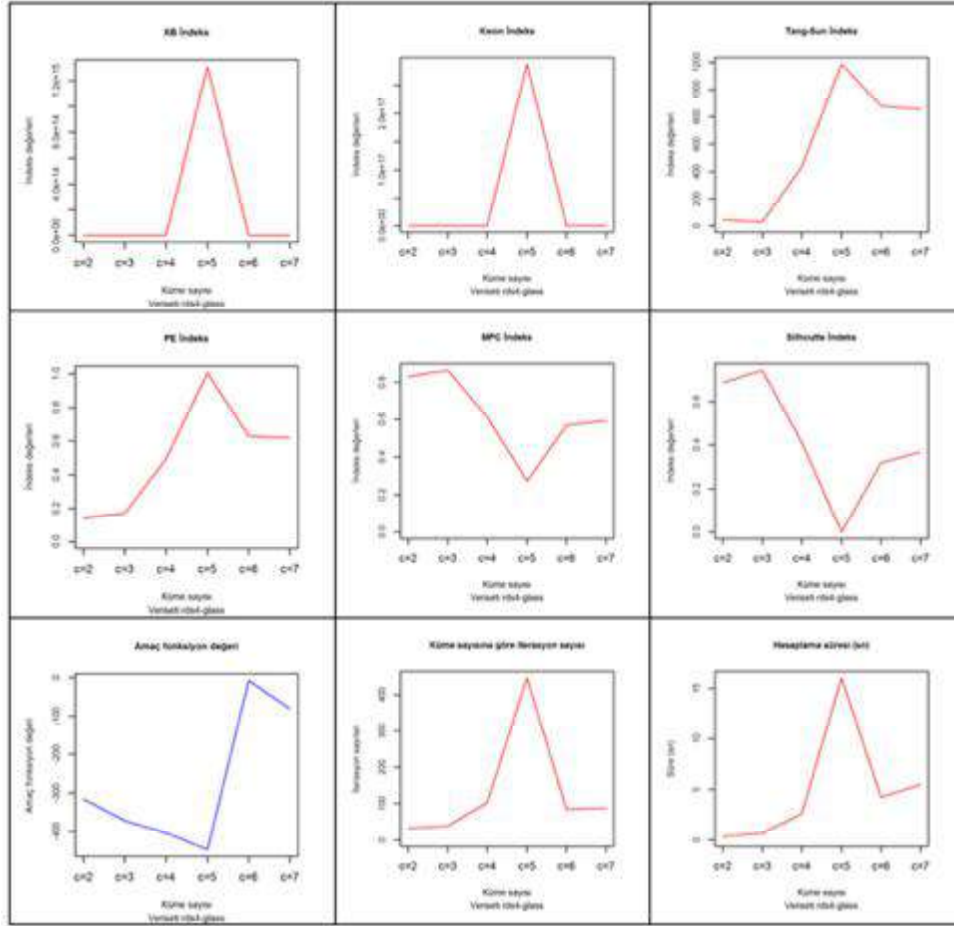
Çizelge 4.203. RDS4-Glass veri setinde UOBCO algoritması ile küme sayısına göre uzatılmış indeks değerleri

C	V_{EXB}	V_{EKwon}	V_{ETS}	V_{EPE}	V_{EPESIM}
c=2	0.214	46.148	44.795	2.252	0.702
c=3	0.134	29.207	28.809	1.436	0.436
c=4	3.874	845.976	533.804	2.059	0.632
c=5	1.75e+14	3.84e+16	1584.174	2.263	0.700
c=6	16.346	3678.937	1125.968	1.819	0.550
c=7	12.885	2918.043	1090.482	1.519	0.458

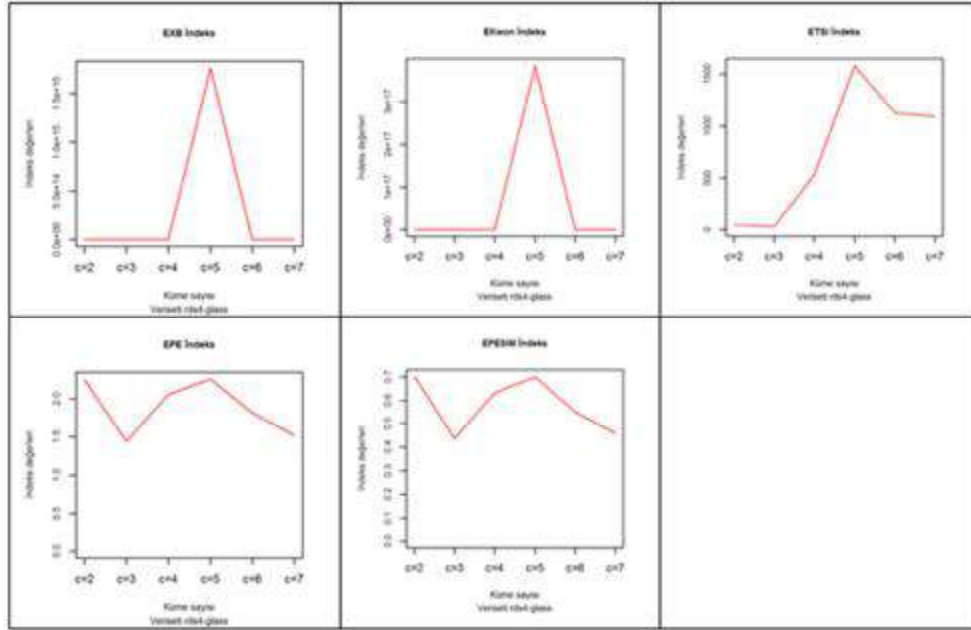
Veri seti üzerinde uygulanan algoritmanın basarım sonuçları Çizelge 4.204'te gerçek küme sayısından (c=6) uzak minimum iterasyon sayısı ve hesaplama sürelerini 2 (c=2) olarak göstermektedirler.

Çizelge 4.204. RDS4-Glass veri seti için UOBEO algoritması başarıımı

C	İterasyon sayısı	Hesaplama Süresi	Fonksiyon değeri
c=2	32	0.320	-316.424
c=3	39	0.660	-371.729
c=4	103	2.600	-404.487
c=5	446	16.030	-448.025
c=6	85	4.220	-7.770
c=7	88	5.470	-80.591



Şekil 4.131. UOBEO algoritması küme geçerlilik indeksleri ve kümeleme başarıımı (RDS4-Glass)



Şekil 4.132. UOBCO algoritması uzatılmış indeks değerleri (RDS4-Glass)

5. TARTIŞMA VE SONUÇLAR

Çizelge 5.1’de önceki bölümde açıklanan bulanık, uzatılmış ve genelleştirilmiş indekslere göre sentetik gürültüsüz veri setleri için önerilen küme sayıları görülmektedir. İncelenen indekslerden PE ve uzatılmış versiyonu olarak EPE’nin hemen algoritma sonuçlarına göre farklı tepki vermediği ve küme sayısını tüm veri setleri için 2 olarak önermiş ve bu nedenle çalışılan veri setlerinde kümeleme doğruluğunu ortaya koymada yetersiz kaldığı anlaşılmaktadır. Diğer indeksler üzerinden elde edilen sonuçlara göre BCO, OCO, BOCO, OBCO, İOCO ve UOBOCO algoritmalarının kümeleme başarımları açısından belirgin farklılıklar olmadığı anlaşılmaktadır. Ancak iki konkav ve yuvarlak şekilli üç küme içeren SDSX veri set bakımından algoritmaların farklı sonuçlar ürettiği görülmektedir. Bu veri seti için OCO algoritması ile 2 küme; İOCO algoritması ile 4 küme ve diğer algoritmalarla 5’er küme saptanmıştır. Bu durumun, tüm algoritmalarda yuvarlak, küresel kümelere üretmeye eğilimli bir uzaklık ölçüsü olarak Öklit uzaklığı kullanılmasından kaynaklanmış olabileceği düşünülmektedir. Böylesi durumlarda genel olarak Öklit yerine küme şekillerini de dikkate alan Manhattan uzaklığı gibi uzaklık ölçülerinin veya Beklenti Çoklaması (Expectation Maximization) gibi algoritmaların kullanılması tavsiye edilmektedir.

Çizelge 5.1. Sentetik gürültüsüz veri setlerinde algoritmalara göre küme sayıları

Veri seti	Küme sayısı	BCO	OCO	BOCO	OBCO	İOCO	UOBCO
SDSC	4	4	4	4	4	3	4
SDSE	4	4	4	4	4	4	4
SDSR	4	4	4	4	4	4	4
SDSX	3	5	2	5	5	4	5

Çizelge 5.2. Sentetik gürültüsüz veri setlerinde algoritmalara göre iterasyon sayıları ve hesaplama süreleri (saniye)

Veri seti	BCO		OCO		BOCO		OBCO		İOCO		UOBCO	
	İS	HS	İS	HS	İS	HS	İS	HS	İS	HS	İS	HS
SDSC	32	1.000	81	4.080	36	1.990	15	1.490	16	1.470	25	2.360
SDSE	54	1.330	177	3.050	56	3.050	26	1.560	11	1.600	42	4.630
SDSR	31	0.750	101	2.110	38	2.000	18	1.830	13	1.670	26	2.540
SDSX	63	1.910	35	0.410	55	3.810	42	5.530	15	1.330	45	3.150

Kümeleme başarımlarını değerlendirmek üzere Çizelge 5.1’de verilen küme sayılarında kümeleme sonuçlarını elde etmek için gereken iterasyon sayıları (İS) ve hesaplama süreleri (HS) Çizelge 5.2’de listelenmiştir. Elde edilen sonuçlara göre İOCO algoritması kümeleme analizini veri setlerine göre 11-16 gibi çok az sayıda iterasyonla tamamlarken OCO algoritması 81-177 iterasyonla en yüksek iterasyon sayısına sahip algoritma olmuştur. Hesaplama süresi açısından değerlendirme yapıldığında ise BCO’nun son veri seti (SDX) hariç diğerlerinde en kısa işlem zamanı gerektirdiği görülmektedir.

Kümeleme algoritmaları sentetik gürültülü veri setlerinde de benzer başarımlar göstermişlerdir. BCO, OCO, BOCO, OBCO, İOCO ve UOBCO yuvarlak, elips ve kare şekilli kümeler içeren veri setlerinde aynı kümeleme sonucu vermişlerdir. Ancak İOCO algoritması kare şekilli kümeler içeren SDSRN veri setini 1 eksik küme ile 3 olarak tahmin etmiştir. Algoritmalara gürültüsüz SDSX veri setinde olduğu gibi SDSX veri seti için de aynı sonuçları vermiş, ancak UOBCO küme sayısını 6 olarak daha yüksek tahmin etmiştir.

Gürültülü veri setleri için yapılan kümelemeler bakımında algoritmaların ihtiyaç duyduğu iterasyon sayıları ve hesaplama süreleri de gürültüsüz veri setleri için elde edilen sonuçlara benzerlik göstermektedir. BCO, SDSXN veri seti hariç tüm veri setlerinde en kısa hesaplama süresine sahip algoritma durumundadır. Ancak SDSXN veri setinde en kısa hesaplama OCO tarafından gerçekleştirilmiştir. İterasyon sayısı bakımından en iyi sonuçları 12-16 arasında değişen iterasyon sayısı ile İOCO üretmiş, bunu OBCO algoritması izlemiştir.

Çizelge 5.3. Sentetik gürültülü veri setlerinde algoritmalara göre küme sayıları

Veri seti	Küme sayısı	BCO	OCO	BOCO	OBCO	İOCO	UOBCO
SDSCN	4	4	4	4	4	4	4
SDSEN	4	4	4	4	4	4	4
SDSRN	4	4	4	4	4	3	4
SDSXN	3	5	2	5	5	4	6

Çizelge 5.4. Sentetik gürültülü veri setlerinde algoritmalara göre iterasyon sayıları ve hesaplama süreleri (saniye)

Veri seti	BCO		OCO		BOCO		OBCO		İOCO		UOBCO	
	İS	HS	İS	HS	İS	HS	İS	HS	İS	HS	İS	HS
SDSCN	32	0.810	70	1.540	32	1.740	15	1.540	14	1.730	24	2.300
SDSEN	37	0.920	92	2.110	43	2.080	22	2.160	13	1.860	30	2.900
SDSRN	31	0.770	80	1.880	32	1.670	16	1.610	12	1.450	26	2.590
SDSXN	96	3.130	36	0.450	84	5.850	63	4.270	16	2.250	68	4.150

Çizelge 5.5. Gerçek veri setlerinde algoritmalara göre küme sayıları

Veri seti	Küme sayısı	BCO	OCO	BOCO	OBCO	İOCO	UOBCO
İris	2,3	2	2	2	2	2	2
Quail	3	2	3	2	2	6	2
Wine	3	2	5	2	2	5	2
Glass	6	3	2	3	2	2	3

Çizelge 5.6. Gerçek veri setlerinde algoritmalara göre iterasyon sayıları ve hesaplama süreleri (saniye)

Veri seti	BCO		OCO		BOCO		OBCO		İOCO		UOBCO	
	İS	HS	İS	HS	İS	HS	İS	HS	İS	HS	İS	HS
İris	16	0.050	43	0.170	17	0.100	14	0.090	9	0.090	17	0.110
Quail	26	0.050	110	0.200	25	0.080	25	0.110	103	0.250	27	0.110
Wine	27	0.090	96	0.270	27	0.170	30	0.260	92	0.450	35	0.270
Glass	30	0.090	15	0.080	40	0.370	44	0.470	30	0.190	32	0.320

Gerçek veri setlerinden İris veri setinde küme sayısı tüm algoritmalar tarafından başarılı şekilde bulunmuştur. Quail veri setindeki küme sayısı OCO tarafından 3 olarak bulunmasına karşın BCO, BOCO, OBCO ve UOBCO algoritmaları küme sayısını bir eksik olmak üzere 2 olarak önermiş, ancak UOCO 6 gibi iki kat yüksek küme sayısı önermiştir. Wine veri setindeki kümelere BCO, BOCO, OBCO ve UOBCO tarafından bir eksikle 2 olarak, OCO ve İOCO tarafından 2 fazlasıyla 5 olarak bulunmuştur. Glass veri setindeki küme sayıları tüm algoritmalara için 2-3 eksik şekilde saptanmıştır. Bu veri setindeki küme sayısı BCO, BOCO ve UOBCO tarafından 3, OCO, OBCO ve İOCO tarafından ise 2 olarak önerilmiştir.

Hesaplama süresi açısından en iyi sonucu tüm veri setlerinde BCO vermiş, bunu Glass veri setindeki performansı ile OCO izlemiştir. İterasyon sayıları açısından BCO Wine veri setinde en az iterasyon sayısı verirken BOCO Quail ve Wine veri setlerinde daha az sayıda iterasyonla kümelemeyi tamamlamıştır. İOCO

algortması Iris veri setinde 9 iterasyon sayısı ile diğer algoritmalara göre daha iyi konumda yer almıştır.

Amaç fonksiyonlar her bir veri noktası ile küme merkezleri arasındaki mesafelerin genel toplamını gösteren değerlerdir. Bu değerler iyi kümeleme sonuçlarına yaklaşıırken küçülmekte ve uzaklaştıkça büyümektedirler. Bu nedenle amaç fonksiyonunun en küçük olduğu kümelemeyi en iyi kümeleme olarak değerlendirdiğimizde UOBKO algoritmasındaki Çizelge 4.171 ve Şekil 4.109'daki grafiğe göre SDSC için en iyi kümelemenin $c=8$ için elde edildiği görülecektir. Tüm algoritma sonuçları incelenip karşılaştırıldığında UOBKO algoritmasından elde edilen sonuçlar negatif değerli olduğundan hem sentetik hem de gerçek veri setleri için en iyi kümelemeyi yine UOBKO algoritması olduğunu söylemek mümkündür. Bunların yanı sıra amaç fonksiyon değerleri incelendiğinde amaç fonksiyon değerinin küme sayısının artışıyla genel bir azalma gösterdiği ve bununla birlikte algoritmaların işaret ettikleri küme sayılarının da en büyük fonksiyon değerlerine sahip oldukları da gözlemlenmektedir. Buna benzer bir diğer gözlem ise hesaplama süresi ve iterasyon sayısının minimum olduğu durumlarda genel anlamda maksimum fonksiyon değerinin hesaplandığı anlaşılmaktadır.

Sentetik ve gerçek veri setlerinde geçerlilik indeksleri ile yapılan kümeleme geçerliliklerine göre algoritmalar arasında farklılıklar bulunmadığı; ancak mevcut algoritmaların konkav ve yuvarlak şekilli kümeler içeren SDSX ve SDSXN veri setlerinde iyi çalışmadığı anlaşılmaktadır. Bu durumun mevcut algoritmalarla Öklit uzaklıklarının kullanılmasından kaynaklandığı ve bu uzaklık ölçüsünün yuvarlak şekilli kümeler üretmeye eğilimli olduğu bildirilmektedir. Hesaplama süresi açısından BKO algoritması daha iyi gözükmeyle birlikte bunun söz konusu algoritmada yalnızca üyelik dereceleri üretmesi dolayısıyla diğer algoritmalara göre doğal olarak daha az işlem yapmasından kaynaklandığı söylenebilir. İOKO ve OKO algoritmaları iterasyon sayısı ve bazı veri setlerinde hesaplama süresi açısından avantajlı gibi gözükmeyle birlikte bu algoritmaların BKO algoritmasından elde edilen üyelik dereceleri matrisine ihtiyaç duyması gibi

bir dezavantajı bulunmaktadır. Ayrıca literatüre göre çakışan kümeler üretebilmektedirler. Karma algoritmalarından OBCO ve BOCO algoritmaları benzer başarımlara sahip olmakla birlikte BOCO algoritmasının nispeten daha iyi sonuçlar verdiği görülmüştür. Ek Bölümünde yer alan grafikler algoritmaların performanslarını gürültü ve aykırı değerler bakımından karşılaştırılmasında daha hassas yorumlamalar yapılmasını sağlayabilecektir. Bu grafiklerde T_{max} ve $T > 0.5$ sırasıyla maksimum tipiklik dereceleri ve 0.5'ten büyük dolayısıyla önemli sayılabilecek tipiklik derecelerine göre küme yapılarını göstermektedir. Burada örneğin Ek Şekil 5.3-5.4'te İOCO ve Ek Şekil 6.3-6.4'de UOBCO algoritmaları için elips şekilli kümeler içeren SDSE ve SDSN veri setleri için T_{max} ve $T > 0.5$ grafikleri incelendiğinde İOCO'nun 0.5'ten büyük tipiklik derecesinde çok az sayıda nesne gösterdiği, ancak UOBCO'nun buna göre daha iyi çalıştığı anlaşılmaktadır.

Sonuç olarak gerek daha önce çalıştırılan bir BCO algoritmasından üyelik dereceleri matrisi istememesi, gerek gürültülü ortamlarda iyi çalışması ve gerekse diğerlerine göre benzer çalışma performansı göstermesi açısından UOBCO algoritmasının kullanılması önerilmektedir.

KAYNAKLAR

- Adekunle, Y.A., Alao, O.D., Seun, E., Jerry, S., Jean-Paul, A., 2014. A New Clustering Validity Index for Fuzzy C Means Algorithm Based on Measure Of Disparity. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* Volume 3 Issue 9.
- Alameluman, N., & Devishree, J., 2012. Automated Clustering of Cancer Cells Using Fuzzy C Means with Repulsions in Ultrasound Images. *J. of Artificial Intelligence Journal of Artificial Intelligence*, 5(1), 14-25. doi:10.3923/jai.2012.14.25
- Ali, M.A., Karmakar, G.C., Dooley, L.S., 2008. Review on Fuzzy Clustering Algorithms. *IETECH Journal of Advanced Computations*, vol. 2, no. 3, pp. 169 – 181.
- Askari, S., Montazerin, N., Zarandi, M., H., F., 2017. Generalized Possibilistic Fuzzy C-Means with novel cluster validity indices for clustering noisy data. *Applied Soft Computing*, 53, 262-283. doi:10.1016/j.asoc.2016.12.049.
- Anderson, E., 1935. The iris of the Gaspe peninsula. *Bulletin American Iris Society*, vol.59, pp. 2-5.
- Barni, M., Cappellini V., Mecocci, A., 1996. Comments on “A Possibilistic Approach to Clustering. *IEEE Trans. Fuzzy Systems*, vol. 4, pp. 393-396.
- Bezdek, J. C., 1974a. Cluster validity with fuzzy sets. *J. Cybernet.* 3: 58–74.
- Bezdek, J. C., 1974b. Numerical taxonomy with fuzzy sets. *J. Math. Biol.* 1: 57–71.
- Bezdek, J. C., 1981. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, New York.
- Bezdek, J. C., Ehrlich, R., Full, W., 1984. FCM: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3), 191-203. doi:10.1016/0098-3004(84)90020-7.

- Bezdek, J. C., Pal, N. R., 1998. Some New Indexes of Cluster Validity. *IEEE Transactions on System, Man, and Sybernetics, Part B*, vol. 28, no.3, pp. 301-315.
- Biju, V., & Mythili, P., 2015. Fuzzy Clustering Algorithms for cDNA Microarray Image Spots Segmentation. *Procedia Computer Science*, 46, 417-424. doi:10.1016/j.procs.2015.02.039
- Bora, D.J., Gupta, A.K., 2014. A Comparative study Between Fuzzy Clustering Algorithm and Hard Clustering Algorithm. *Int. J. of Computer Trends and Technology*, vol. 10, no. 2, pp. 108-113.
- Borgelt, C., Kruse, R., 2005. Fuzzy and Probabilistic Clustering with Shape and Size Constraints. *Proc. of the 11th Int. Fuzzy Systems Association World Congress*, Beijing, China, pp. 945-950.
- Cano, C., Adarve, L., López, J., & Blanco, A., 2007. Possibilistic Approach For Biclustering Microarray Data. *Computers in Biology and Medicine* 37.10, 1426-1436.
- Cebeci Z., Yıldız F., 2015. Bulanık C-Ortalamlar Algoritmasının Farklı Küme Büyüklükleri için Hesaplama Başarımı ve Kümeleme Geçerliliğinin Karşılaştırılması. *9. Ulusal Zootekni Bilim Kongresi*, 3-5 Eylül 2015, Konya – Türkiye. Bildiriler Kitabı, p. 227-239.
- Cebeci Z., Yıldız F., 2015. Comparison Of K-Means And Fuzzy C-Means Algorithms On Different Cluster Structures. *Journal of Agricultural Informatics*, vol.6, no.3.
- Cebeci, Z., Yıldız, F., Kayaalp, G.T., 2015. K-Ortalamlar Kümelemesinde Optimum K Değeri Seçilmesi. *2. Ulusal Yönetim Bilişim Sistemleri Kongresi*, 8-10 Ekim 2015, Erzurum. Bildiriler Kitabı (Ed: Ü. Özen ve ark.), s. 231-242. Orka Ofset Matbaacılık, Erzurum, ISBN:978-975-442-738-7.

- Chattopadhyay S., Pratihari D.K., 2011. A Comparative Study of Fuzzy C-Means Algorithm and Entropy Based Fuzzy Clustering Algorithms. *Computing and Informatics*, vol.30, no. 1, pp. 701-720.
- Correa, C., Valero, C., Barreiro P., Diago, M.P., Tardaguila, J., 2011. A Comparison of Fuzzy Clustering Algorithms Applied to Feature Extraction on Vineyard. E.T.S.I. Agronomos. Departamento de Ingenieria Rural Universidad Politecnica de Madrid..
- Dave, R.N., 1996. Validating fuzzy partition obtained through c-shells clustering, *Pattern Recognition Lett.* 17: 613–623.
- Dave, R. N. , Krishnapuram, R., 1997. Robust clustering methods: a unified view. *IEEE Trans. Fuzzy Systems*, vol. 5 no. 2, pp.270-293.
- Desgraupes, B. (2016). clusterCrit: Clustering Indices. R package version 1.2.7. <https://CRAN.R-project.org/package=clusterCrit>
- Di Martino, F., Loia, V., Sessa, S., 2007. Extended Fuzzy C-means Clustering Algorithm for Hotspot Events in Spatial Analysis. *Int. J of Hybrid Intelligent Systems*, no. 4, pp. 1–14.
- Di Martino, F., Sessa, S., 2009. Implementation of the Extended Fuzzy C-Means Algorithm in Geographic Information Systems. *J. of Uncertain Systems*, vol. 3, no. 4, pp. 298-306.
- Dong, W, Ren, J.D., Zhang, D., 2011. Hierarchical K-Means Clustering Algorithm Based on Silhouette and Entropy. H. Deng ve ark. (Eds): AICI 2011, Part I, LNAI vol. 7002, pp. 339-347. Springer-Verlag Berlin, Heidelberg.
- Döring, C., Lesot, M-J., Kruse, R., 2006. Data analysis with fuzzy clustering methods. *Computational Statistics & Data Analysis*, vol. 51, pp. 192 –214.
- Duncan M. ve Chow, E.D. (2013). ellipse: Functions for drawing ellipses and ellipse-like confidence regions. R package version 0.3-8. <https://CRAN.R-project.org/package=ellipse>

- Dunn, J.C., 1973. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *J. of Cybernetics*, vol.3, no.3, pp. 32-57.
- El-Melegy, M., Zanaty E. A., Abd-Elhafiez W.A., Farag A., 2007. On Cluster Validity Indexes In Fuzzy and Hard Clustering Algorithms For Image Segmentation. [IEEE International Conference on/](#) DOI: [10.1109/ICIP.2007.4379507](https://doi.org/10.1109/ICIP.2007.4379507).
- Ferraro, M.B., Giordani, F., 2015. A Toolbox for Fuzzy Clustering Using the R Programming Language. *Fuzzy Sets and Systems* (in press), <http://dx.doi.org/10.1016/j.fss.2015.05.001>
- Filippone, M., Masulli, F., & Rovetta, S., 2010. Applying the Possibilistic c-Mean Algorithm in Kernel-Induced Spaces. *IEEE Trans. Fuzzy Syst. IEEE Transactions on Fuzzy Systems*, 18(3), 572-584. doi:10.1109/tfuzz.2010.2043440.
- Fisher, R., 1936. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, vol.7, no. 2, pp. 179-188.
- Fu, L., Medico, E., 2007. FLAME, a Novel Fuzzy Clustering Method for the Analysis of DNA Microarray Data. *BMC Bioinformatics*, vol. 8, no. 1, pp. 3.
- Ghosh, S., Dubey, S.K., 2013. Comparative Analysis of K-Means and Fuzzy C-Means Algorithms. *Int. J. Advanced Computer Science and Applications*, vol. 4, no.4, pp. 35-39.
- Gionis, A., Mannila, H., Tsaparas, P., 2007. Clustering Aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 1, no.1, pp. 1-30.
- Gomathi, M., & Thangaraj, P., 2010. A Parameter Based Modified Fuzzy Possibilistic C-Means Clustering Algorithm for Lung Image Segmentation. *Global Journal of Computer Science and Technology*, 10(4), 85-91.

- Halkidi, M., Batistakis, Y., and Vazirgiannis., M., 2001. On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, 17.2/3, 107-145.
- Hogo, M.A., 2010. Evaluation of E-Learners Behaviour Using Different Fuzzy Clustering Models: A Comparative Study. (*IJCSIS*) *International Journal of Computer Science and Information Security*, Vol. 7, No. 2. 2010.
- Jain, A. K., Dubes, R. C., 1988. *Algorithms for clustering data*. Prentice Hall, New Jersey.
- Jain, A, Law, M., 2005. Data Clustering: A User's Dilemma. *Lecture Notes in Computer Science*, 3776, pp. 1-10.
- Ji, Z., Xia, Y., Sun, Q., & Cao, G., 2013. Interval-valued possibilistic fuzzy C-means clustering algorithm. *Fuzzy Sets and Systems* 253, 138-156.
- Jipkate, B.R., Gohokar, V.V., 2012. A Comparative Analysis of Fuzzy C-Means Clustering and K Means Clustering Algorithms. *Int. J. of Computational Engineering*, vol. 2, no. 3, pp. 737-739.
- Joopudi, Sreeram et al. 2013. A New Cluster Validity Index For Fuzzy Clustering. *IFAC Proceedings Volumes* 46.32, 325-330. Mumbai, India.
- Kaur, M., Kaur, U., 2013. Comparison Between K-Mean and Hierarchical Algorithm Using Query Redirection. *Int. J. of Advanced Research in Computer Science and Software Engineering*, vol. 3 , no. 7, pp. 1454-1459.
- Krishnan, P. H., Ramamoorthy, P., 2012. An Efficient Modified Fuzzy Possibilistic C-Means Algorithm for MRI Brain Image Segmentation. *International Journal of Engineering Research and Applications*, 2(2), 1106-1110.
- Krishnapuram, R., Keller, J., 1993. A possibilistic approach to clustering. *IEEE Trans. Fuzzy Systems*, vol. 1, pp. 98-110.
- Krishnapuram, R., Keller, J., 1996. The Possibilistic C-Means algorithm: Insights and recommendations. *IEEE Trans. Fuzzy Systems*, vol. 4, pp. 385-393.

- Kumari N, Sharma B., Gaur D., 2012. Implementation of Possibilistic Fuzzy C-Means Clustering Algorithm in Matlab. *International Journal of Scientific & Engineering Research*, Volume 3, Issue 11, November-2012 1 ISSN 2229-5518.
- Kwon, S.H., 1998. Cluster Validity Index for Fuzzy Clustering. *Electron. Lett.* 34 (22) : 2176–2178.
- Leisch, F. ve Dimitriadou, E. (2010). mlbench: Machine Learning Benchmark Problems. R package version 2.1-1.
- MacQueen, J.B., 1967. Some Methods for Classification and Analysis of Multivariate Observations. *Proc. of 5th Berkeley Symp. on Mathematical Statistics and Probability*, Berkeley, University of California Press, pp. 281-297.
- Madhukumar, S., Santhiyakumari, N., 2015. Evaluation of K-Means and Fuzzy C-means Segmentation on MR Images of Brain. *The Egyptian J. of Radiology and Nuclear Medicine*, vol. 46, no. 2, pp. 475–479.
- Martino, F.D., Sessa, S., 2009. Implementation of the Extended Fuzzy C-Means Algorithm in Geographic Information Systems. *Journal of Uncertain Systems* Vol.3, No.4, pp.298-306.
- Filippone, M., Masulli, F., & Rovetta, S., 2010. Applying the Possibilistic c-Mean Algorithm in Kernel-Induced Spaces. *IEEE Trans. Fuzzy Syst. IEEE Transactions on Fuzzy Systems*, 18(3), 572-584. doi:10.1109/tfuzz.2010.2043440
- Menárd, M., Courboulay, V., Dardignac, P., 2003. Possibistic and Probabilistic Fuzzy Clustering: Unification within the Framework of the Non-Extensive Thermostatistics. *Pattern Recognition*, vol. 36, no. 6, pp. 1325-1342.
- Quintanilla-Dominguez, J. et al., 2011. Image Segmentation By Fuzzy And Possibilistic Clustering Algorithms For The Identification Of Microcalcifications. *Scientia Iranica* 18.3, 580-589.

- Pakhira, M. K., Bandyopadhyay, S., and Maulik, U., 2004. Validity Index For Crisp And Fuzzy Clusters. *Pattern Recognition* 37.3 487-501.
- Pal, N.R. and J.C. Bezdek., 1995. On Cluster Validity For The Fuzzy C-Means Model. *IEEE Transactions on Fuzzy Systems*. 3.3. 370-379.
- Pal N.R., Pal K., Keller J.M., Bezdek J.C., 2005. A Possibilistic Fuzzy C-Means Clustering Algorithm. *IEEE Transactions On Fuzzy Systems*, Vol. 13, No. 4.
- Pal, N. R., Pal, K., Bezdek J. C., 1997. A mixed c-means clustering model. in *Proc. of the Sixth IEEE International Conference on Fuzzy Systems*, vol. 1, pp. 11-21.
- Panda, S, Sahu, S, Jena, P & Chattopadhyay, S., 2012. Comparing Fuzzy-C Means and K-Means Clustering Techniques: A Comprehensive Study. *Advances in Intelligent and Soft Computing*, vol. 166, pp. 451-460
- Pimentel, B. A., & Souza, R. M., 2013. A multivariate fuzzy c-means method. *Applied Soft Computing*, 13(4), 1592-1607. doi:10.1016/j.asoc.2012.12.024.
- R Core Team 2015 “R: A Language and Environment for Statistical Computing”. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.r-project.org>.
- Ramesh, D., Vardhan, B.V., 2013. Data Mining Techniques and Applications to Agricultural Yield Data. *Int. J. of Advanced Research in Computer and Communication Engineering*, vol. 2, no. 9, pp. 3477-3480.
- Rezaee, B., 2010. A Cluster Validity Index For Fuzzy Clustering. *Fuzzy Sets and Systems* 161.23 3014-3025.
- Ramze Rezaee, M., B.P.F. Lelieveldt, and J.H.C. Reiber., 1998. A New Cluster Validity Index For The Fuzzy C-Mean. *Pattern Recognition Letters* 19.3-4, 237-246. Web.

- Rawashdeh M., Ralescu A., 2012. Fuzzy Cluster Validity with Generalized Silhouettes. School of Computing Sciences and Informatics University of Cincinnati Cincinnati, USA.
- Rosso G., 2014. Outliers Emphasis on Cluster Analysis. (diğer bilgiler bulunamadı)
- Rowlingson, R. ve Diggle, P. (2016). splancs: Spatial and Space-Time Point Pattern Analysis. R package version 2.01-39. <https://CRAN.R-project.org/package=splancs>
- Saad, M.F., Alimi, A.M., 2009. Modified Fuzzy Possibilistic C-Means. Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol IIMECS 2009, March 18 - 20, 2009, Hong Kong
- Saad, M.F., Alimi, A.M., 2012. Validity Index and number of clusters. IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 1, No 3, January 2012, Research Group on Intelligent Machines, University of Sfax, ENIS Sfax, 3038, Sfax, Tunisia.
- Shapira, O., Wachs, J. P., 2004. A Graphical method based on the Xie-Beni Validity index to improve the “Possibilistic C-Means with Repulsion”. Algorithm to appear in *IMS 2004*, Sakarya, Turkey.
- Sheshasayee, A., Sharmila, P., 2014. Comparative Study of Fuzzy C-means and K-means Algorithm for Requirements Clustering. *Indian J. of Science and Technology*, vol. 7, no 6, pp. 853–857.
- Singhal, R., Deepika, N., 2016. Classification of Words: Using PFCM Clustering. *International Journal of Computer Science and Mobile Computing*, Vol.5 Issue.4, April- 2016, pg. 114-117

- Singla, S., Khehra, B.S., 2016. A Comparative Study of K-Means, Fuzzy C-Means and Possibilistic Fuzzy C-Means Algorithm on Noisy Grayscale Images. *International Journal of Computer Applications (0975 – 8887) International Conference on Advances in Emerging Technology (ICAET 2016)*
- Sivarathri, S., Govardhan, A., 2014. Experiments on Hypothesis Fuzzy K-Means is Better Than K-Means for Clustering. *Int. J. Data Mining & Knowledge Management Process*, vol. 4, no. 5. pp. 21-34.
- Sneath, P. H. A., Sokal, R. R., 1973. Numerical taxonomy: the principles and practice of ! numerical classification. San Francisco: Freeman, 1973. 573 p. [Medical Research Council Microbial Systematics Unit, Univ. Leicester, England and Dept. Ecology and Evolution, State Univ. New York, Stony Brook, NY].
- Suganya, R., Shanthi, R., 2012. Fuzzy C- Means Algorithm - A Review. *Int. J. of Scientific and Research Publications*, vol. 2, no. 11, pp. 1-3.
- Tang, Y.G., Sun, Z.Q., 2005. Improved validation index for fuzzy clustering, in: American Control Conf., June 8–10, 2005, Portland, OR, USA. pp. 1121-1125.
- Timm, H., Borgelt, C., Kruse, R., 2001. Fuzzy cluster analysis with cluster repulsion. in *Proc. of the European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems*, Tenerife, Spain.
- Tiwari, M., Misra, B., 2011. Application of Cluster Analysis in Agriculture - A Review Article. *Int. J. of Computer Applications*, vol. 36, no.4, pp. 43-47.
- Vanisri D., Loganathan Dr. C., 2011. An Enhanced Fuzzy Possibilistic C-means with Repulsion and Cluster Validity Index. *IJCSNS International Journal of Computer Science and Network Security*, vol.11 no.2.

- Velmurugan, T., 2012. Performance Comparison between K-Means and Fuzzy C-Means Algorithms Using Arbitrary Data Points. *Wulfenia Journal*, vol. 19, no. 8, pp. 234-241.
- Wachs, J., Stern, H., Edan, Y., 2003. Parameter Search for an Image Processing Fuzzy C-Means Hand Gesture Recognition System in *Proc. of IEEE International Conference on Image Processing*, vol. 3, pp. 341-345.
- Wachs, J., Shapira, O., Stern, H., 2004. A method to enhance the ‘‘Possibilistic C-means with Repulsion’ algorithm based on cluster validity index. In *Proc. the 9th World Conf. Soft Computing in Industry Application*, 2004, pp. 77-87.
- Wang, W., Zhanga, Y., 2007. On fuzzy cluster validity indices. *Fuzzy Sets and Systems* vol.158, pp. 2095 – 2117.
- Wu, Y., Wu, B., Sun, J., Fu, H., 2010. Unsupervised possibilistic fuzzy clustering. *J. of Information & Computational Science*, vol. 7, no. 5, pp. 1075-1080.
- Xenaki, S. D., Koutroumbas, K. D., & Rontogiannis, A. A., 2014. Sparse adaptive possibilistic clustering. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. doi:10.1109/icassp.2014.6854165
- Xie, X.L, Beni, G., 1991. A Validity Measure for Fuzzy Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 13 (8) : 841–847.
- Xu, R. (2005). Survey of Clustering Algorithms, *IEEE Transactions On Neural Networks*, vol. 16, no. 3, pp. 645-678.
- Yang, W., Wu, K.L., 2006. Unsupervised possibilistic clustering, *Pattern Recognition*, vol. 39, no. 1, pp. 5-21.
- Yu, J., Li, C-X., 2006. Novel cluster validity index for FCM algorithm. *J. Comp. Sci. & Tech*, vol. 21, no. 1, pp. 137-140.
- Zadeh, L. A., 1965. Fuzzy Sets. *Information and Control*, vol. 8, pp. 338-353.
- Zang, C., Zhou, Y., Martin, T., 2008. A validity index for fuzzy and possibilistic c-means algorithm. in *Proc. of IPMU’08*, pp. 877-882.

- Zhang, C., Zhou, Y., Martin, T., 2008. Similarity Based Fuzzy and Possibilistic C-means Algorithm. *Proceedings of the 11th Joint Conference on Information Sciences (JCIS)*. doi:10.2991/jcis.2008.
- Zhang F., Qian X., 2012. A New Validity Index for Fuzzy Clustering. *Journal of Computational Information Systems* 8: 14 (2012) 5875–5883, *School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China*.
- Zhao, X., Li, Y., & Zhao, Q., 2015. Mahalanobis distance based on fuzzy clustering algorithm for image segmentation. *Digital Signal Processing*, 43, 8-16. doi:10.1016/j.dsp.2015.04.009

ÖZGEÇMİŞ

18/11/1989 yılında Adana'da doğdu. İlk, orta ve lise öğrenimini Adana'nın Ceyhan ilçesinde tamamladı. 2009 yılında başladığı Çukurova Üniversitesi, Fen-Edebiyat Fakültesi, İstatistik Bölümü'nden 2013 yılında mezun oldu ve 2014 yılında Ziraat Fakültesi, Zootečni Bölümü, Biyometri ve Genetik Anabilim dalında yüksek lisansa başladı. Yüksek Lisans eğitimi boyunca 2014/2015 Güz Döneminde Erasmus öğrenci değişimi programı dahilinde University of Debrecen(Macaristan) üniversitesinde eğitim görmüştür. Ayrıca 2015/2016 Güz Döneminde Erasmus staj programı dahilinde University of Helsinki(Finlandiya) eğitim görmüştür.

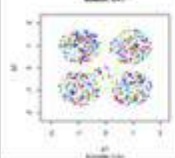
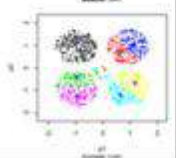
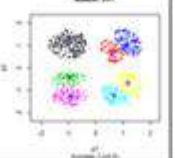
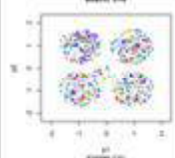
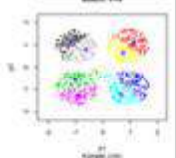
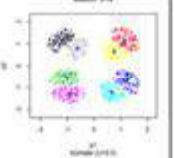
EKLER

EK 1

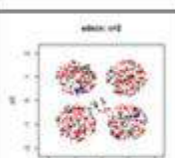
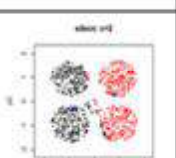
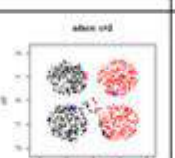
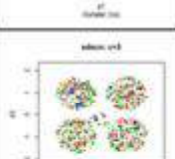
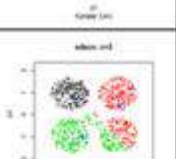
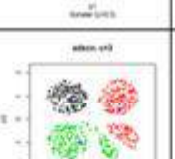
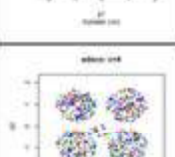
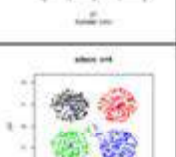
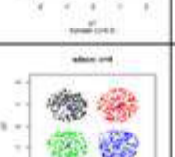
Ek 1.1. BCO küme saylarına göre kümeler ve küme yapıları (SDSC):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_s > 0.5$)	Kümeler (T_{max})	Kümeler ($T_s > 0.5$)
2					
3					
4					
5					
6					

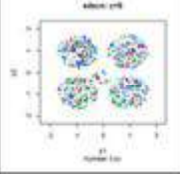
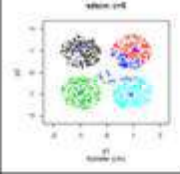
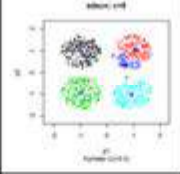
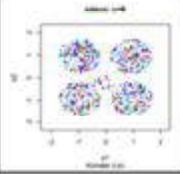
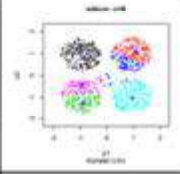
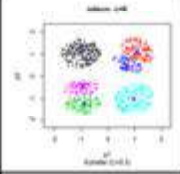
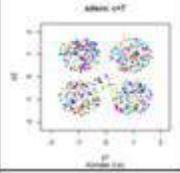
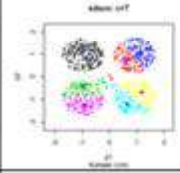
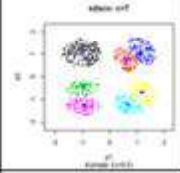
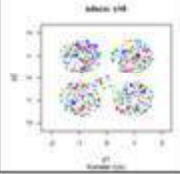
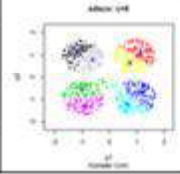
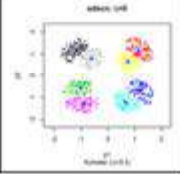
Ek 1.1. (Devamı) BCO küme sayılarına göre kümeler ve küme yapıları (SDSC):

7					
8					

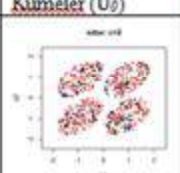
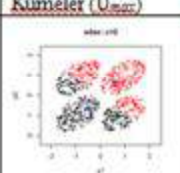
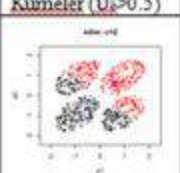
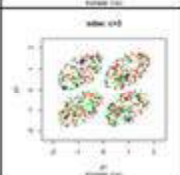
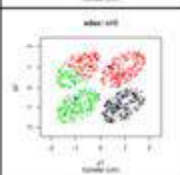
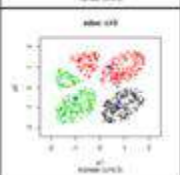
Ek 1.2. BCO küme sayılarına göre kümeler ve küme yapıları (SCSCN)

C	Kümeleler (U_0)	Kümeleler (U_{max})	Kümeleler ($U_s > 0.5$)	Kümeleler (T_{max})	Kümeleler ($T_s > 0.5$)
2					
3					
4					

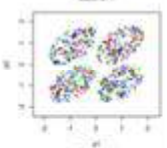
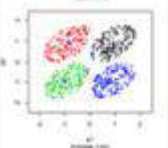
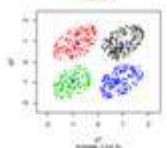
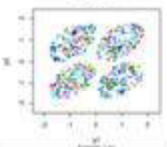
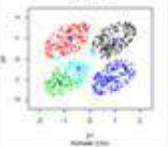
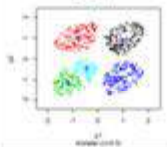
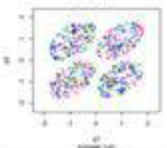
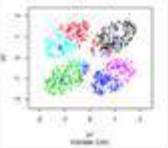
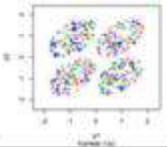
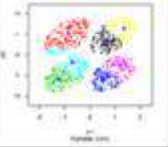
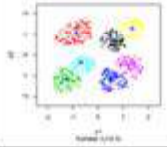
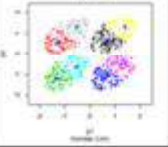
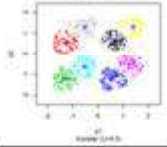
Ek 1.2. (Devamı) BCO küme sayılarına göre kümeler ve küme yapıları (SCSCN):

5					
6					
7					
8					

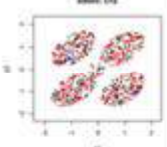
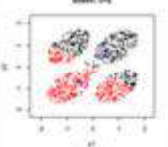
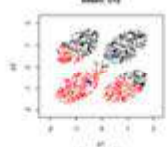
Ek 1.3. BCO küme sayılarına göre kümeler ve küme yapıları (SCSE):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_0 > 0.5$)	Kümeler (T_{max})	Kümeler ($T_0 > 0.5$)
2					
3					

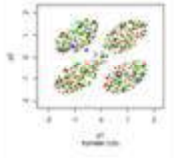
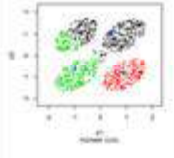
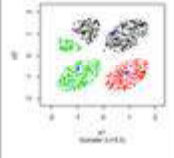
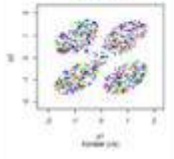
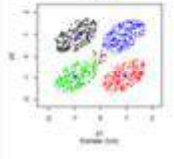
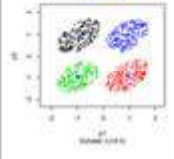
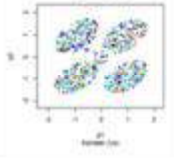
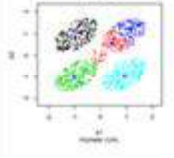
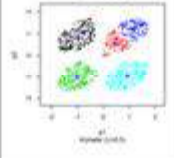
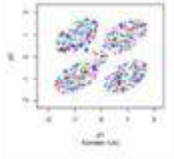
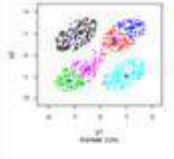
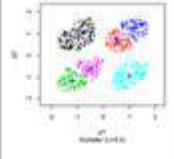
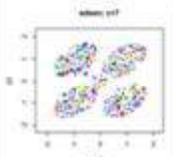
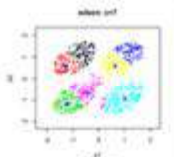
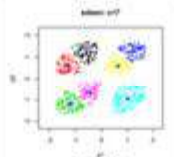
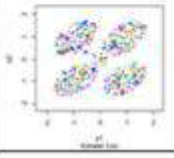
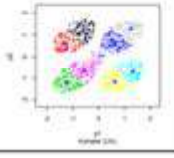
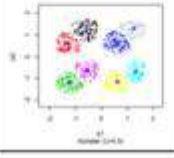
Ek 1.3. (Devamı) BCO küme sayılarına göre kümeler ve küme yapıları (SCSE):

4					
5					
6					
7					
8					

Ek 1.4. BCO küme sayılarına göre kümeler ve küme yapıları (SCSEN):

2					
---	---	---	---	--	--

Ek 1.4. (Devamı) BCO küme sayılarına göre kümeler ve küme yapıları (SCSEN):

3					
4					
5					
6					
7					
8					

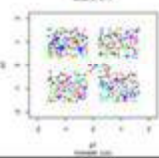
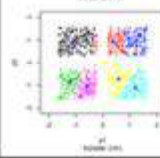
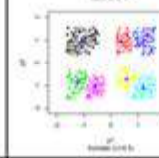
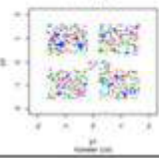
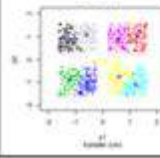
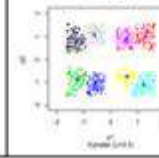
Ek 1.5. BCO küme sayılarına göre kümeler ve küme yapıları (SDSR):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_0 > 0.5$)	Kümeler (T_{max})	Kümeler ($T_0 > 0.5$)
2					
3					
4					
5					
6					
7					
8					

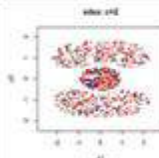
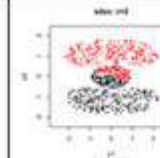
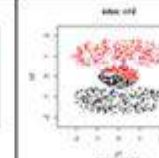
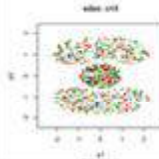
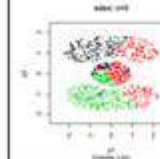
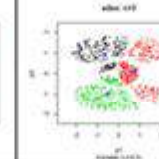
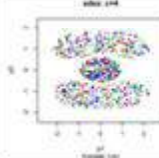
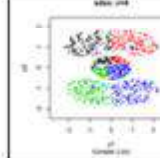
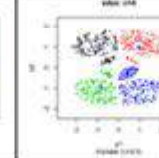
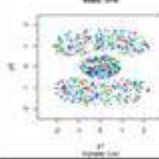
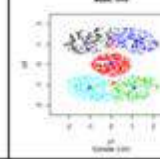
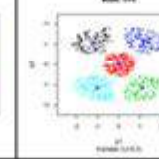
Ek 1.6. BCO küme saylarına göre kümeler ve küme yapıları (SDSRN):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_0 > 0.5$)	Kümeler (T_{max})	Kümeler ($T_0 > 0.5$)
2					
3					
4					
5					
6					

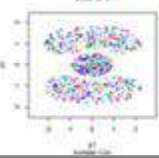
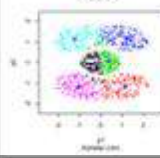
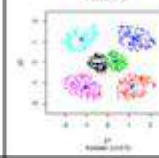
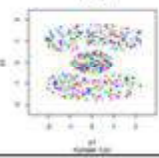
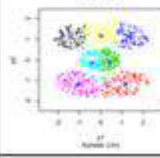
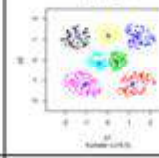
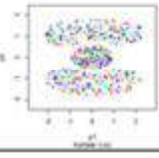
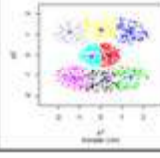
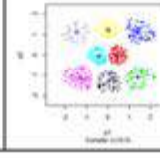
Ek 1.6. (Devamı) BCO küme sayılarına göre kümeler ve küme yapıları (SDSRN):

7					
8					

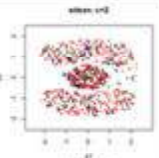
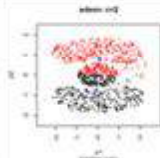
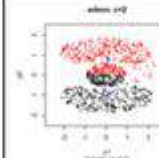
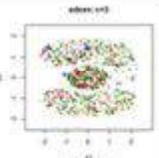
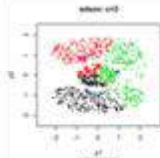
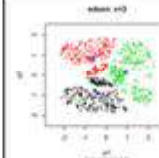
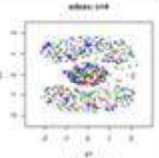
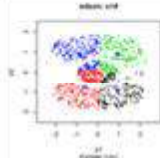
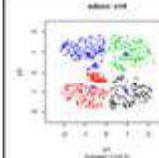
Ek 1.7. BCO küme sayılarına göre kümeler ve küme yapıları (SDSX):

	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_j > 0.5$)	Kümeler (T_{max})	Kümeler ($T_j > 0.5$)
2					
3					
4					
5					

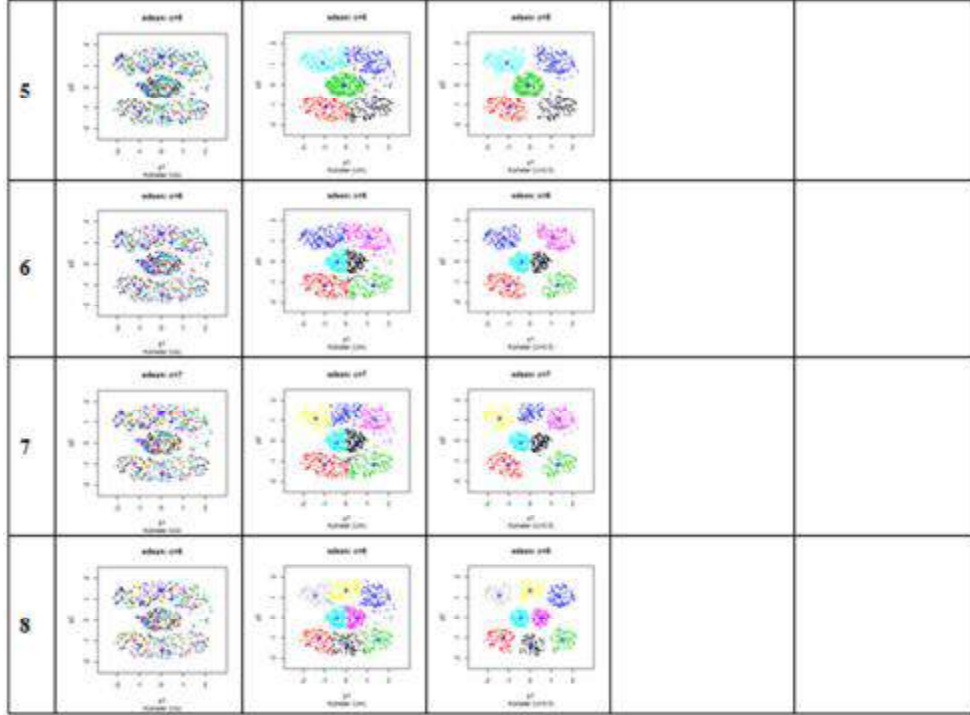
Ek 1.7. (Devamı) BCO küme sayılarına göre kümeler ve küme yapıları (SDSX):

6					
7					
8					

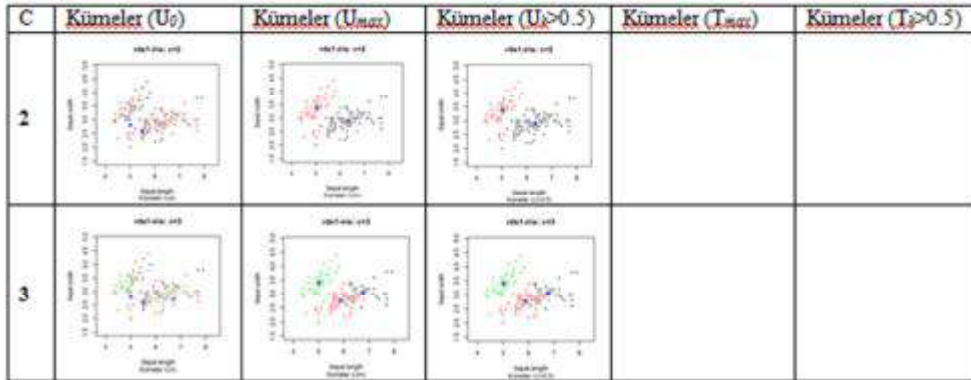
Ek 1.8. BCO küme sayılarına göre kümeler ve küme yapıları (SDSXN):

C	Kümeleşim (U_0)	Kümeleşim (U_{max})	Kümeleşim ($U_j > 0.5$)	Kümeleşim (T_{max})	Kümeleşim ($T_j > 0.5$)
2					
3					
4					

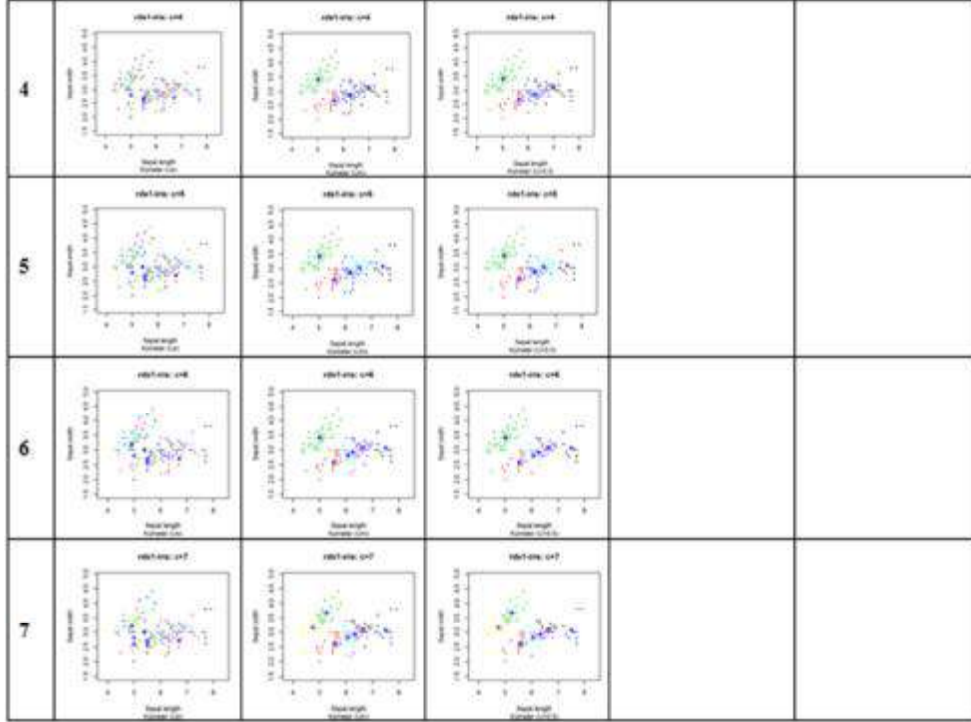
Ek 1.8. (Devamı) BCO küme sayılarına göre kümeler ve küme yapıları (SDSXN):



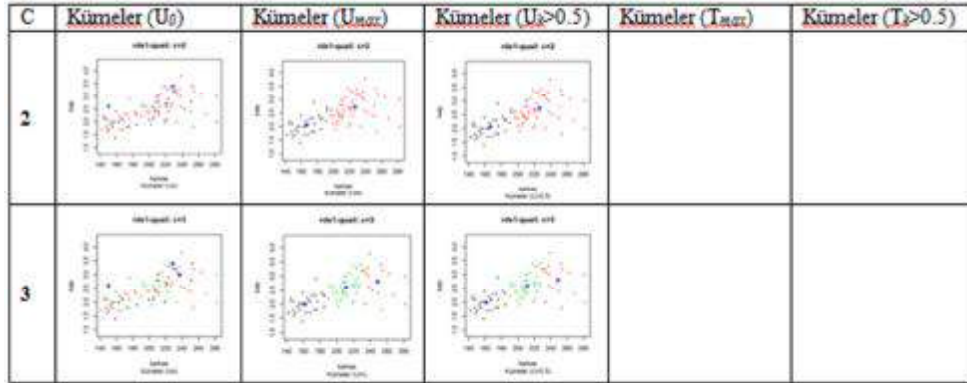
Ek 1.9. BCO küme sayılarına göre kümeler ve küme yapıları (RDS1-Iris):



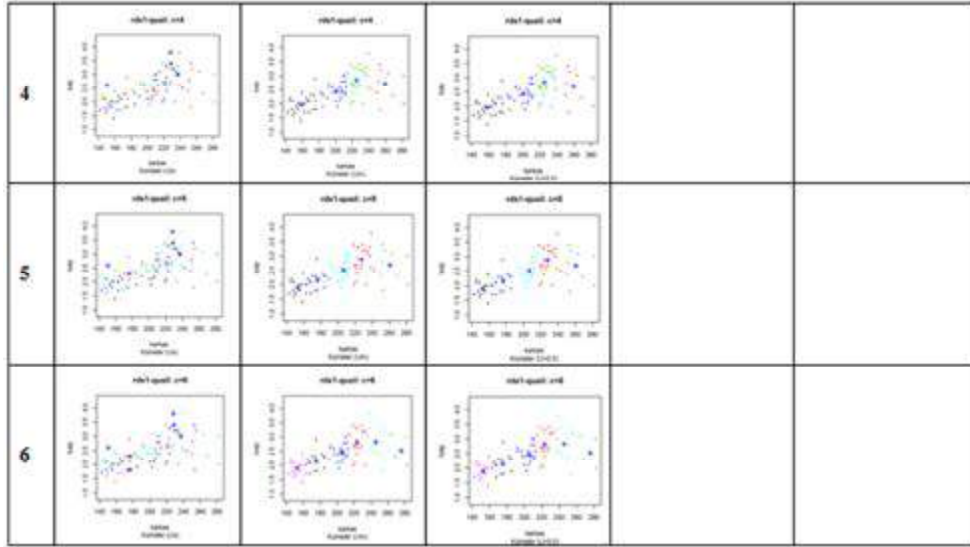
Ek 1.9. (Devamı) BCO küme sayılarına göre kümeler ve küme yapıları (RDS1-Iris):



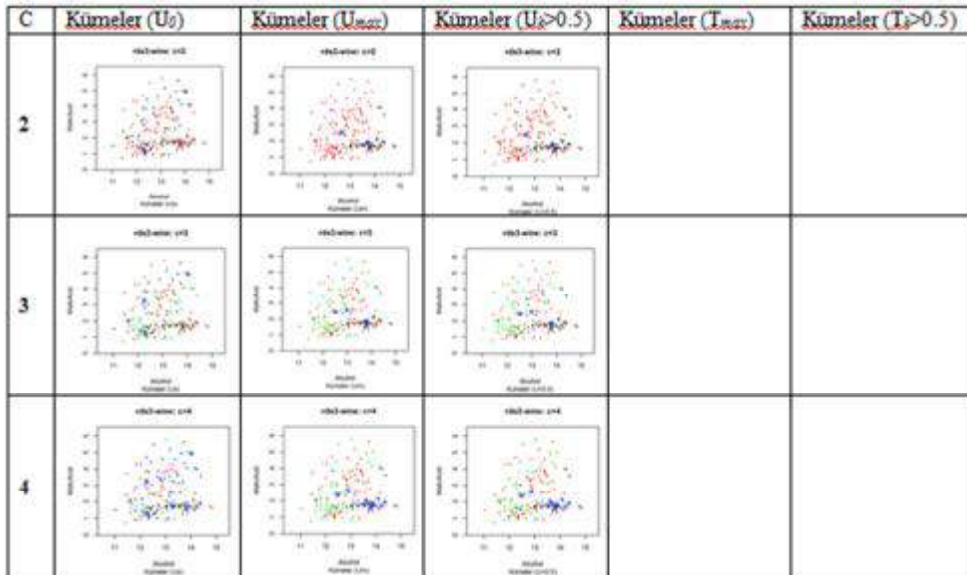
Ek 1.10. BCO küme sayılarına göre kümeler ve küme yapıları (RDS2-Quail):



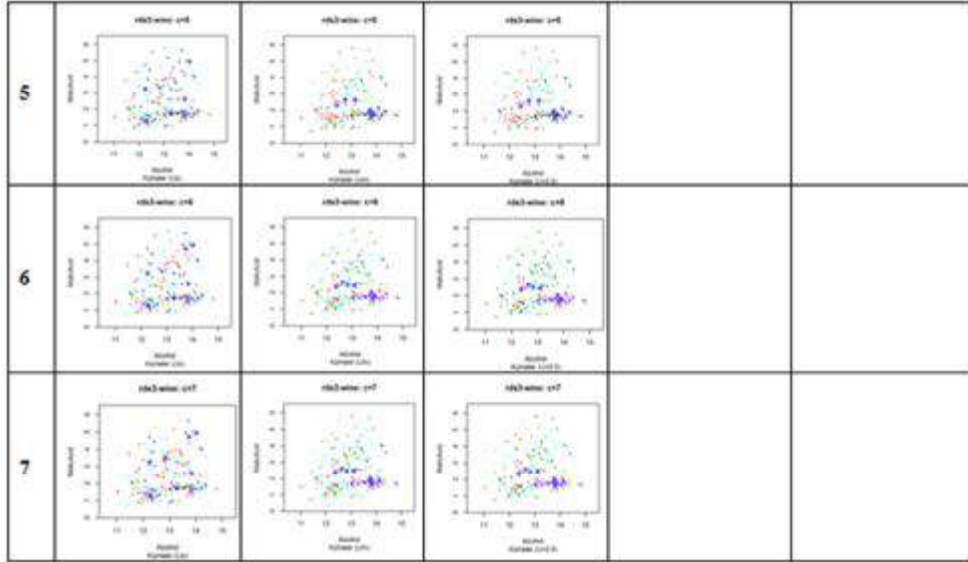
Ek 1.10. (Devamı) BCO küme saylarına göre kümeler ve küme yapıları (RDS2-Quail):



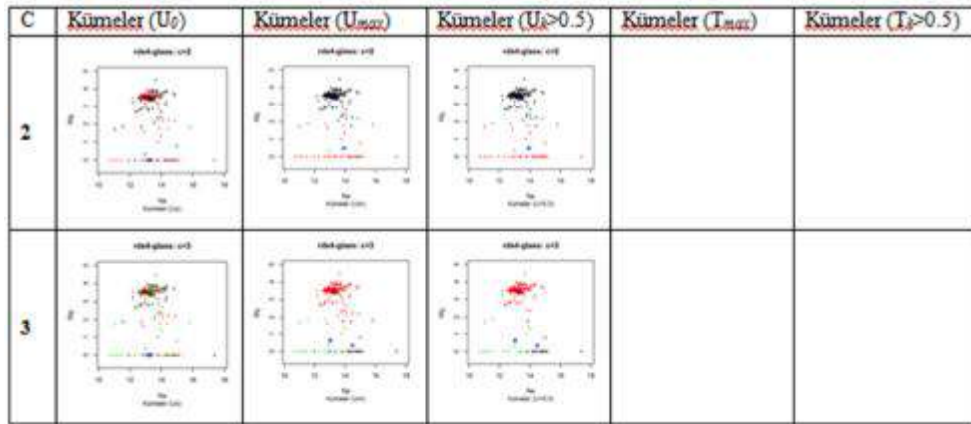
Ek 1.11. BCO küme saylarına göre kümeler ve küme yapıları (RDS3-Wine):



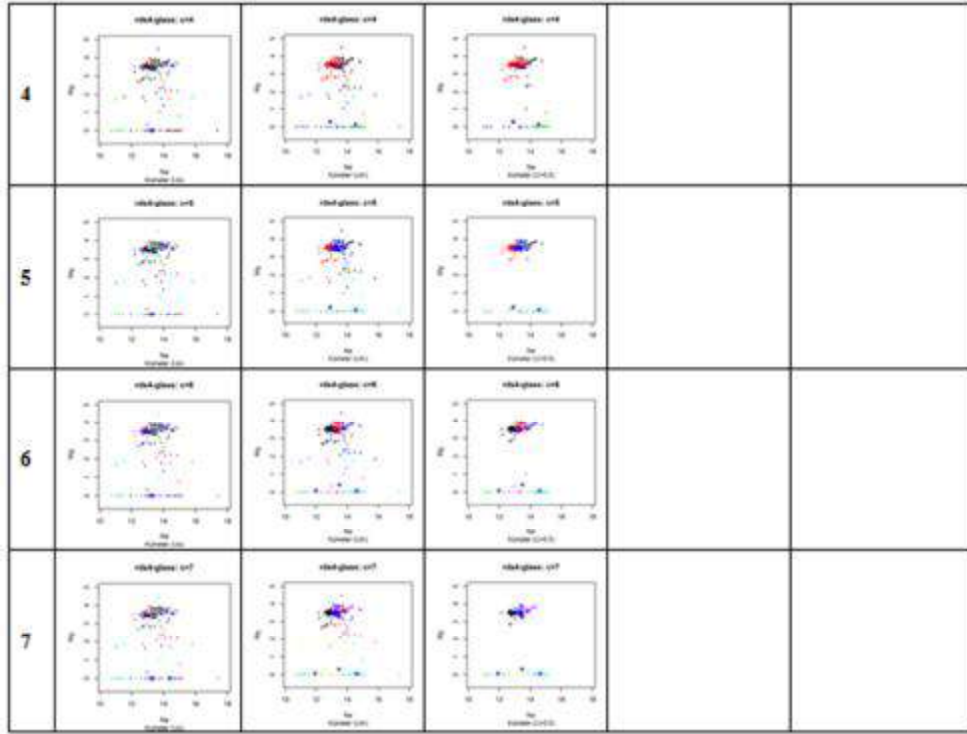
Ek 1.11. (Devamı) BCO küme saylarına göre kümeler ve küme yapıları (RDS3-Wine):



Ek 1.12. BCO küme saylarına göre kümeler ve küme yapıları (RDS4-Glass):

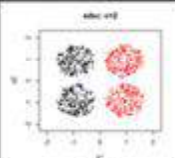
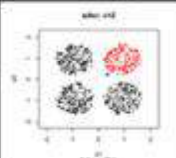
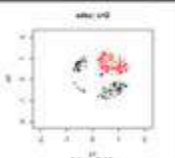
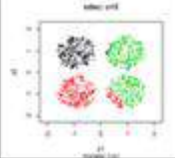
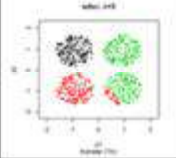
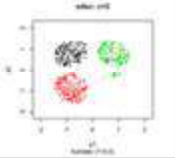


Ek 1.12. (Devamı) BCO küme saylarına göre kümeler ve küme yapıları (RDS4-Glass):

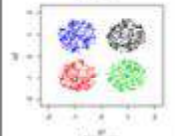
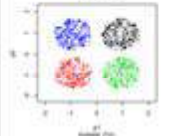
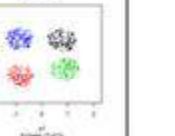
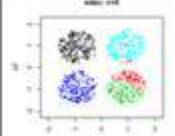
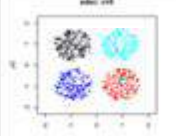
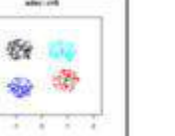
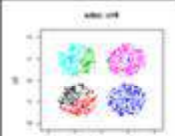
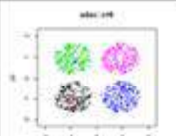
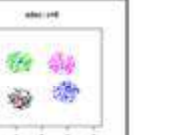
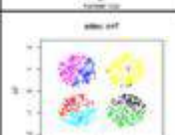
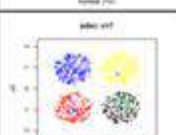
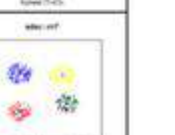
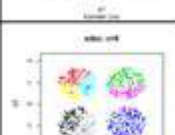
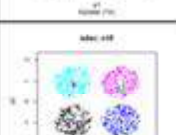
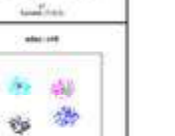


EK 2

Ek 2.1. OCO küme sayılarına göre kümeler ve küme yapıları (SDSC):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_0 > 0.5$)	Kümeler (T_{max})	Kümeler ($T_0 > 0.5$)
2					
3					

Ek 2.1. (Devamı) OCO küme saylarına göre kümeler ve küme yapıları (SDSC):

4					
5					
6					
7					
8					

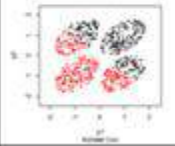
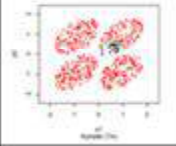
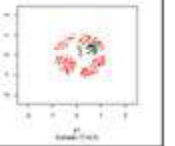
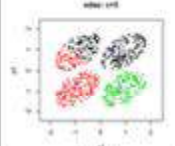
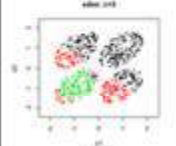
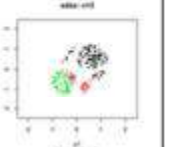
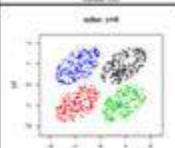
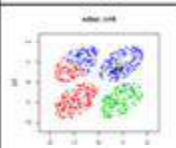
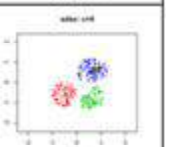
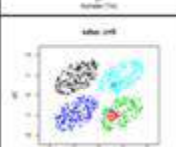
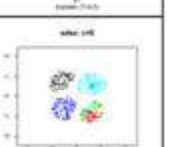
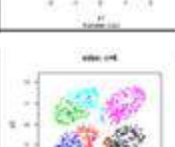
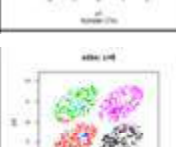
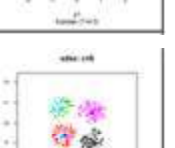
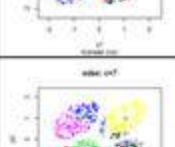
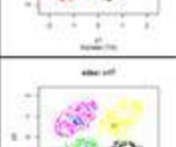
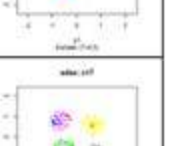
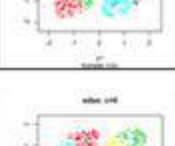
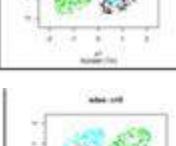
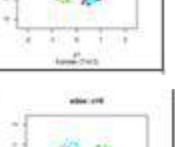
Ek 2.2. OCO küme saylarına göre kümeler ve küme yapıları (SDSCN):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_0 > 0.5$)	Kümeler (T_{max})	Kümeler ($T_0 > 0.5$)
2					

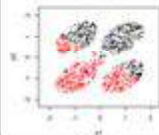
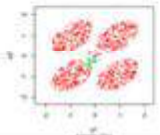
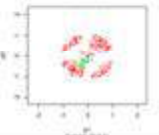
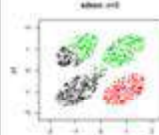
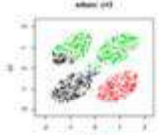
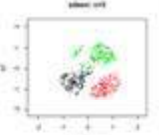
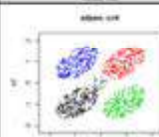
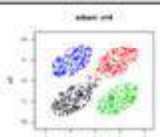
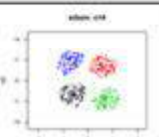
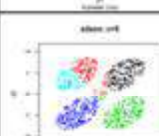
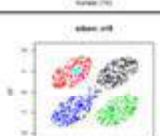
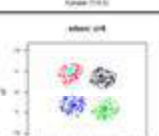
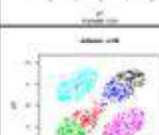
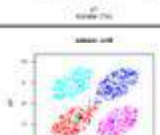
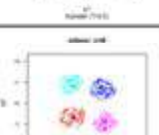
Ek 2.2. (Devamı) OCO küme saylarına göre kümeler ve küme yapıları (SDSCN):



Ek 2.3. OCO küme sayılarına göre kümeler ve küme yapıları (SDSE)

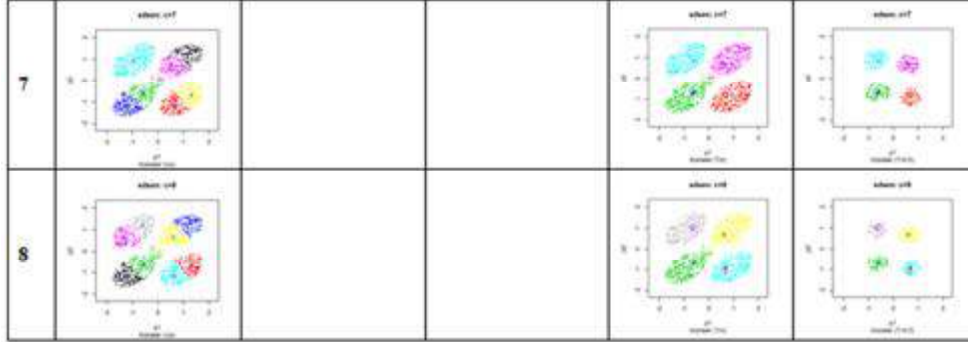
C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_0 > 0.5$)	Kümeler (T_{max})	Kümeler ($T_0 > 0.5$)
2					
3					
4					
5					
6					
7					
8					

Ek 2.4. OCO küme sayılarına göre kümeler ve küme yapıları (SDSEN)

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_k > 0.5$)	Kümeler (T_{max})	Kümeler ($T_k > 0.5$)
2					
3					
4					
5					
6					

1

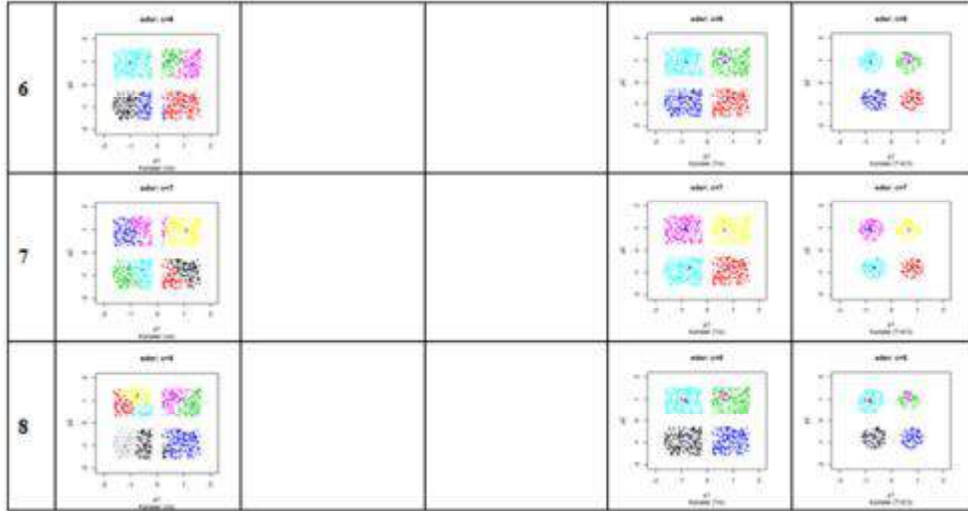
Ek 2.4. (Devamı) OCO küme saylarına göre kümeler ve küme yapıları (SDSEN)



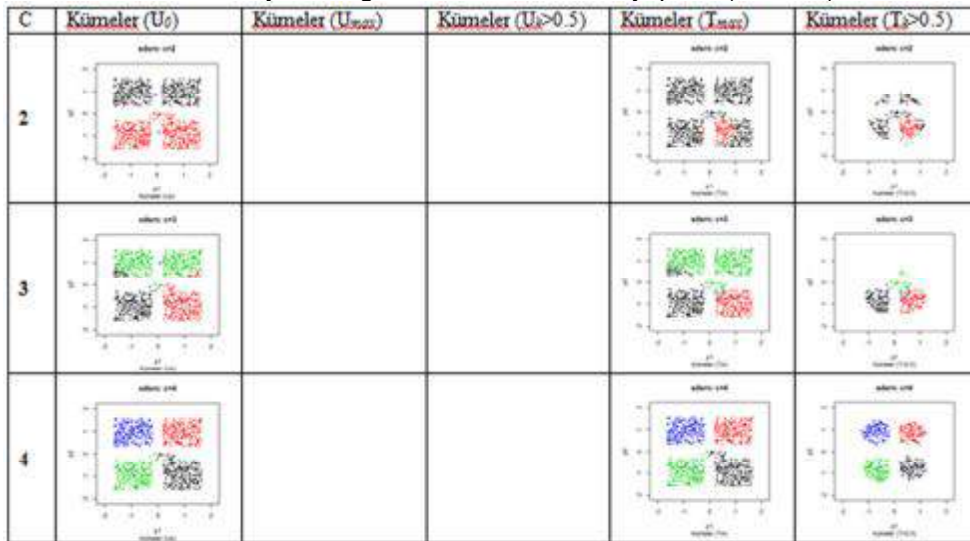
Ek 2.5. OCO küme saylarına göre kümeler ve küme yapıları (SDSR):



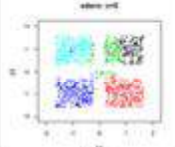
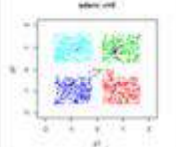
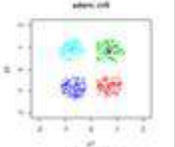
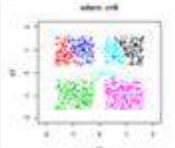
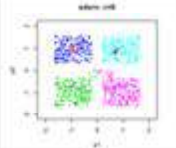
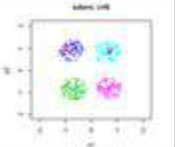
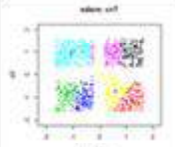
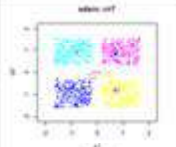
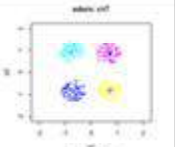
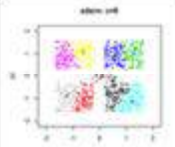
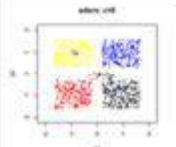
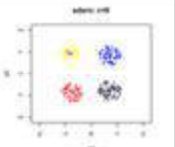
Ek 2.5. (Devamı) OCO küme sayılarına göre kümeler ve küme yapıları (SDSR)



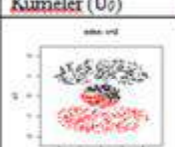
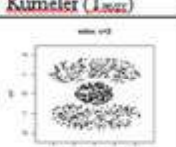
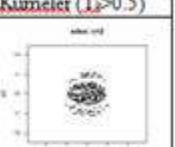
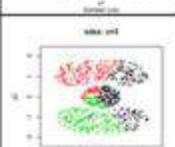
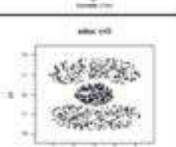
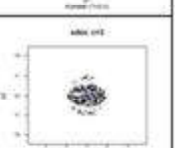
Ek 2.6. OCO küme sayılarına göre kümeler ve küme yapıları (SDSRN):



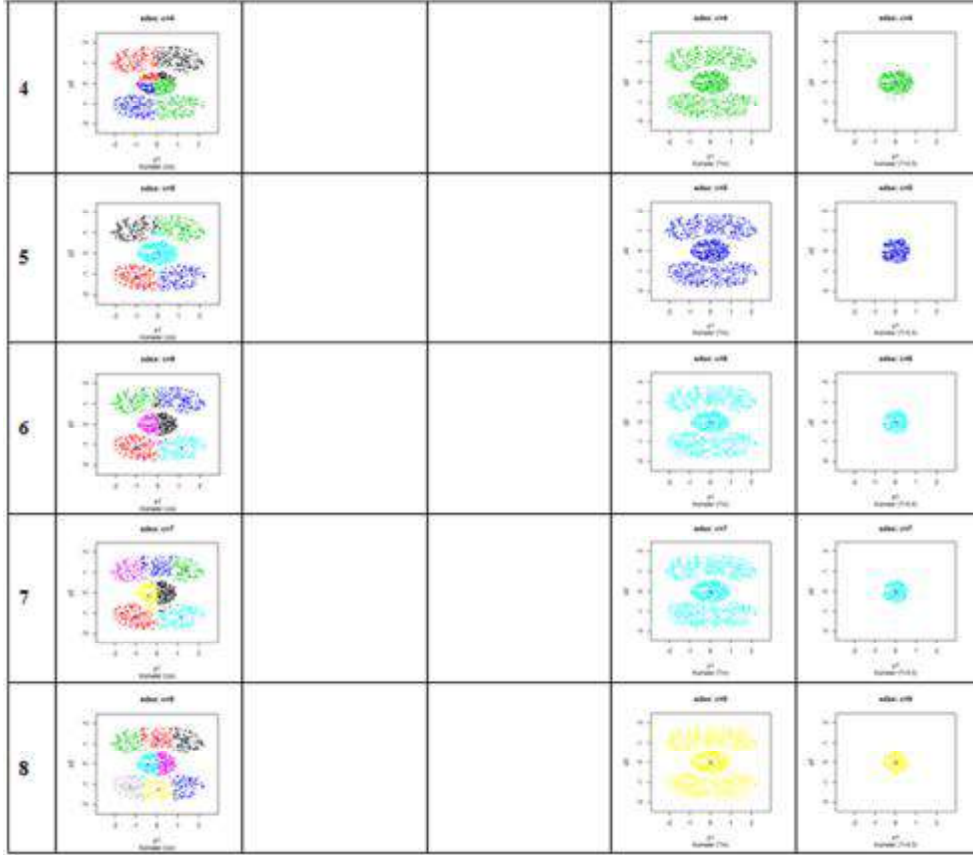
Ek 2.6. (Devamı) OCO küme saylarına göre kümeler ve küme yapıları (SDSRN):

5					
6					
7					
8					

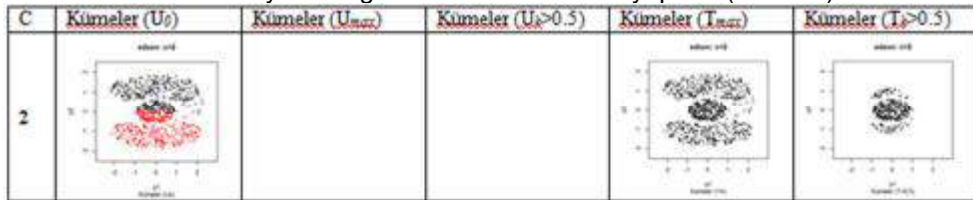
Ek 2.7. OCO küme saylarına göre kümeler ve küme yapıları (SDSX):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_i > 0.5$)	Kümeler (T_{max})	Kümeler ($T_i > 0.5$)
2					
3					

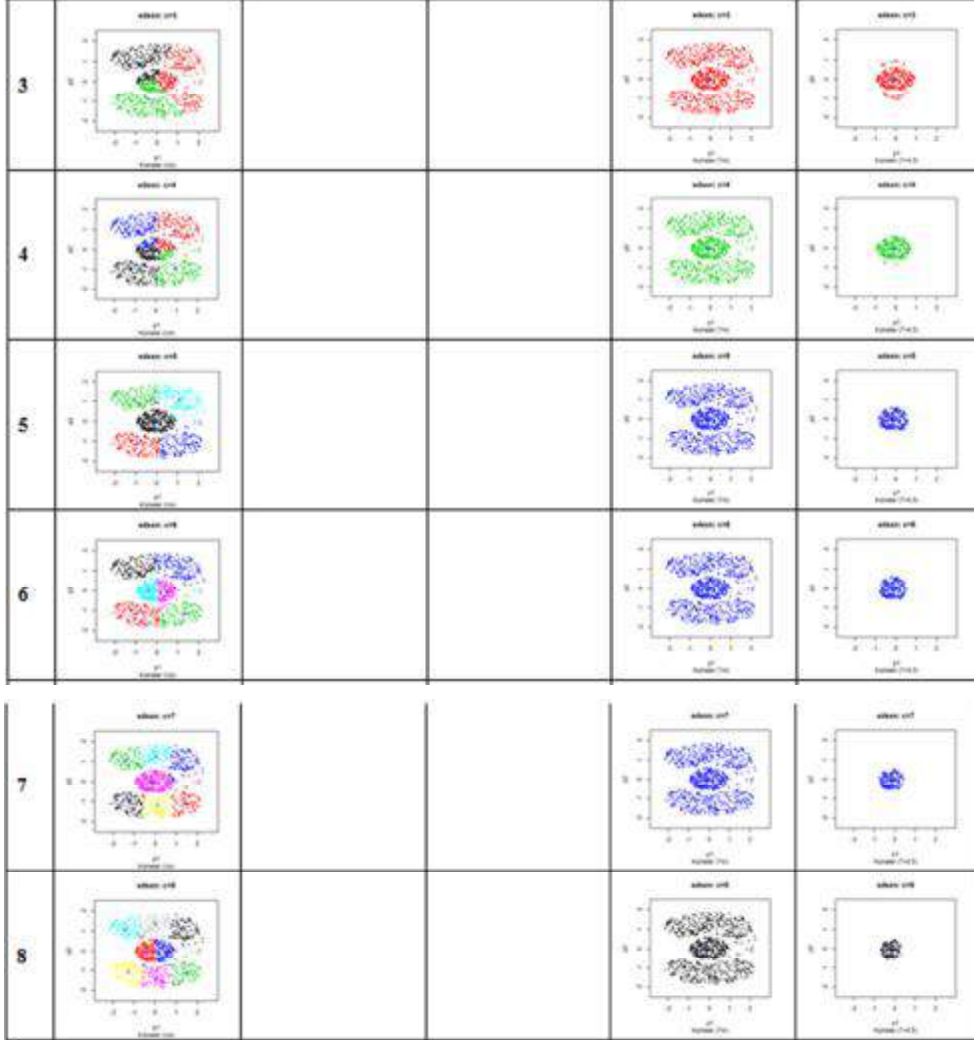
Ek 2.7. (Devamı) OCO küme sayılarına göre kümeler ve küme yapıları (SDSX):



Ek 2.8. OCO küme sayılarına göre kümeler ve küme yapıları (SDSXN):



Ek 2.8. (Devamı) OCO küme saylarına göre kümeler ve küme yapıları (SDSXXN)



Ek 2.9. OCO küme sayılarına göre kümeler ve küme yapıları (RDS1-Iris):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_{i>0.5}$)	Kümeler (T_{max})	Kümeler ($T_{i>0.5}$)
2					
3					
4					
5					
6					
7					

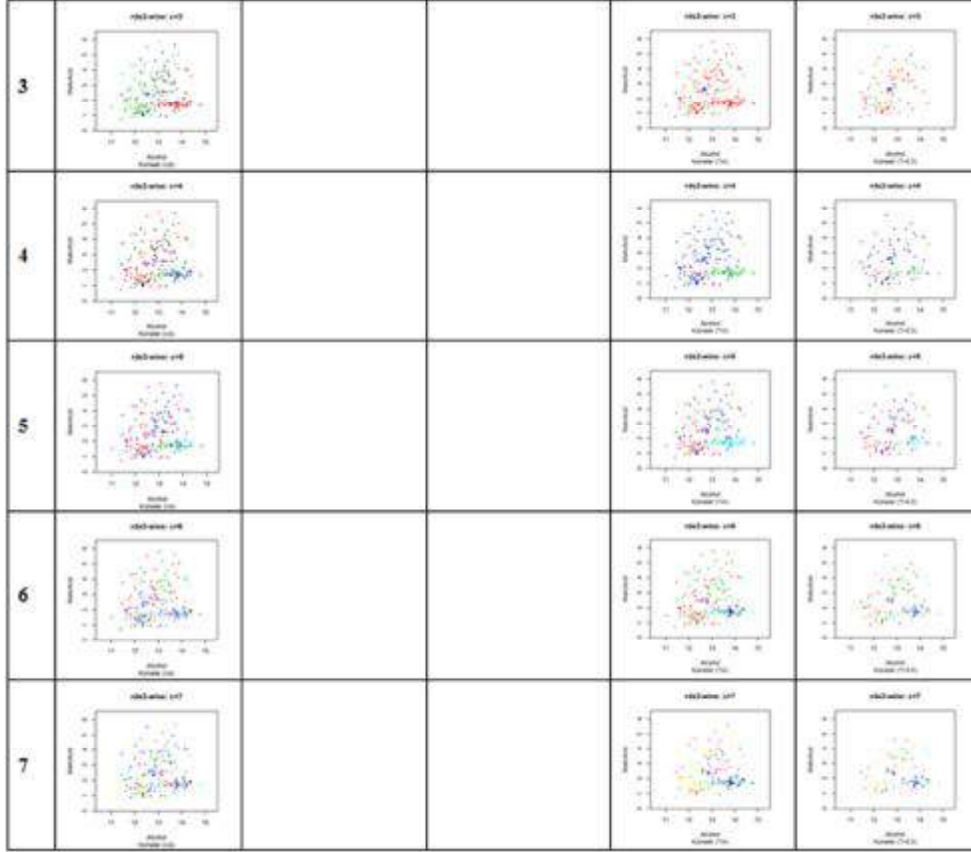
Ek 2.10. OCO küme sayılarına göre kümeler ve küme yapıları (RDS2-Quail):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_k > 0.5$)	Kümeler (T_{max})	Kümeler ($T_k > 0.5$)
2					
3					
4					
5					
6					

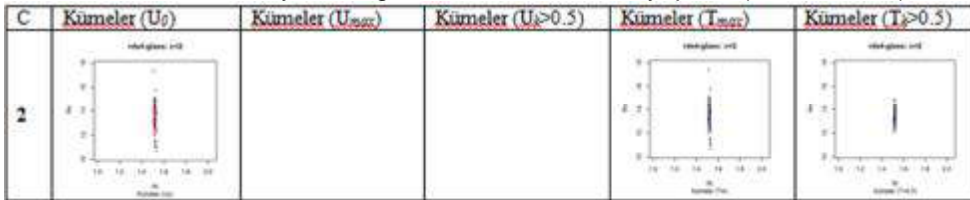
Ek 2.11. OCO küme sayılarına göre kümeler ve küme yapıları (RDS3-Wine):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_k > 0.5$)	Kümeler (T_{max})	Kümeler ($T_k > 0.5$)
2					

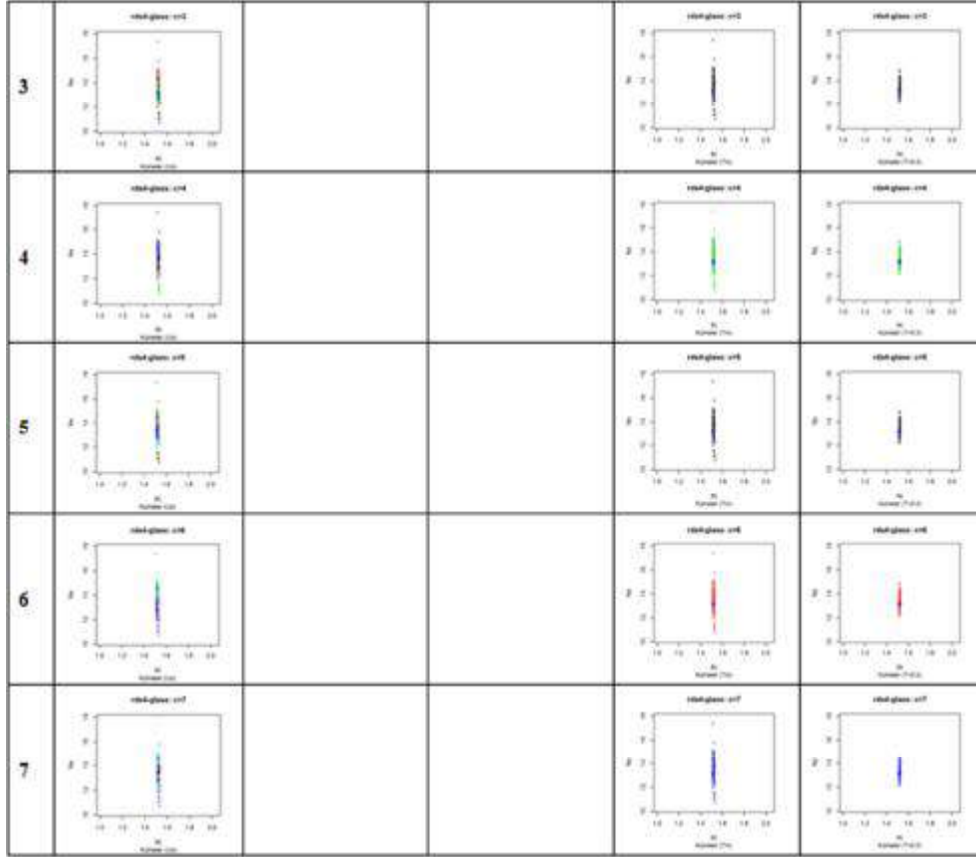
Ek 2.11. (Devamı) OCO küme sayılarına göre kümeler ve küme yapıları (RDS3-Wine):



Ek 2.12. OCO küme sayılarına göre kümeler ve küme yapıları (RDS4-Glass):

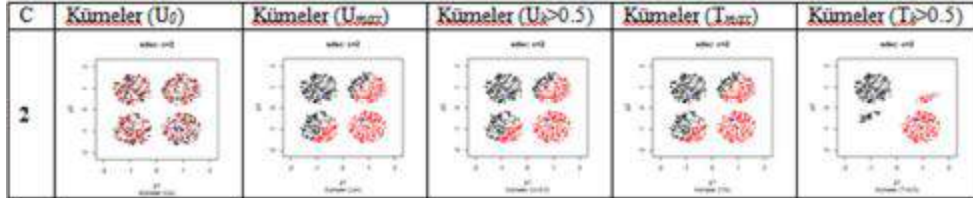


Ek 2.12. (Devamı) OCO küme saylarına göre kümeler ve küme yapıları (RDS4-Glass):

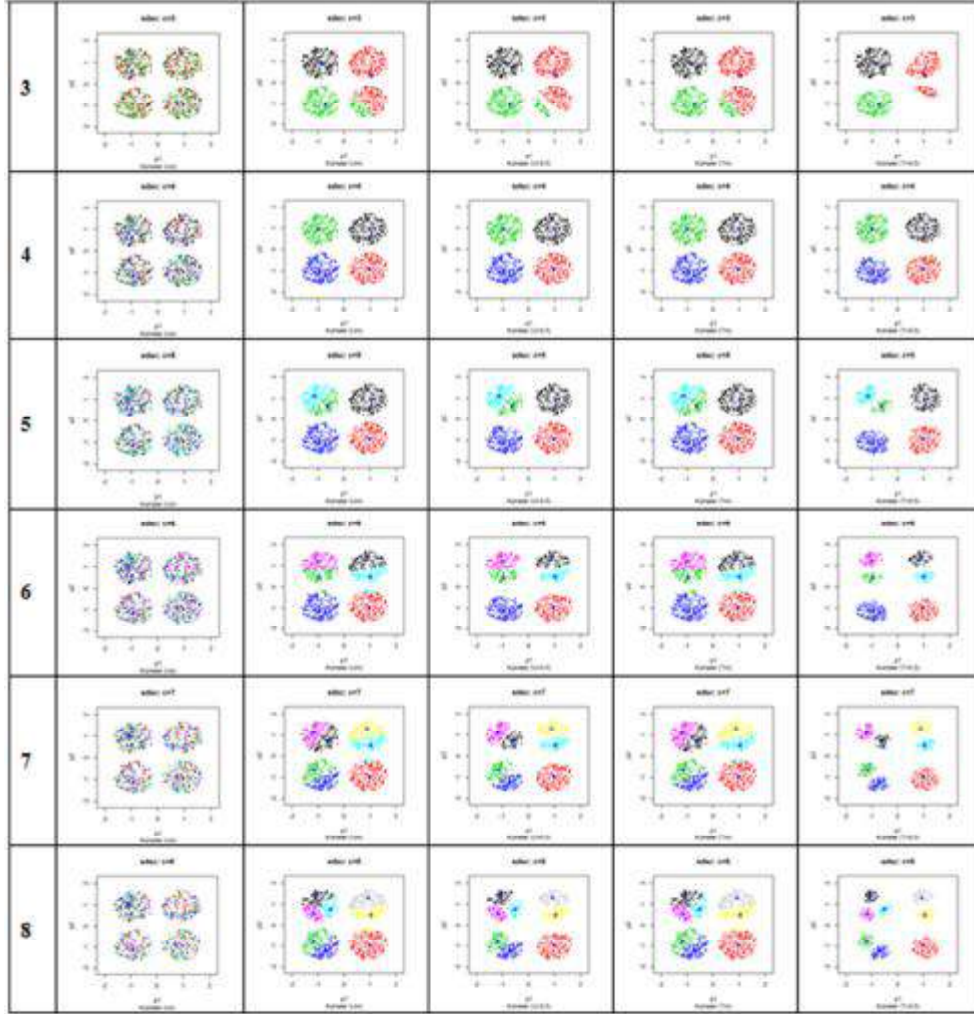


EK 3

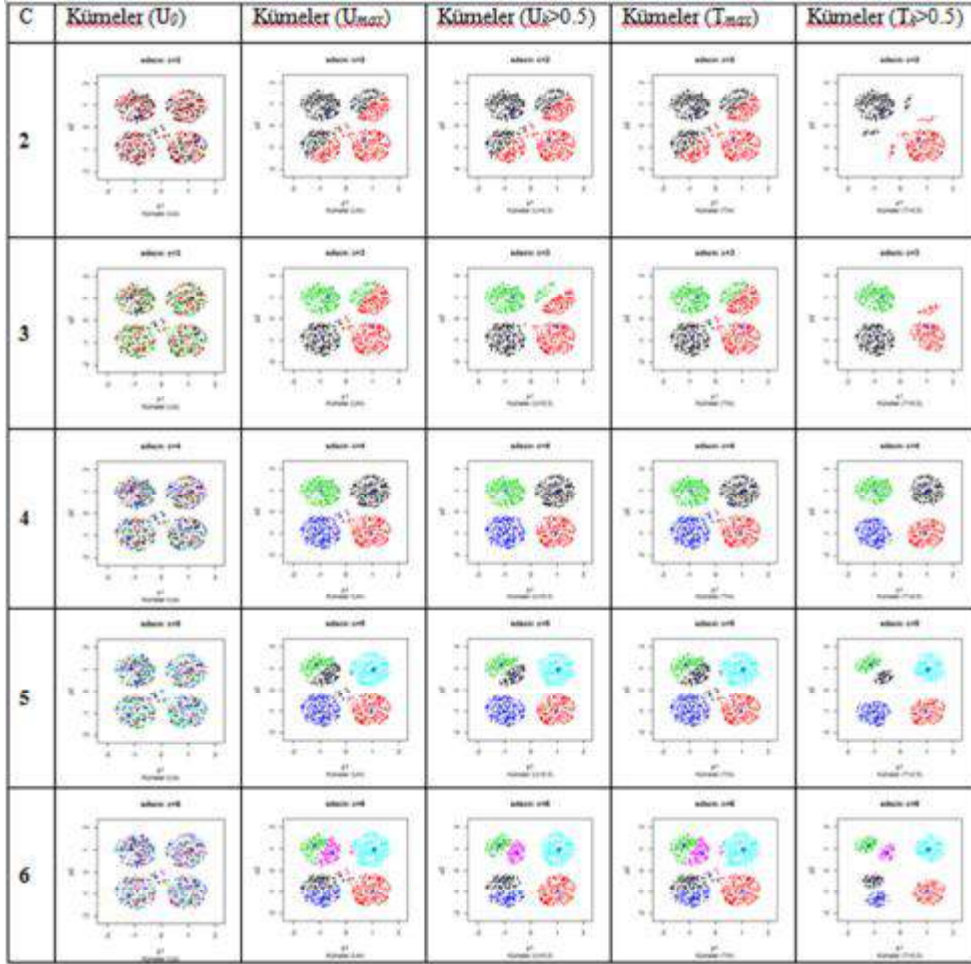
Ek 3.1. BOCO küme sayılarına göre kümeler ve küme yapıları (SDSC):



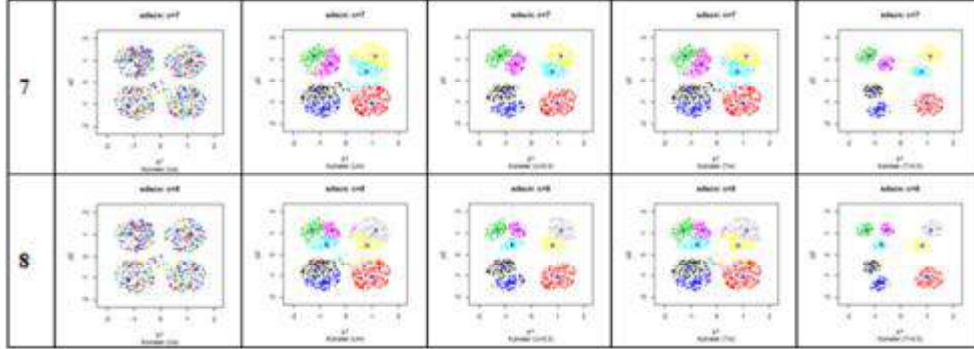
Ek 3.1. (Devamı) BOCO küme sayılarına göre kümeler ve küme yapıları (SDSC):



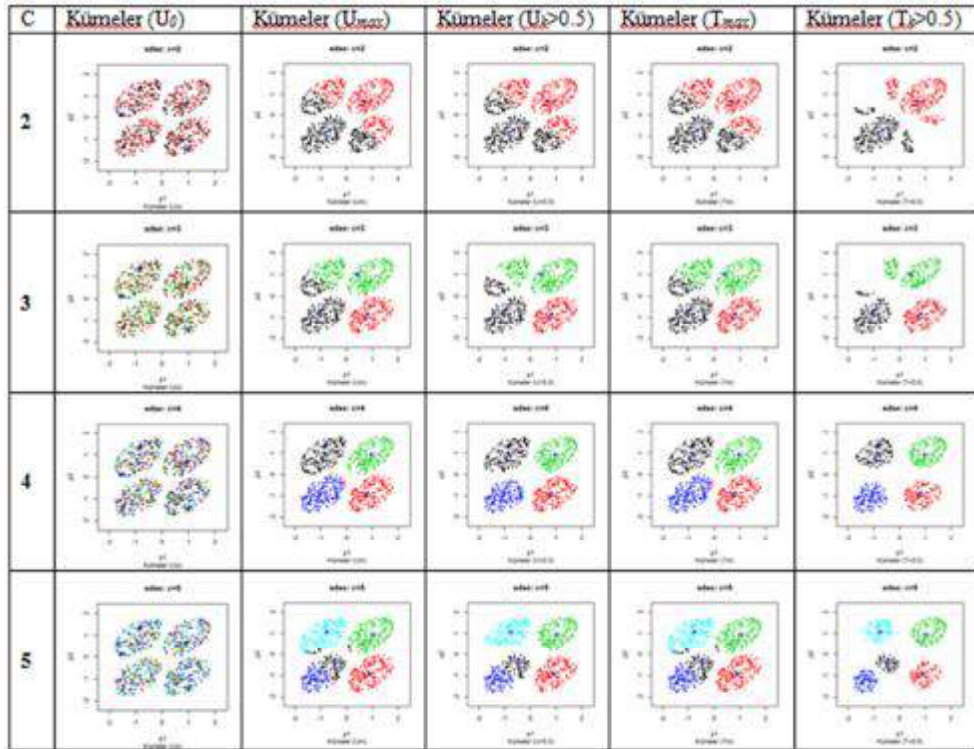
Ek 3.2. BOCO küme sayılarına göre kümeler ve küme yapıları (SDSCN):



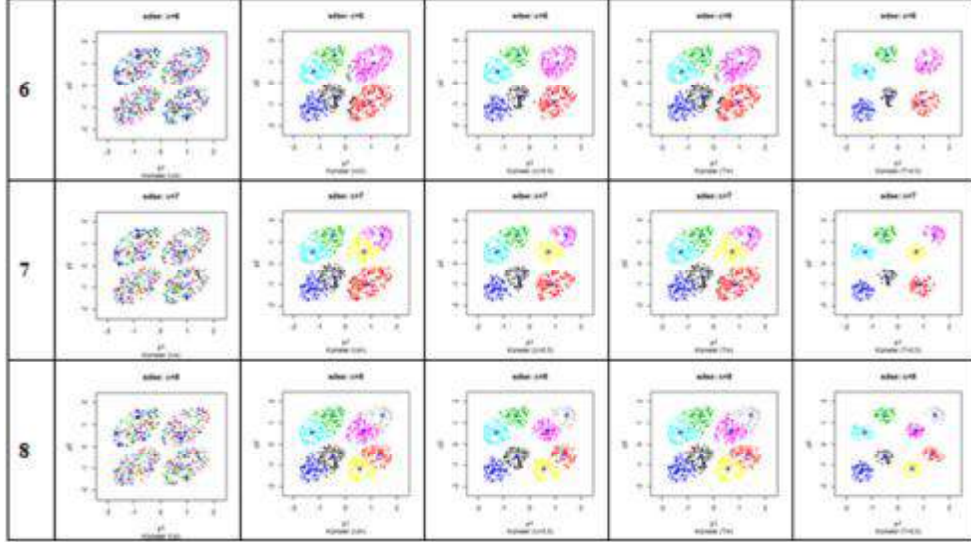
Ek 3.2. (Devamı) BOCO küme saylarına göre kümeler ve küme yapıları (SDSCN):



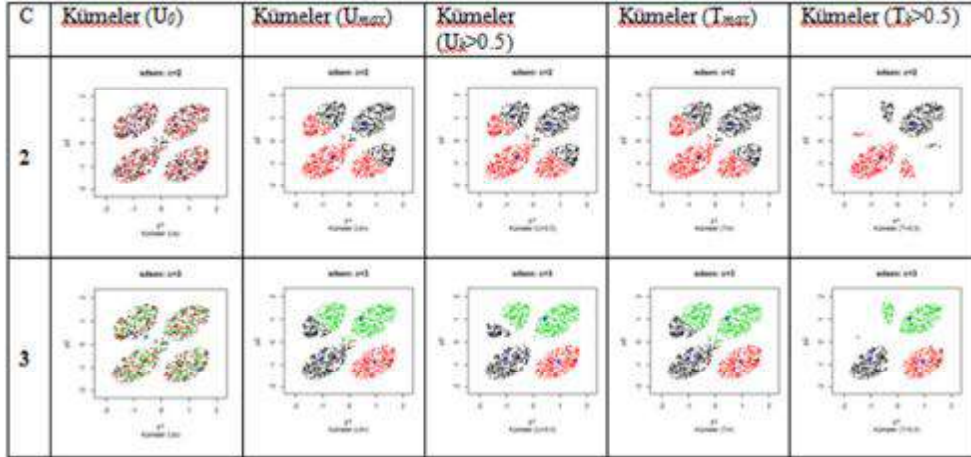
Ek 3.3. BOCO küme saylarına göre kümeler ve küme yapıları (SDSE):



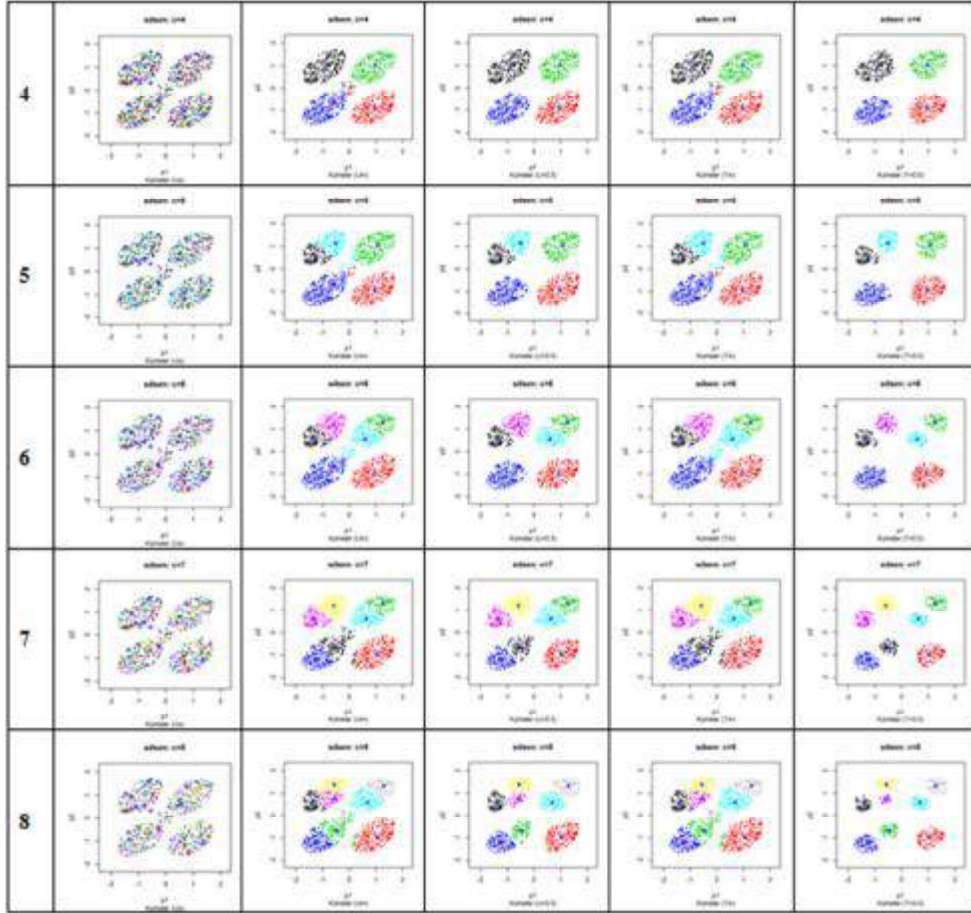
Ek 3.3. (Devamı) BOCO küme sayılarına göre kümeler ve küme yapıları (SDSE):



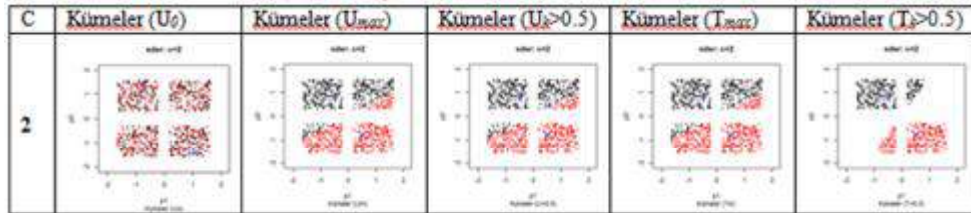
Ek 3.4. BOCO küme sayılarına göre kümeler ve küme yapıları (SDSEN):



Ek 3.4. (Devamı) BOCO küme sayılarına göre kümeler ve küme yapıları (SDSEN):



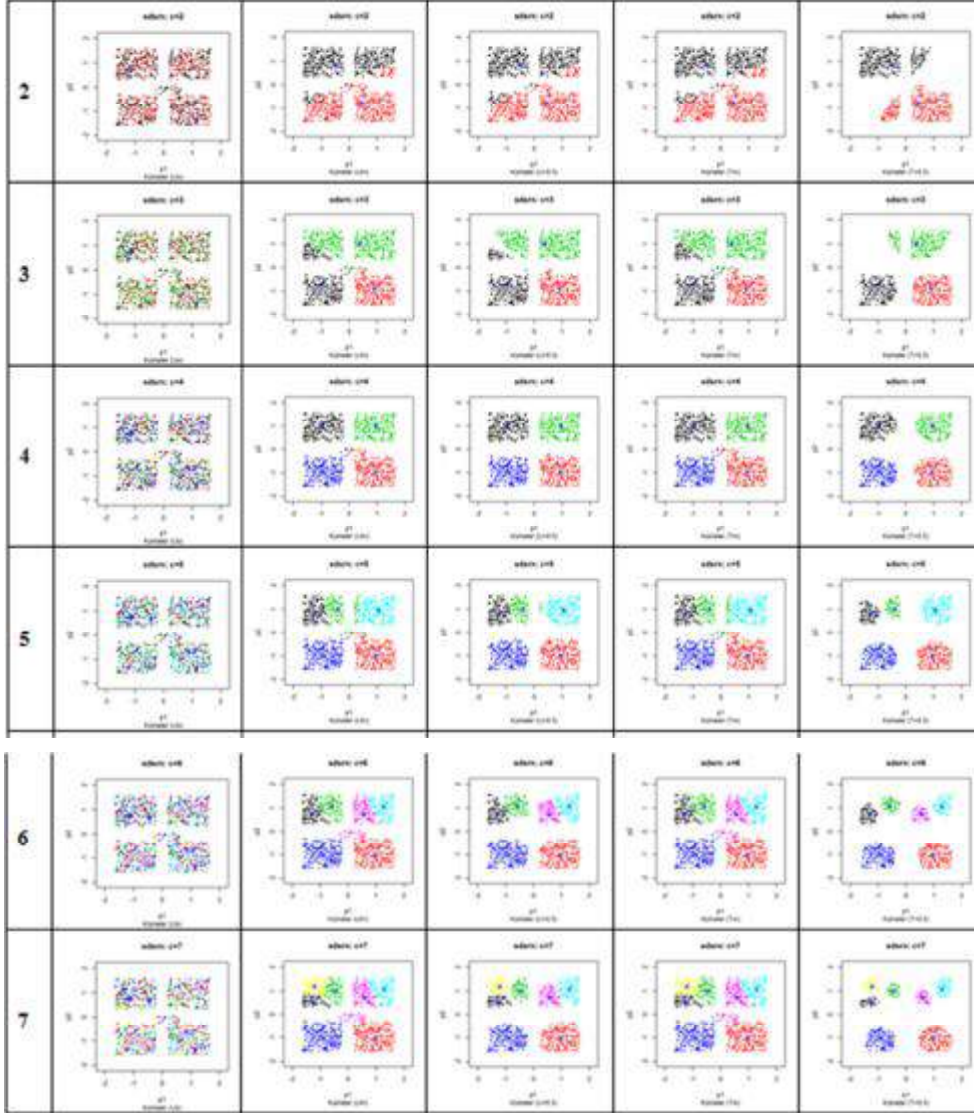
Ek 3.5. BOCO küme sayılarına göre kümeler ve küme yapıları (SDSR):



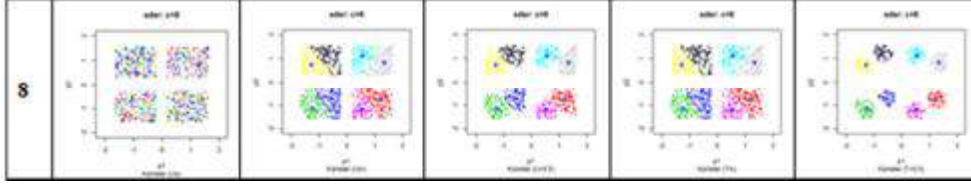
Ek 3.5. (Devamı) BOCO küme saylarına göre kümeler ve küme yapıları (SDSR):



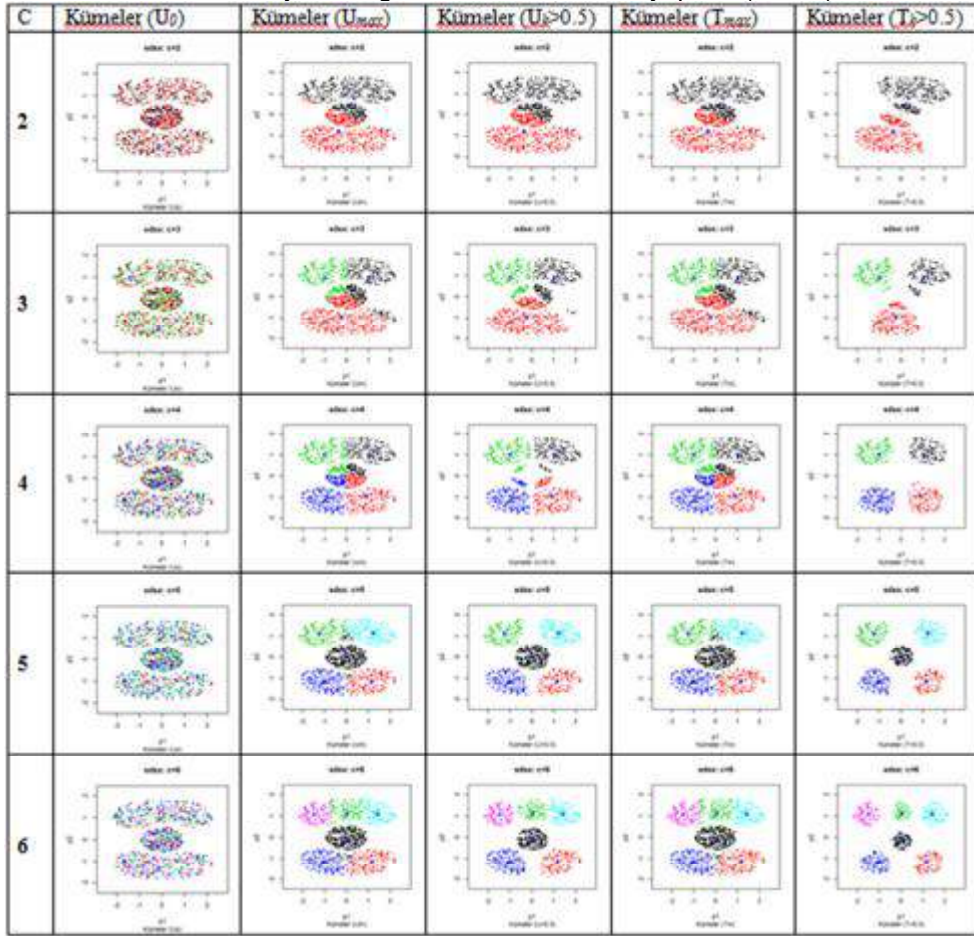
Ek 3.6. BOCO küme sayılarına göre kümeler ve küme yapıları (SDSRN):



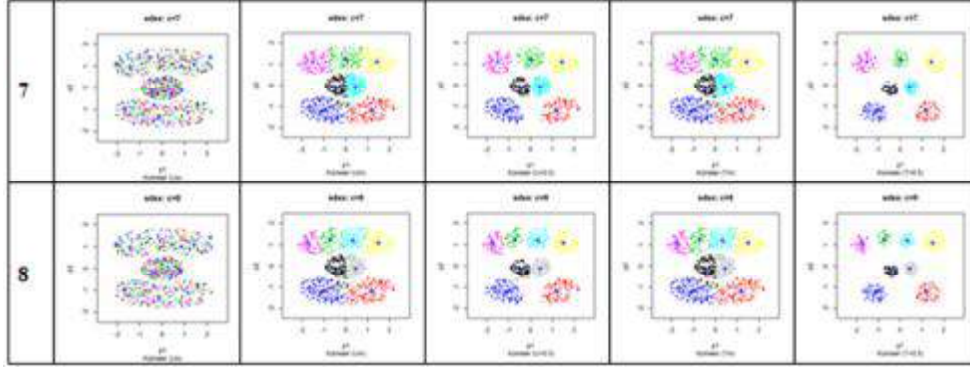
Ek 3.6. (Devamı) BOCO küme saylarına göre kümeler ve küme yapıları (SDSRN):



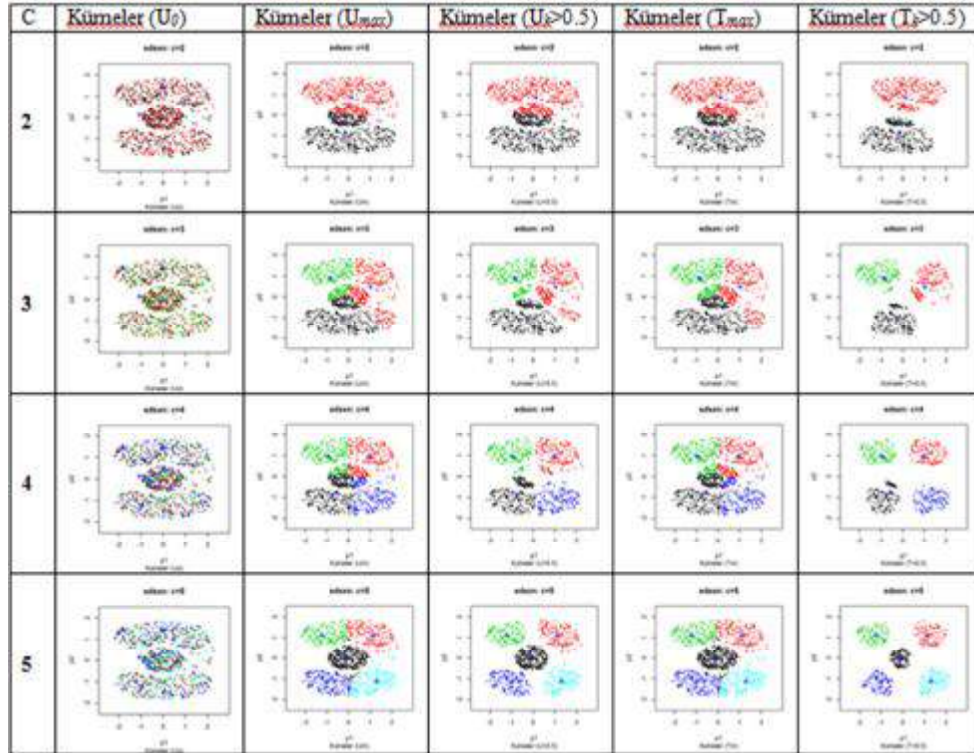
Ek 3.7. BOCO küme saylarına göre kümeler ve küme yapıları (SDSX):



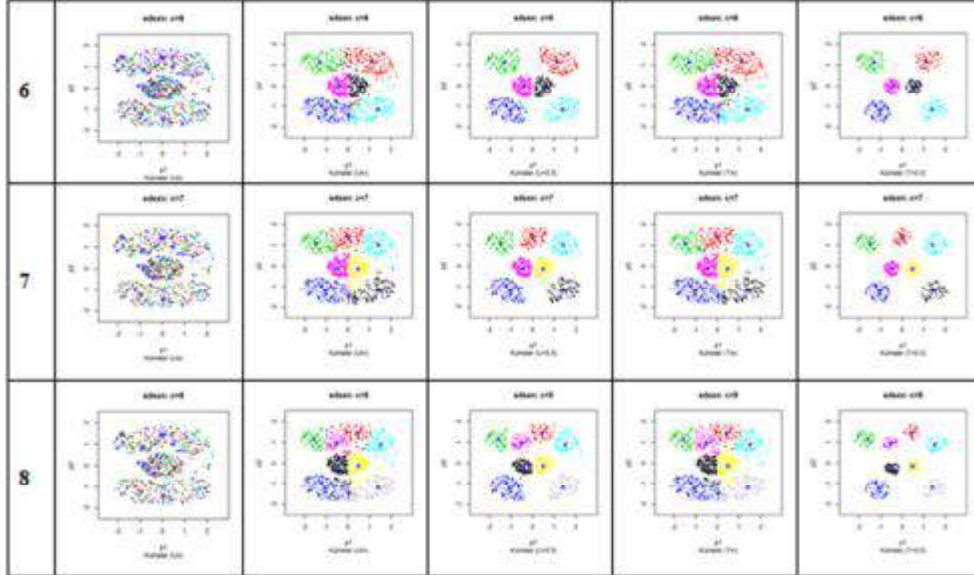
Ek 3.7. (Devamı) BOCO küme saylarına göre kümeler ve küme yapıları (SDSX):



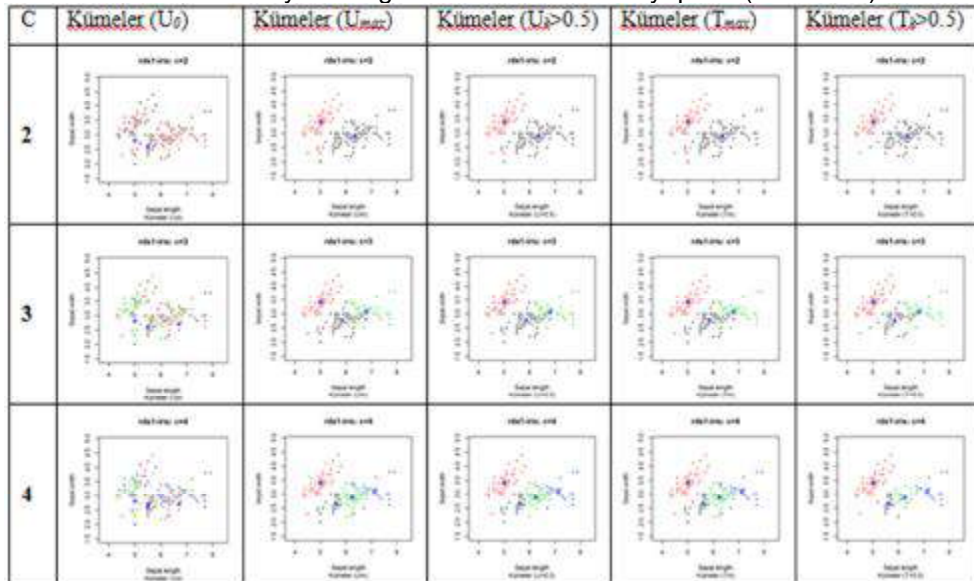
Ek 3.8. BOCO küme saylarına göre kümeler ve küme yapıları (SDSXN)



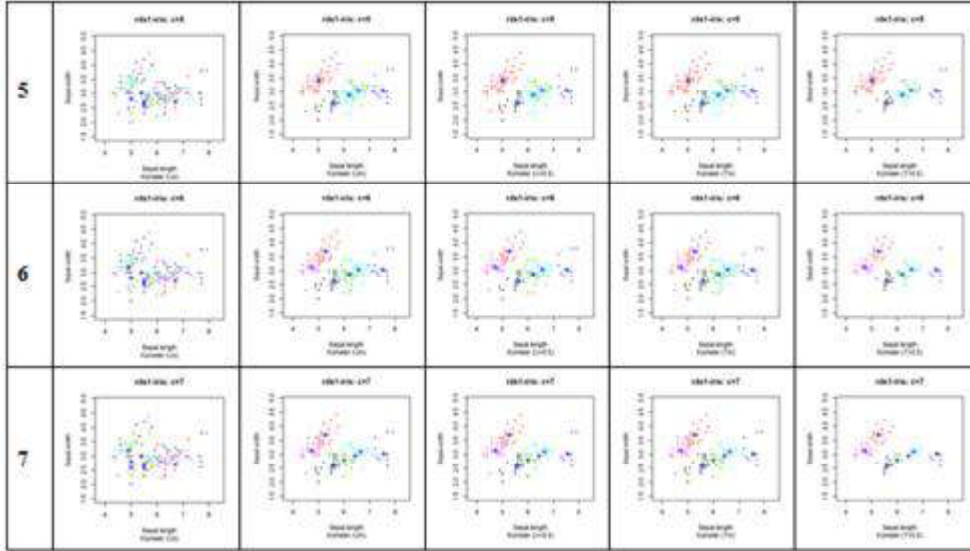
Ek 3.8. (Devamı) BOCO küme sayılarına göre kümeler ve küme yapıları (SDSXN)



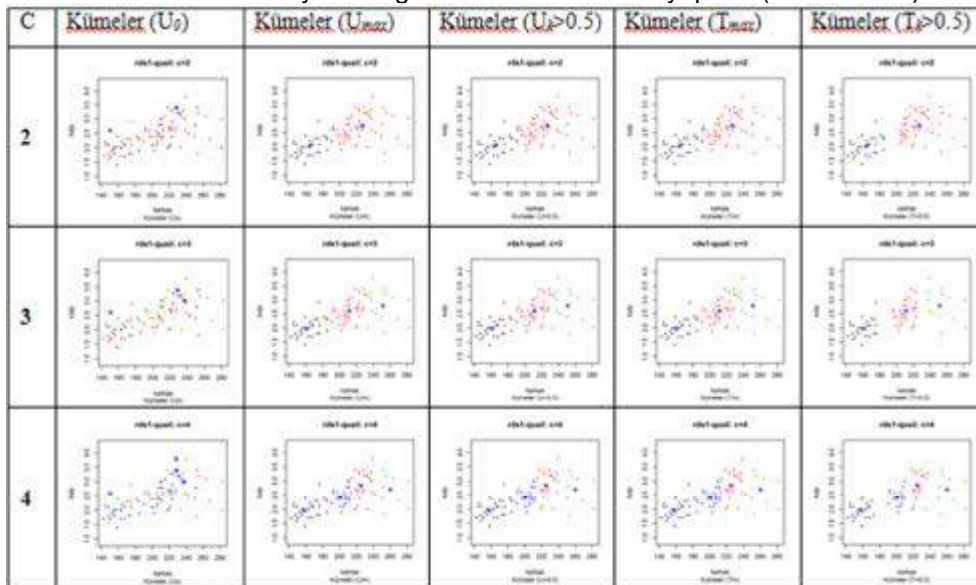
Ek 3.9. BOCO küme sayılarına göre kümeler ve küme yapıları (RDS1-Iris)



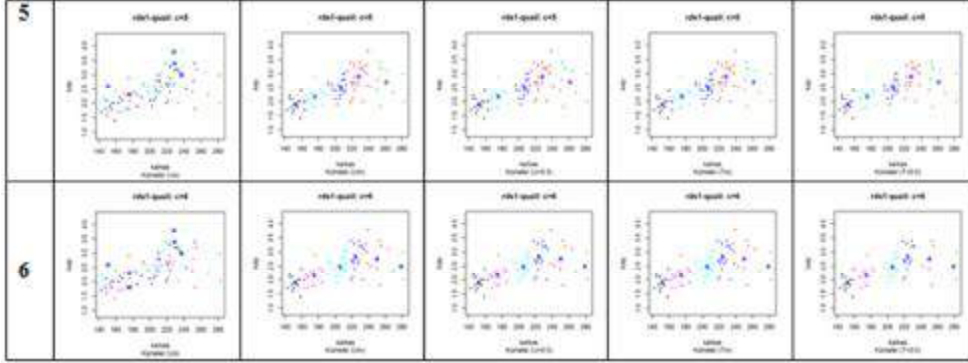
Ek 3.9. (Devamı) BOCO küme sayılarına göre kümeler ve küme yapıları (RDS1-Iris)



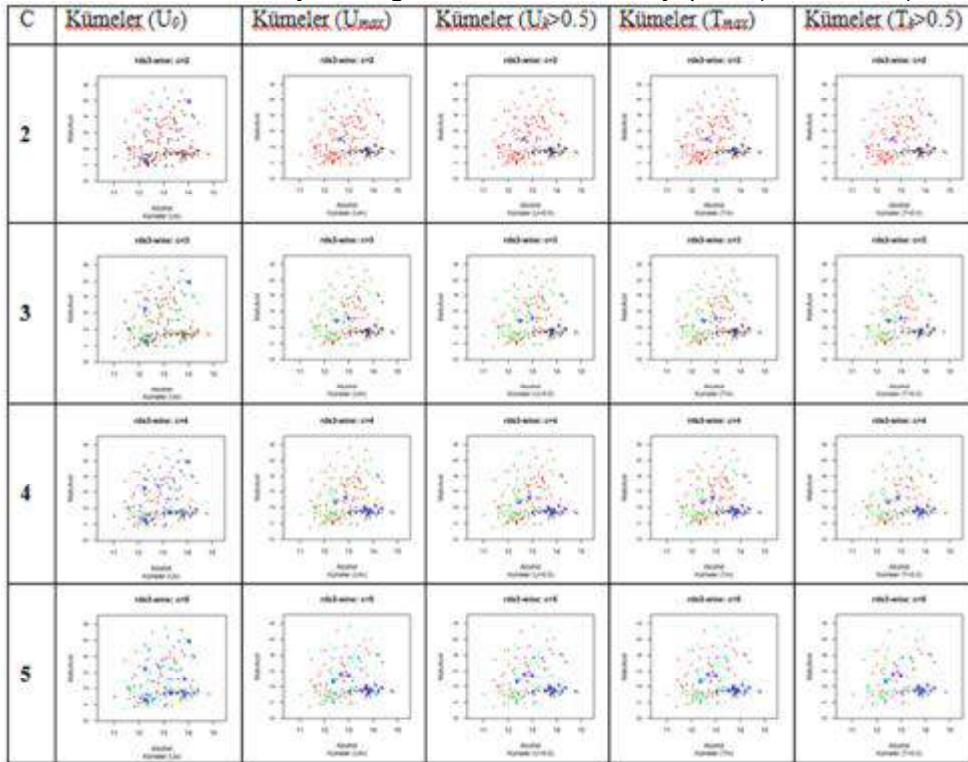
Ek 3.10. BOCO küme sayılarına göre kümeler ve küme yapıları (RDS2-Quail)



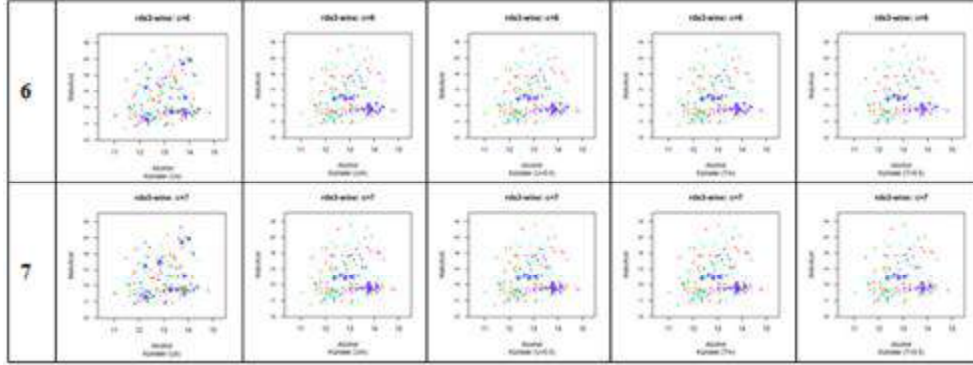
Ek 3.10. (Devamı) BOCO küme sayılarına göre kümeler ve küme yapıları (RDS2-Quail)



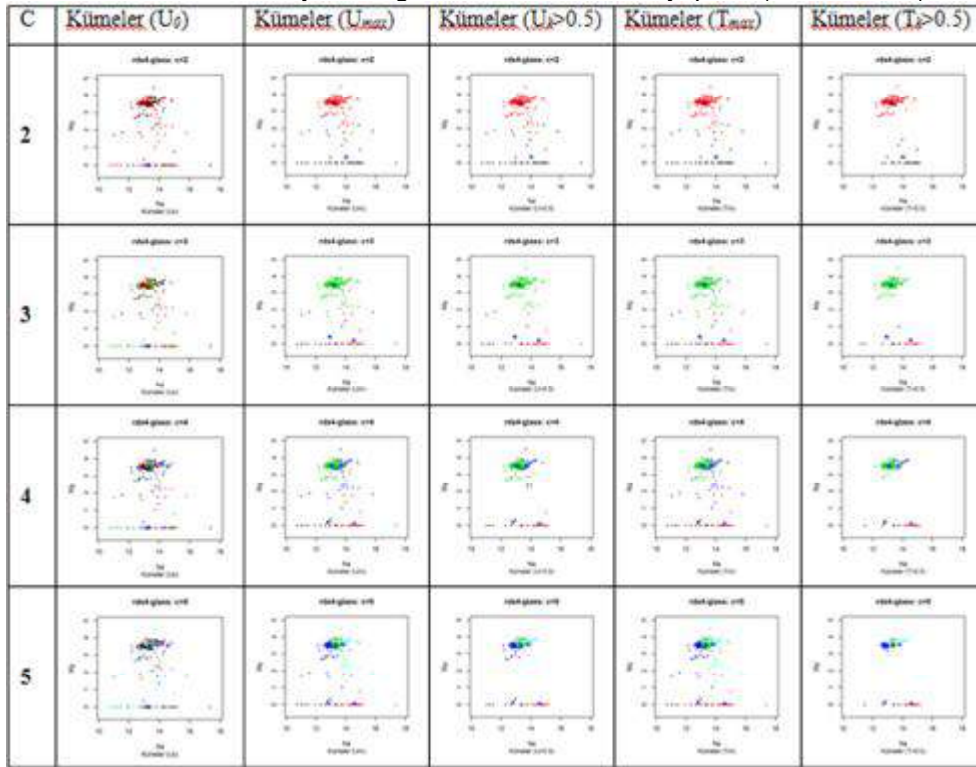
Ek 3.11. BOCO küme sayılarına göre kümeler ve küme yapıları (RDS3-Wine)



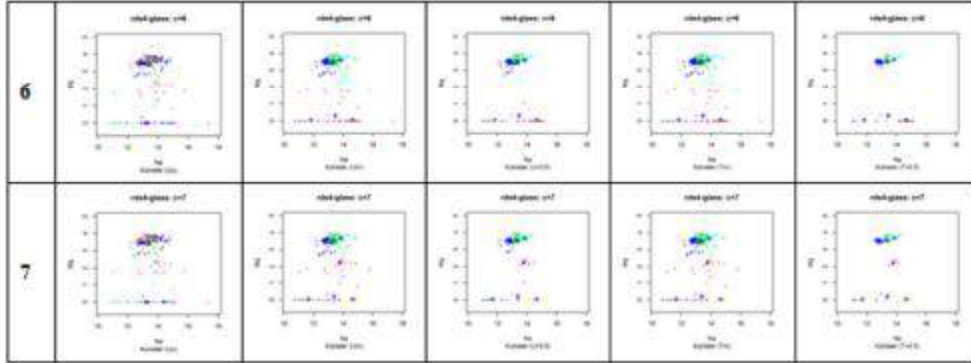
Ek 3.11. (Devamı) BOCO küme sayılarına göre kümeler ve küme yapıları (RDS3-Wine)



Ek 3.12. BOCO küme sayılarına göre kümeler ve küme yapıları (RDS4-Glass)

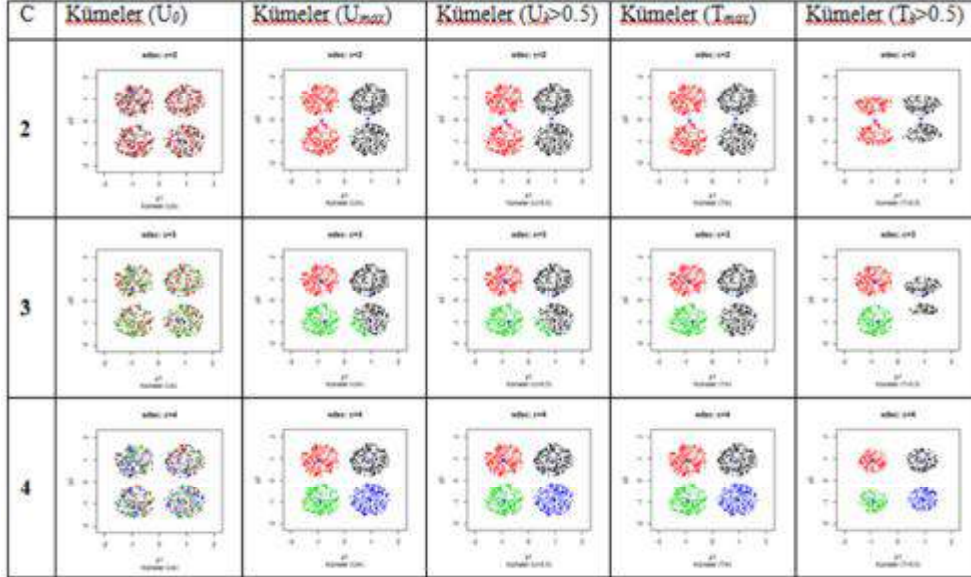


Ek 3.12. (Devamı) BOCO küme sayılarına göre kümeler ve küme yapıları (RDS4-Glass)

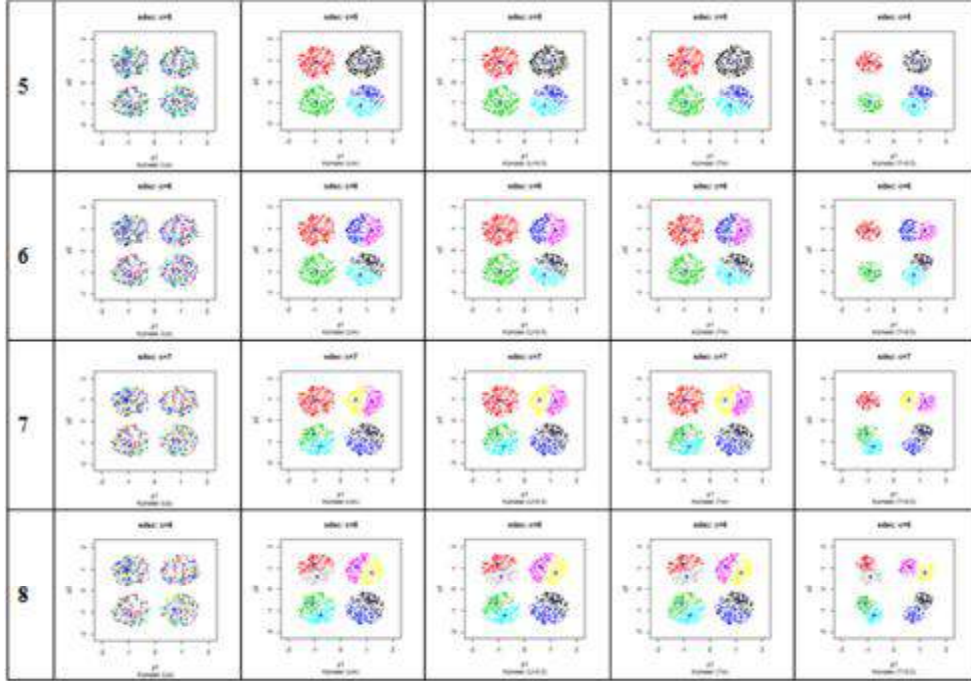


EK 4

Ek 4.1. OBCO küme sayılarına göre kümeler ve küme yapıları (SDSC)



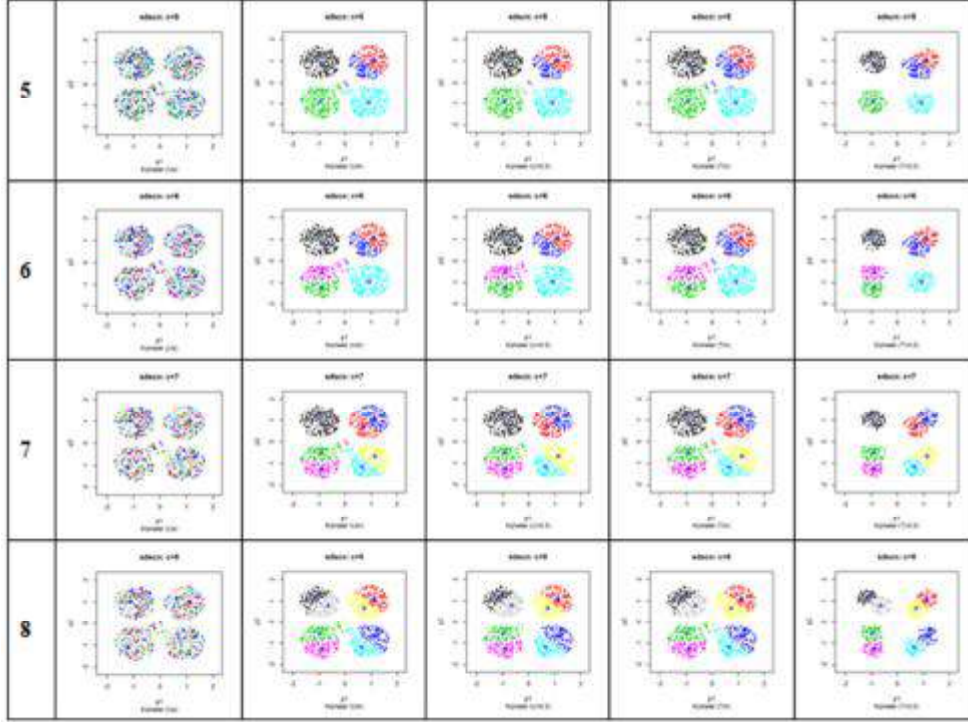
Ek 4.1. (Devamı) OBCO küme sayılarına göre kümeler ve küme yapıları (SDSC)



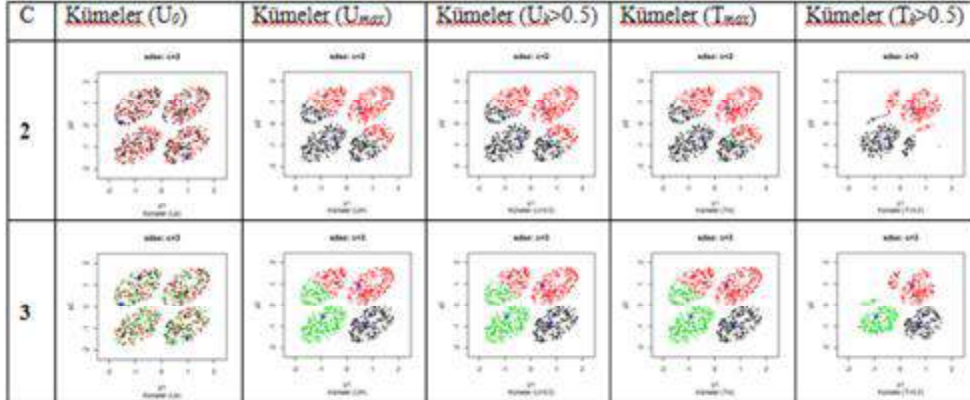
Ek 4.2. OBCO küme sayılarına göre kümeler ve küme yapıları (SDSCN)

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_p > 0.5$)	Kümeler (T_{max})	Kümeler ($T_p > 0.5$)
2					
3					
4					

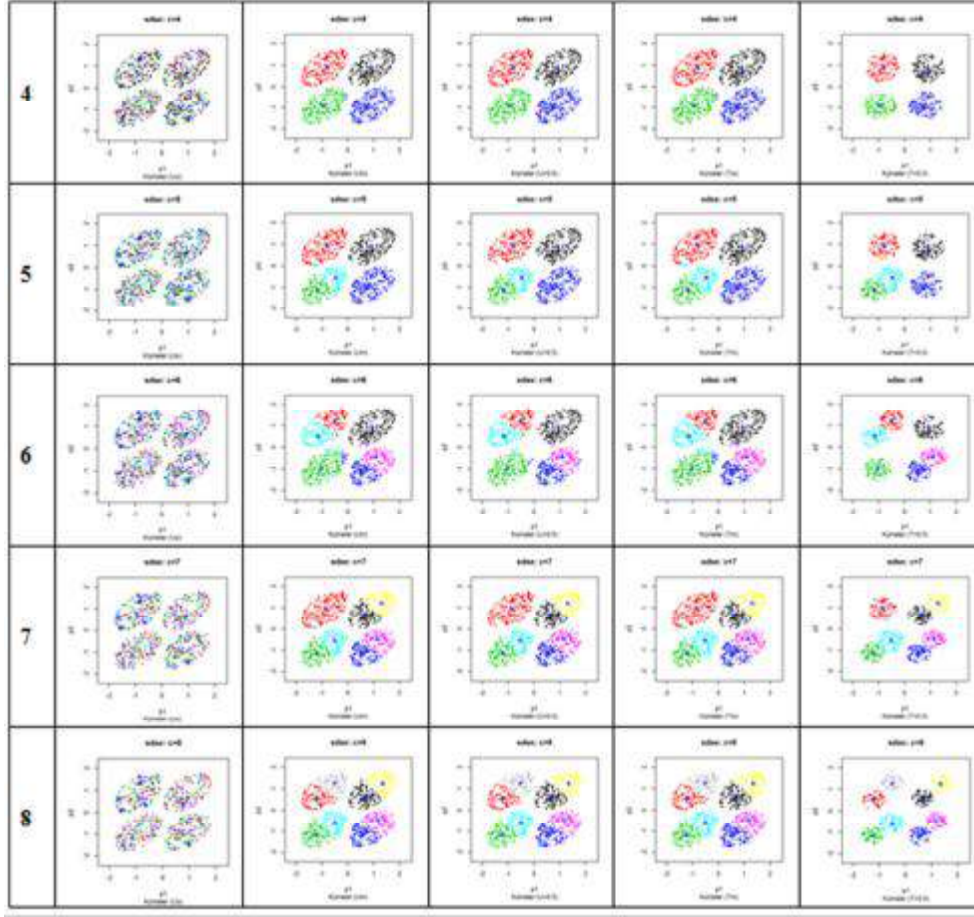
Ek 4.2. (Devamı) OBCO küme saylarına göre kümeler ve küme yapıları (SDSCN)



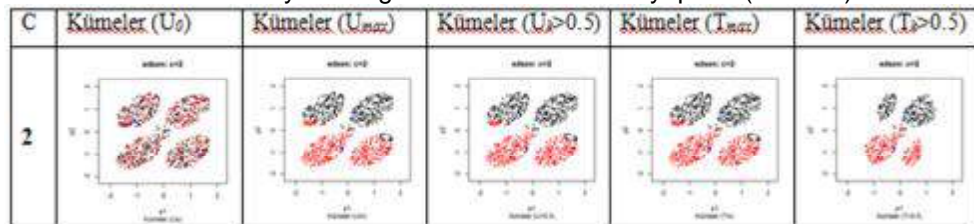
Ek 4.3. OBCO küme saylarına göre kümeler ve küme yapıları (SDSE)



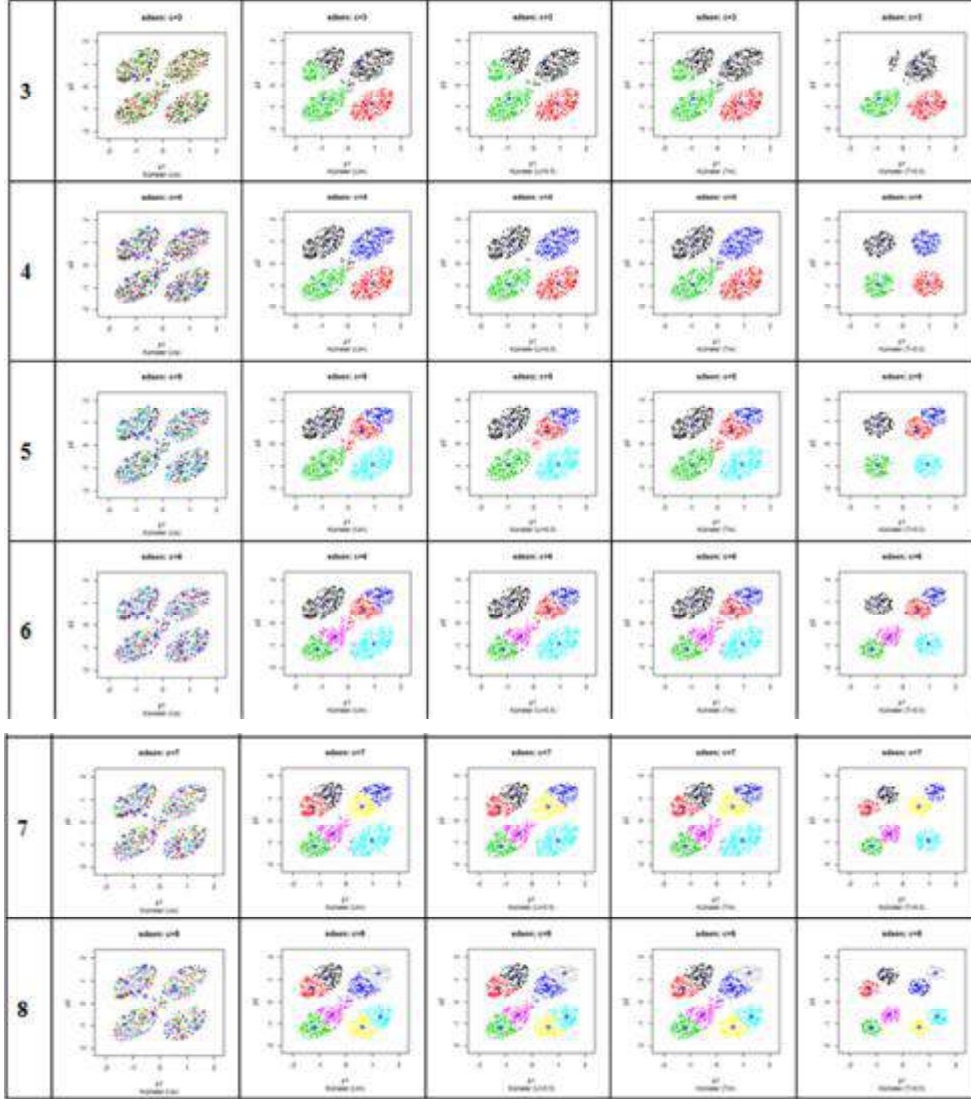
Ek 4.3. (Devamı) OBCO küme saylarına göre kümeler ve küme yapıları (SDSE)



Ek 4.4. OBCO küme saylarına göre kümeler ve küme yapıları (SDSEN)



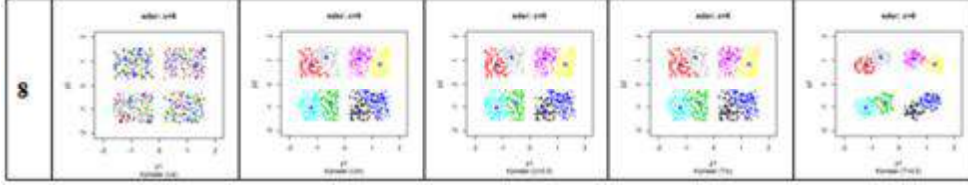
Ek 4.4. (Devamı) OBCO küme saylarına göre kümeler ve küme yapıları (SDSEN)



Ek 4.5. OBCO küme sayılarına göre kümeler ve küme yapıları (SDSR)



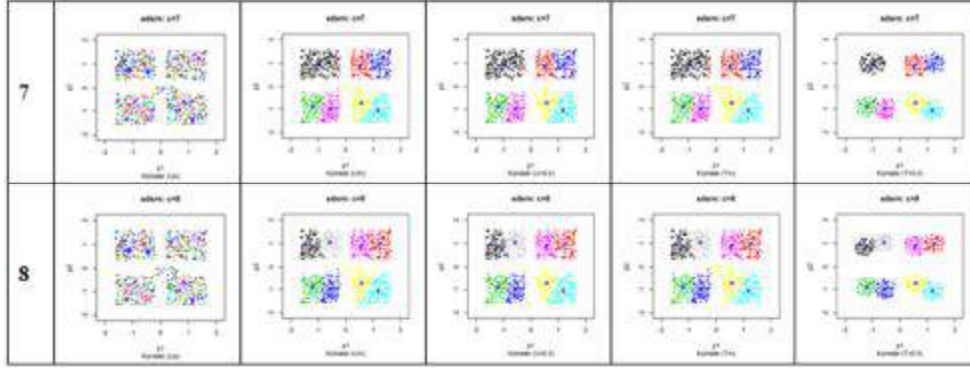
Ek 4.5. (Devamı) OBCO küme saylarına göre kümeler ve küme yapıları (SDSR):



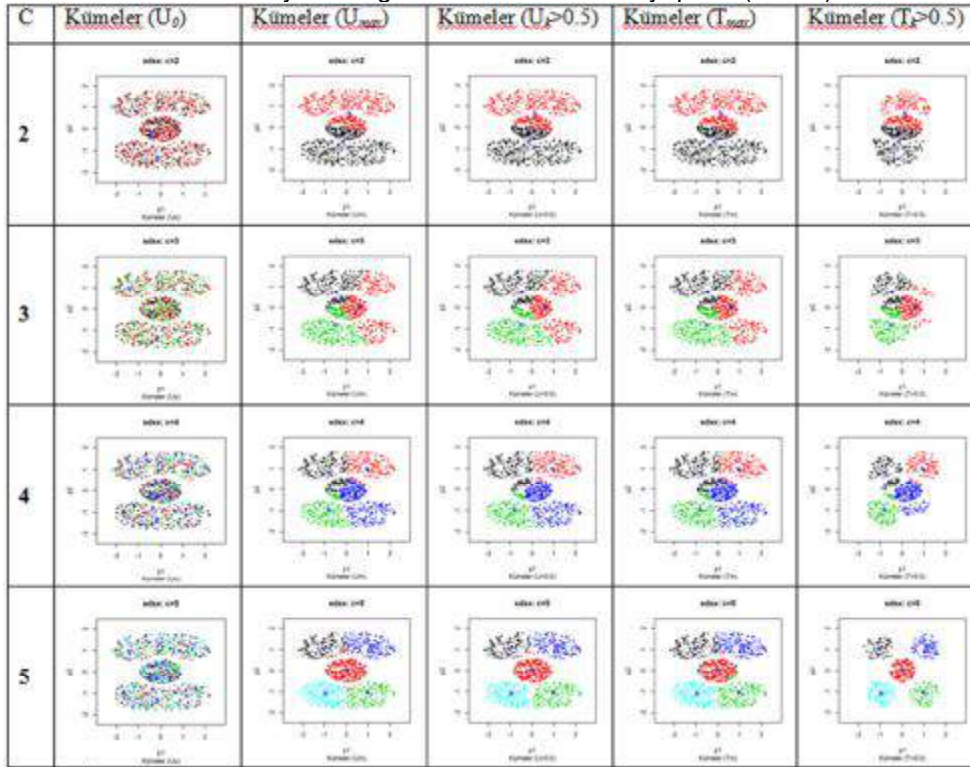
Ek 4.6. OBCO küme saylarına göre kümeler ve küme yapıları (SDSRN):

C	Küme (U ₀)	Küme (U _{max})	Küme (U _k >0.5)	Küme (T _{max})	Küme (T _k >0.5)
2					
3					
4					
5					
6					

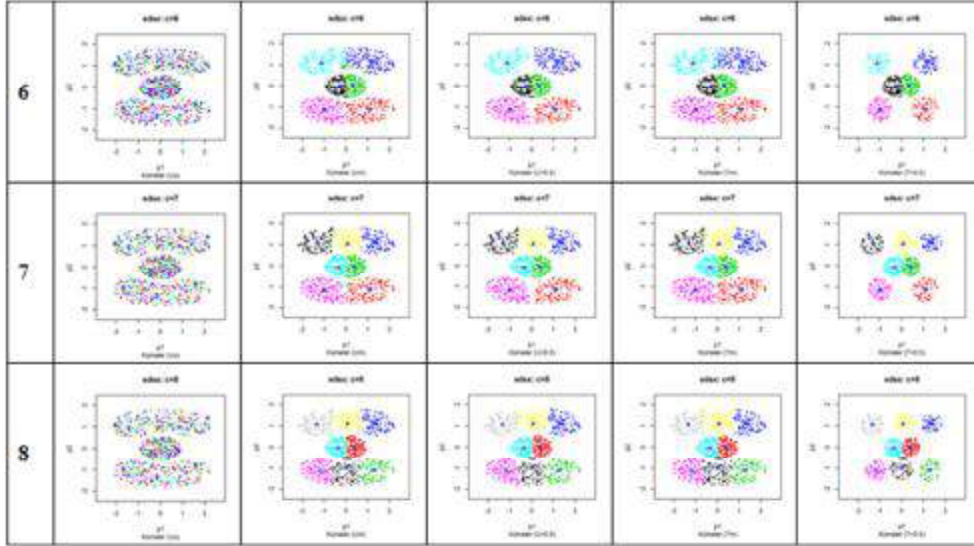
Ek 4.6. (Devamı) OBCO küme saylarına göre kümeler ve küme yapıları (SDSRN):



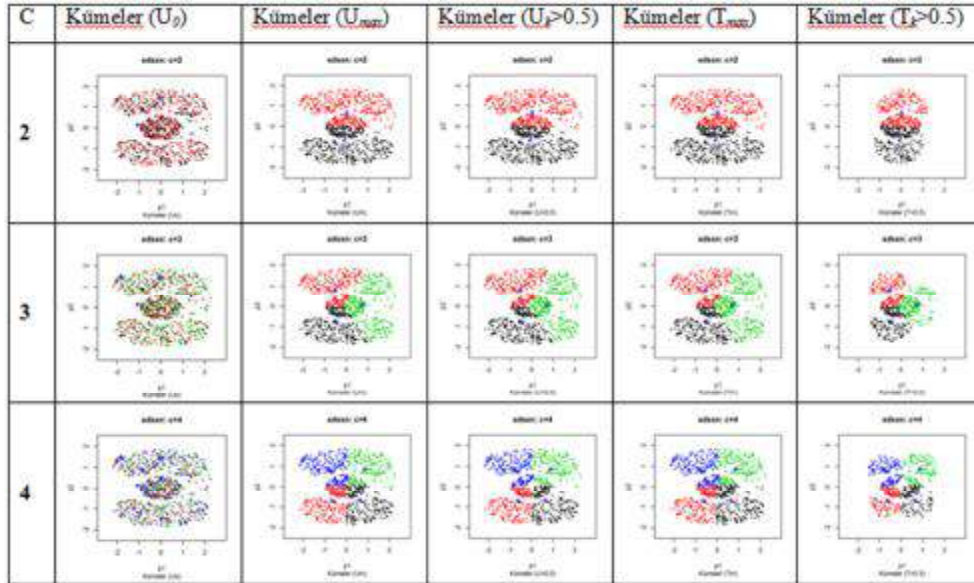
Ek 4.7. OBCO küme saylarına göre kümeler ve küme yapıları (SDSX):



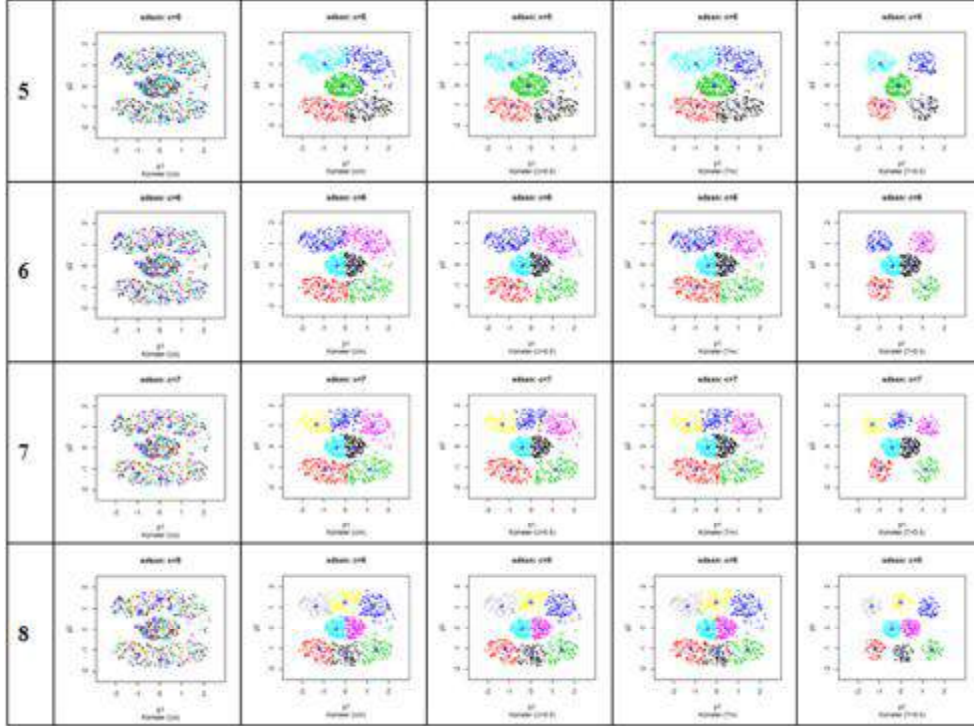
Ek 4.7. (Devamı) OBCO küme sayılarına göre kümeler ve küme yapıları (SDSX):



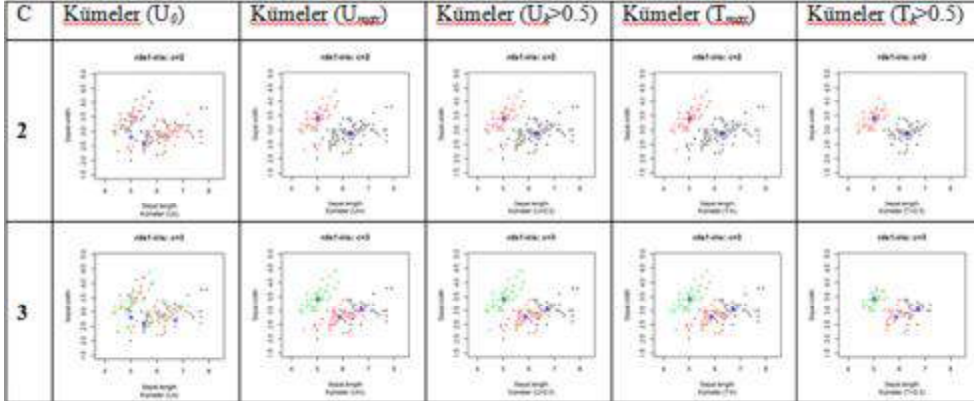
Ek 4.8. OBCO küme sayılarına göre kümeler ve küme yapıları (SDSXN):



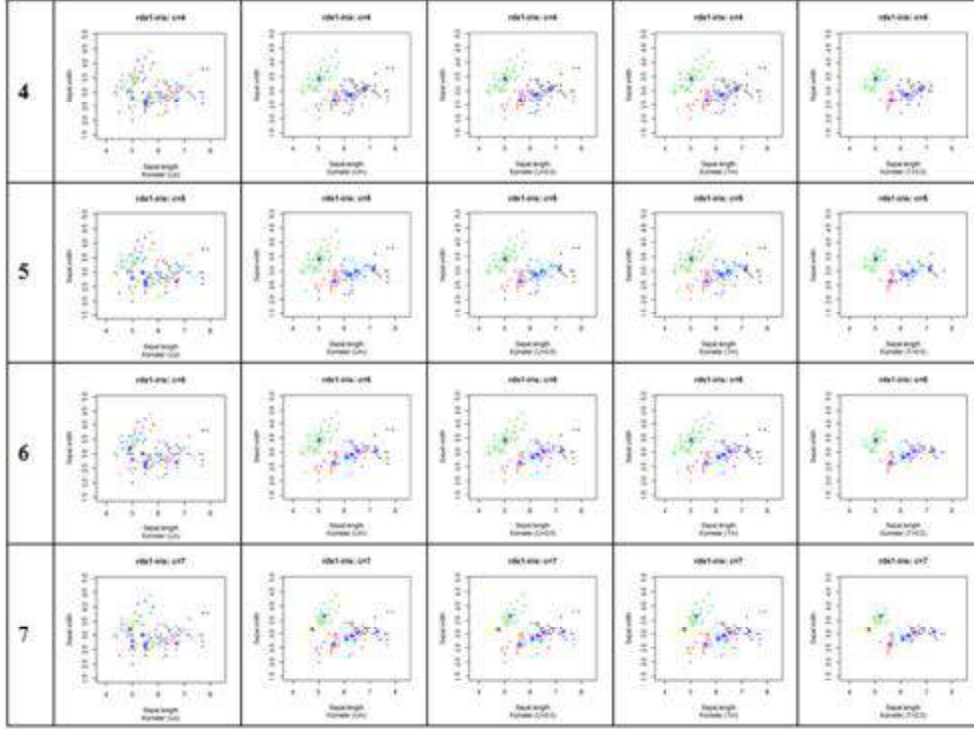
Ek 4.8. (Devamı) OBCO küme sayılarına göre kümeler ve küme yapıları (SDSXN):



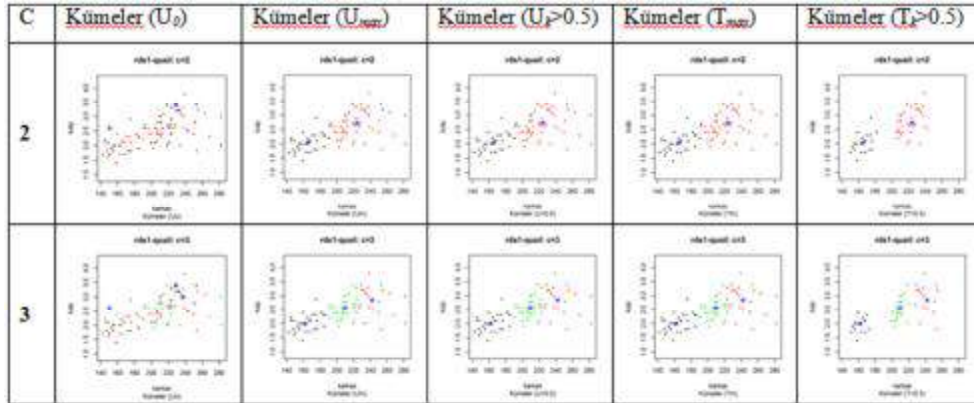
Ek 4.9. OBCO küme sayılarına göre kümeler ve küme yapıları (RDS1-Iris):



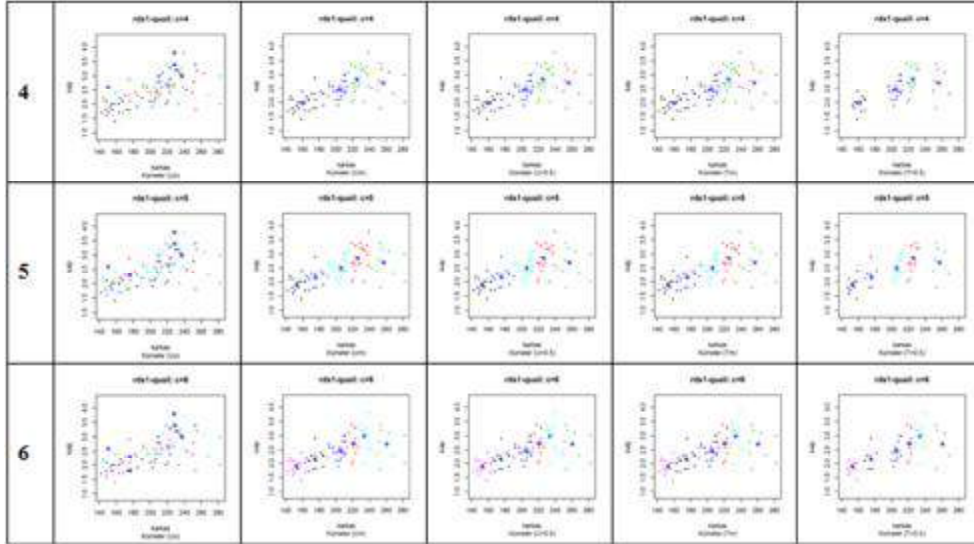
Ek 4.9. (Devamı) OBCO küme sayılarına göre kümeler ve küme yapıları (RDS1-Iris):



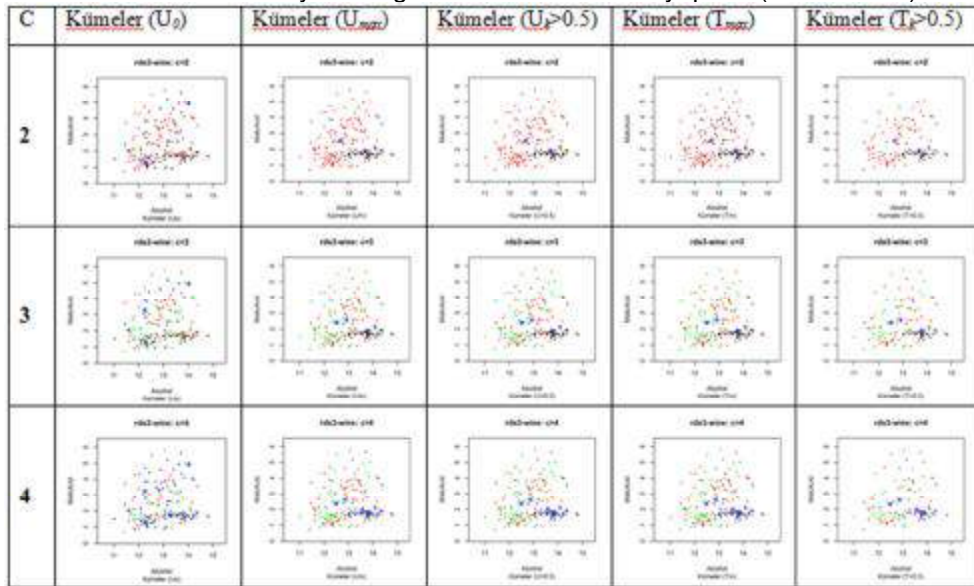
Ek 4.10. OBCO küme sayılarına göre kümeler ve küme yapıları (RDS2-Quail):



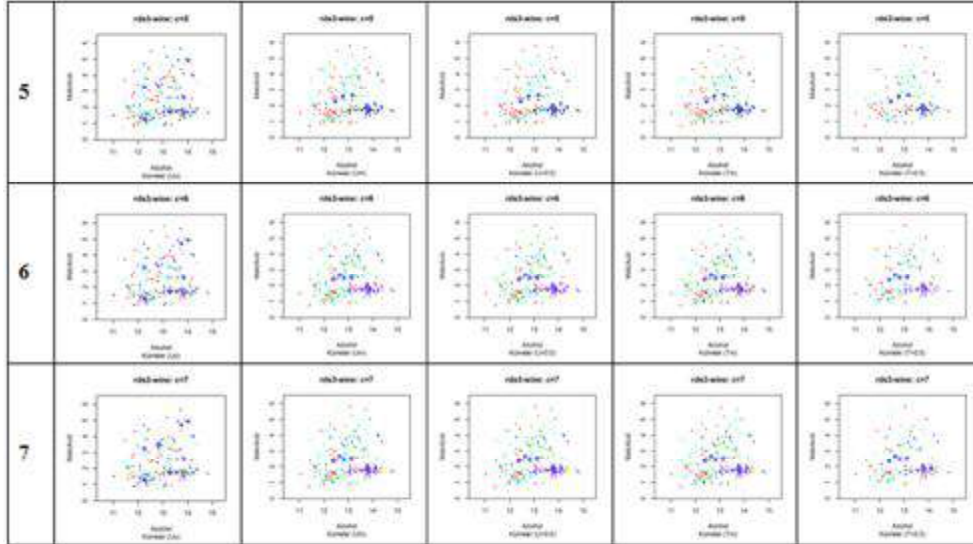
Ek 4.10. (Devamı) OBCO küme sayılarına göre kümeler ve küme yapıları (RDS2-Quail):



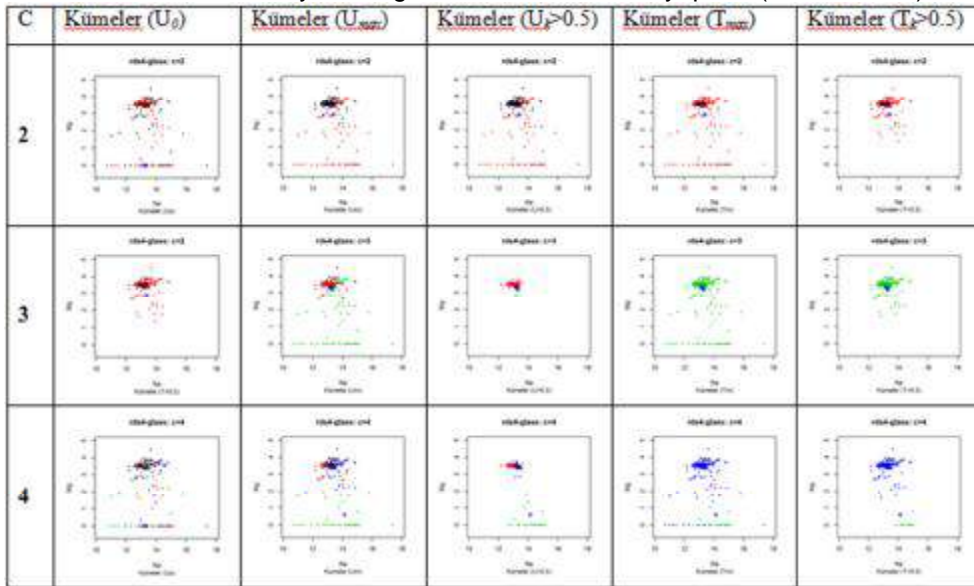
Ek 4.11. OBCO küme sayılarına göre kümeler ve küme yapıları (RDS3-Wine):



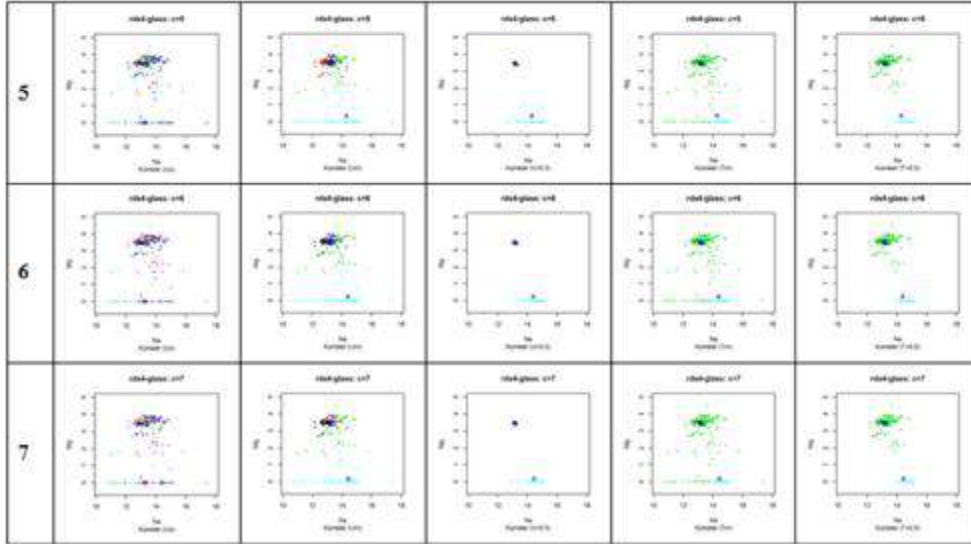
Ek 4.11. (Devamı) OBCO küme sayılarına göre kümeler ve küme yapıları (RDS3-Wine):



Ek 4.12. OBCO küme sayılarına göre kümeler ve küme yapıları (RDS4-Glass):

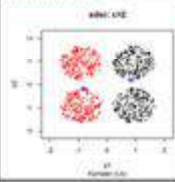
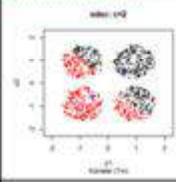
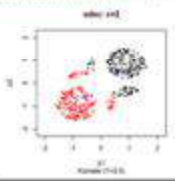
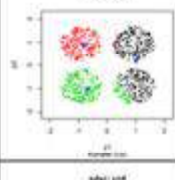
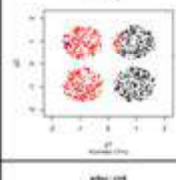
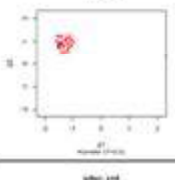
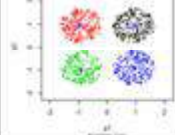
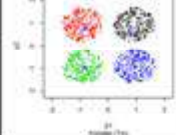
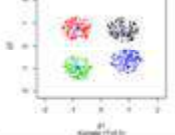


Ek 4.12. (Devamı) OBCO küme sayılarına göre kümeler ve küme yapıları (RDS4-Glass):

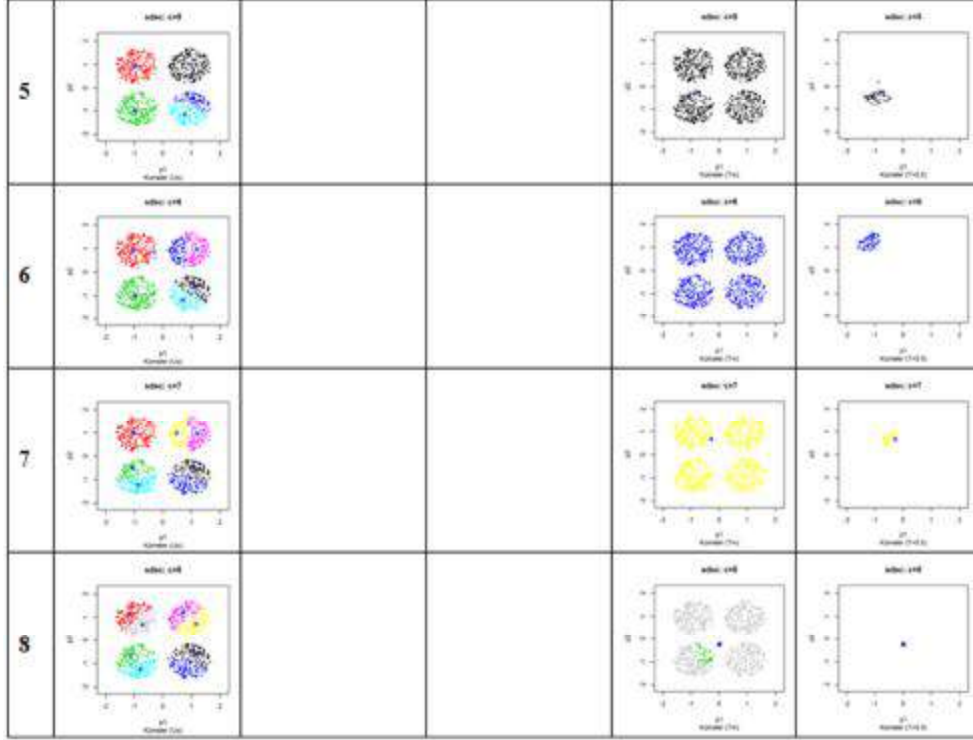


EK 5

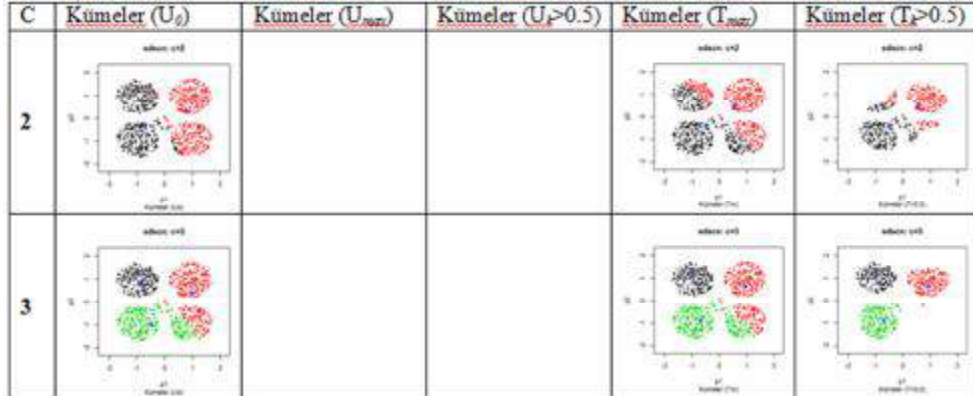
Ek 5.1. İOCO küme sayılarına göre kümeler ve küme yapıları (SDSC):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_{f>0.5}$)	Kümeler (T_{max})	Kümeler ($T_{f>0.5}$)
2					
3					
4					

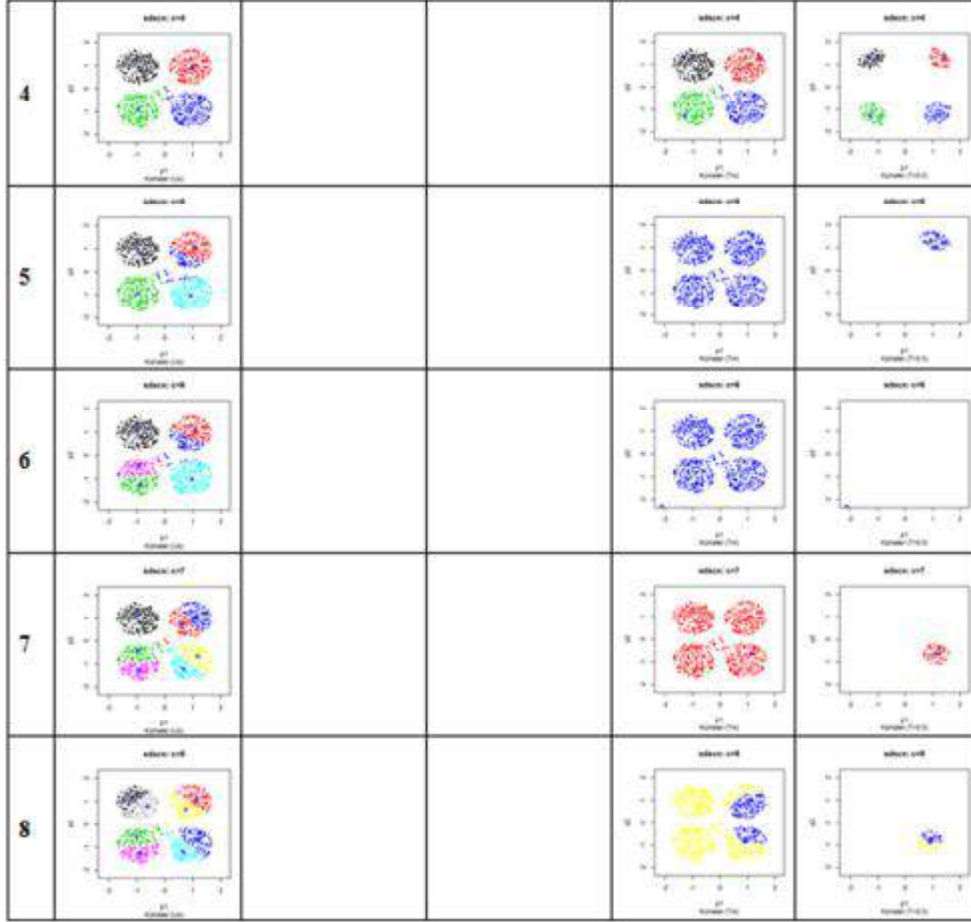
Ek 5.1. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (SDSC):



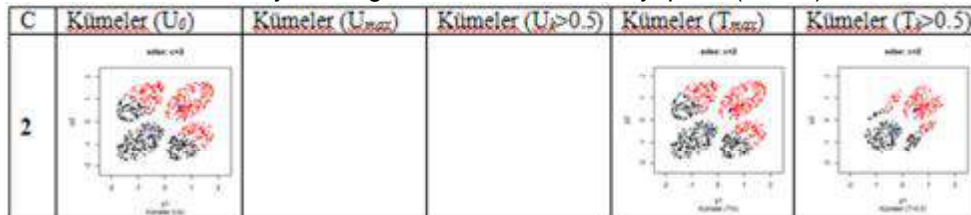
Ek 5.2. İOCO küme sayılarına göre kümeler ve küme yapıları (SDSCN):



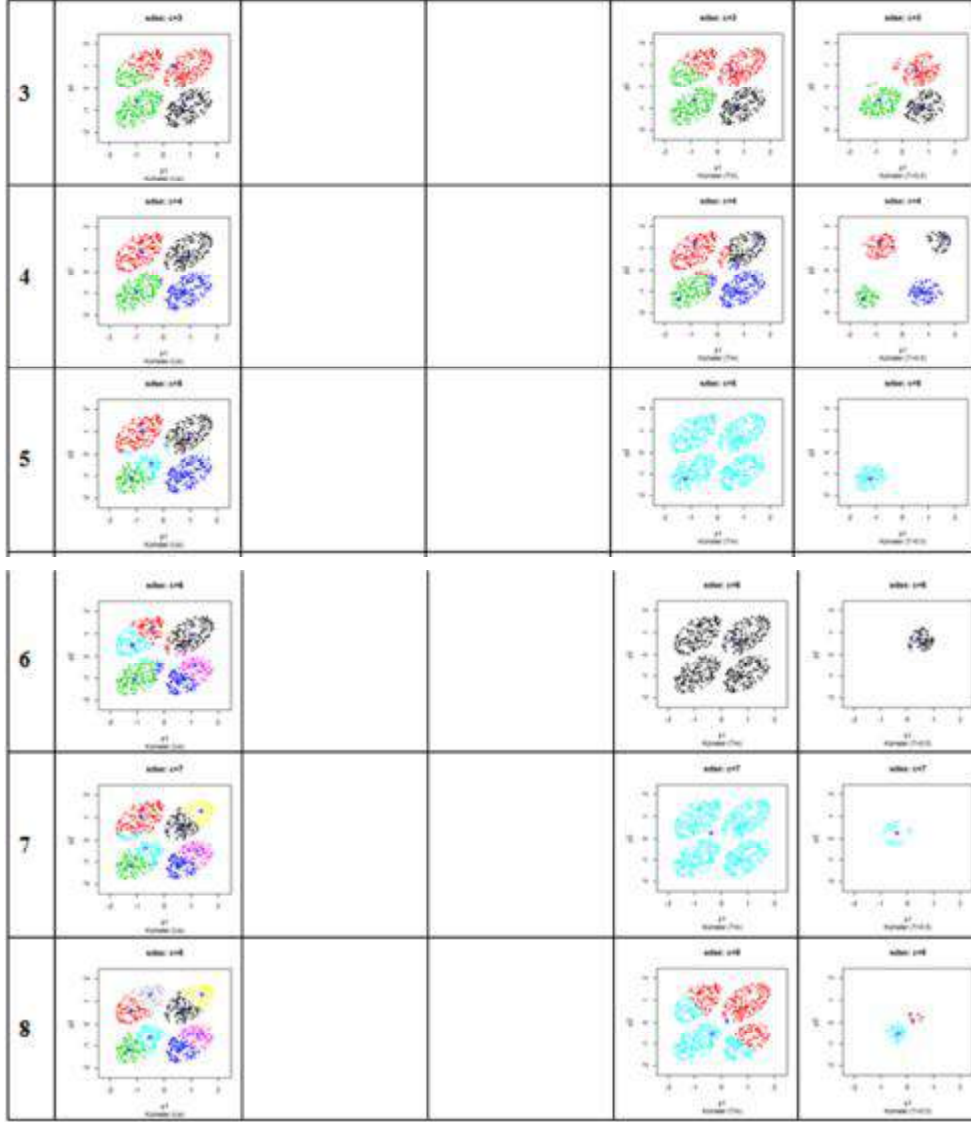
Ek 5.2. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (SDSCN):



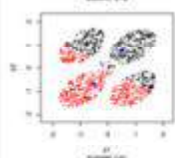
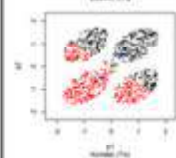
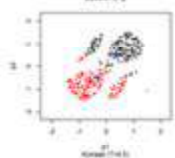
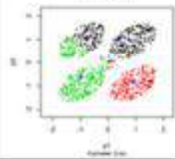
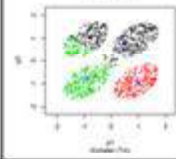
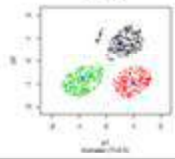
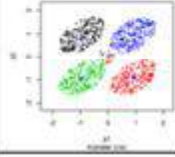
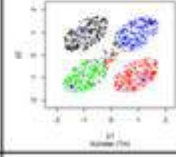
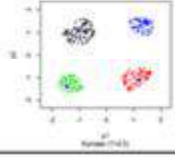
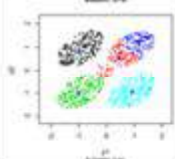
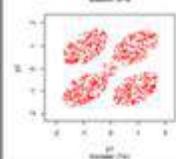
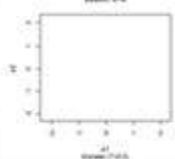
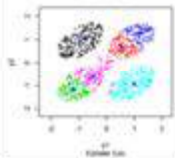
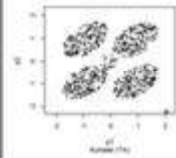
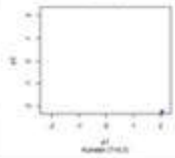
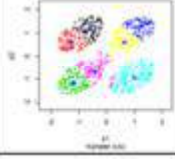
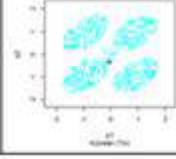
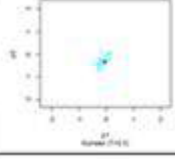
Ek 5.3. İOCO küme sayılarına göre kümeler ve küme yapıları (SDSE):



Ek 5.3. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (SDSE):



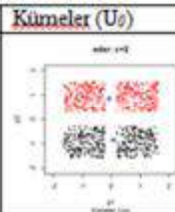
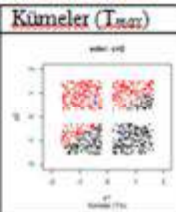
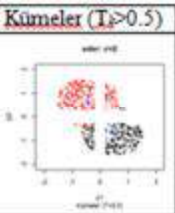
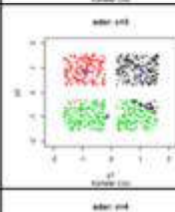
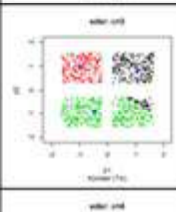
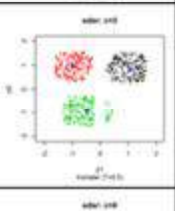
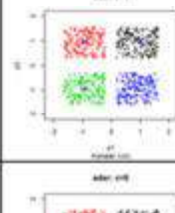
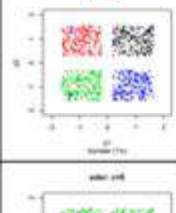
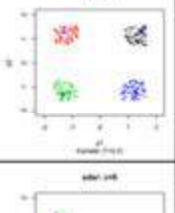
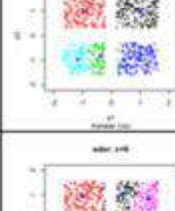
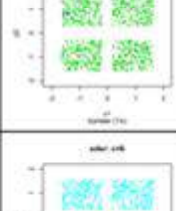
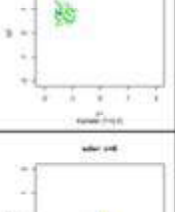
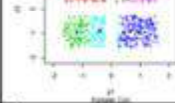
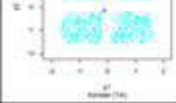
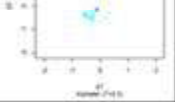
Ek 5.4. İOCO küme sayılarına göre kümeler ve küme yapıları (SDSEN):

C	Kümeler (U_{θ})	Kümeler ($U_{\theta_{opt}}$)	Kümeler ($U_{\theta > 0.5}$)	Kümeler ($T_{\theta_{opt}}$)	Kümeler ($T_{\theta > 0.5}$)
2					
3					
4					
5					
6					
7					

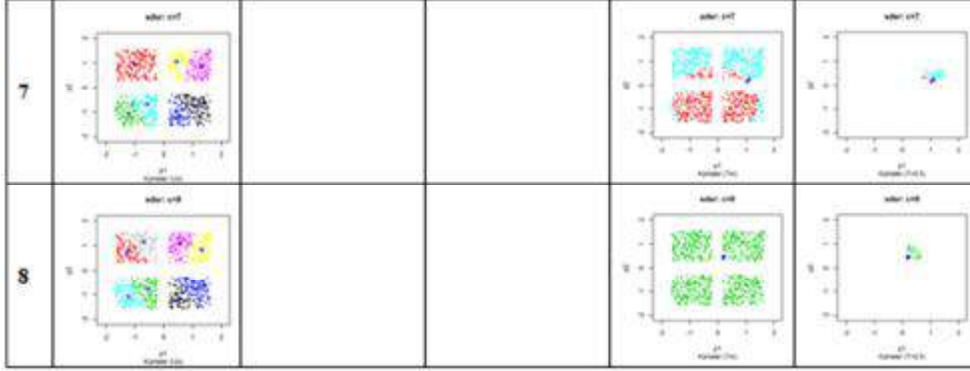
Ek 5.4. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (SDSEN):



Ek 5.5. İOCO küme sayılarına göre kümeler ve küme yapıları (SDSR):

C	Kümeler (U_0)	Kümeler (U_{max})	Kümeler ($U_{\lambda>0.5}$)	Kümeler (T_{max})	Kümeler ($T_{\lambda>0.5}$)
2					
3					
4					
5					
6					

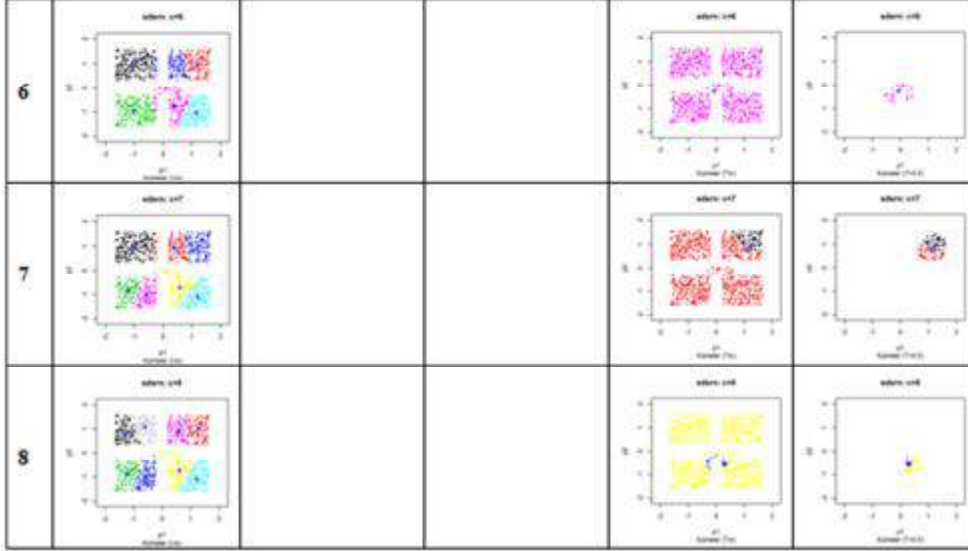
Ek 5.5. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (SDSR):



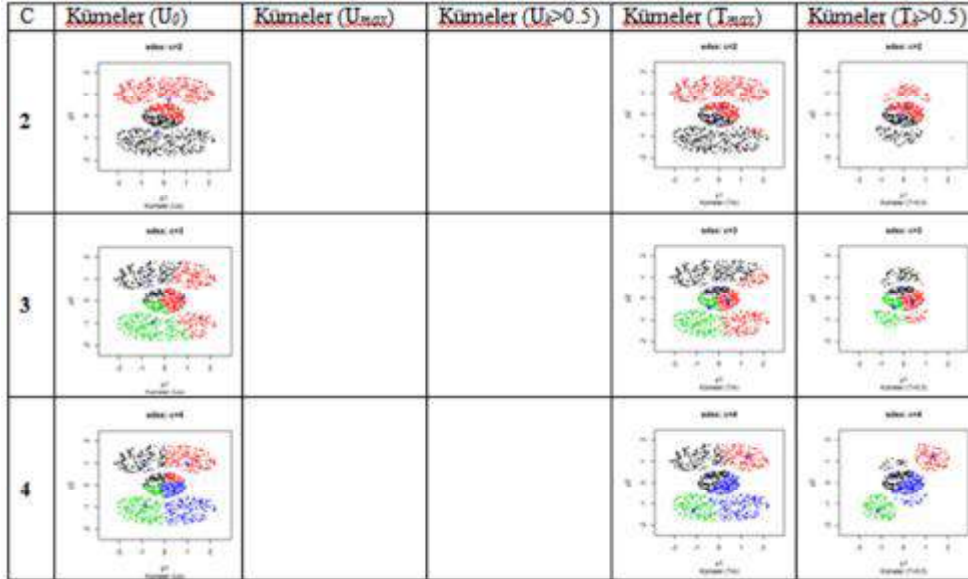
Ek 5.6. İOCO küme sayılarına göre kümeler ve küme yapıları (SDSRN):



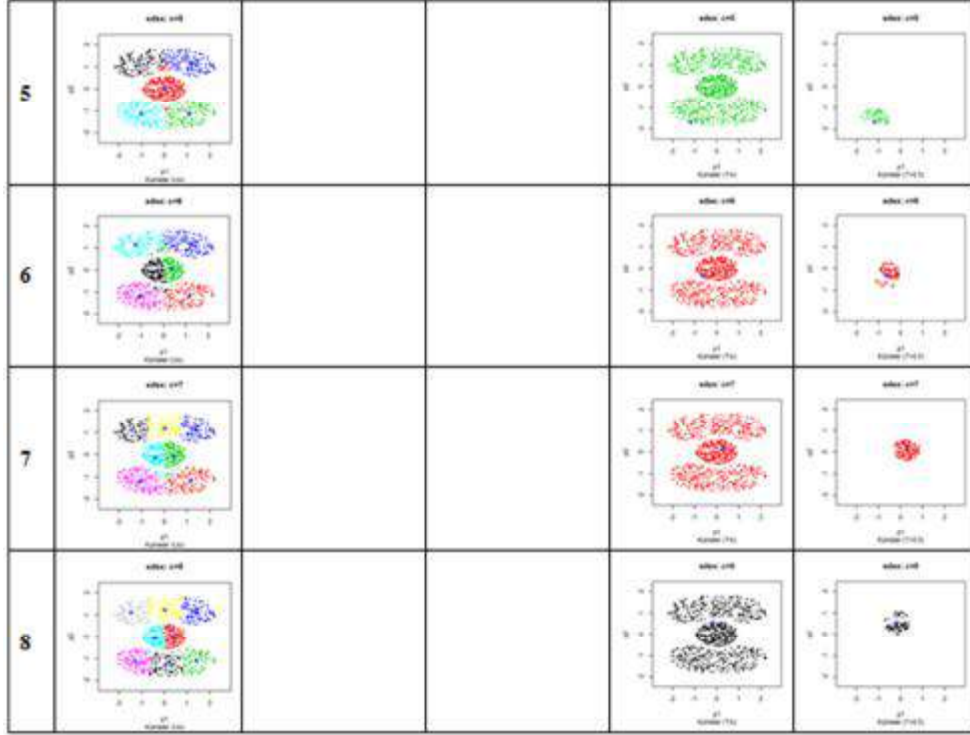
Ek 5.6. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (SDSRN):



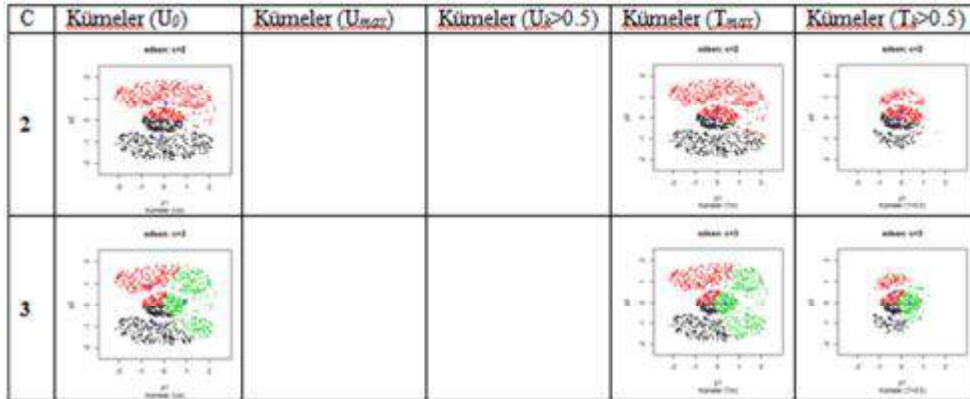
Ek 5.7. İOCO küme sayılarına göre kümeler ve küme yapıları (SDSX):



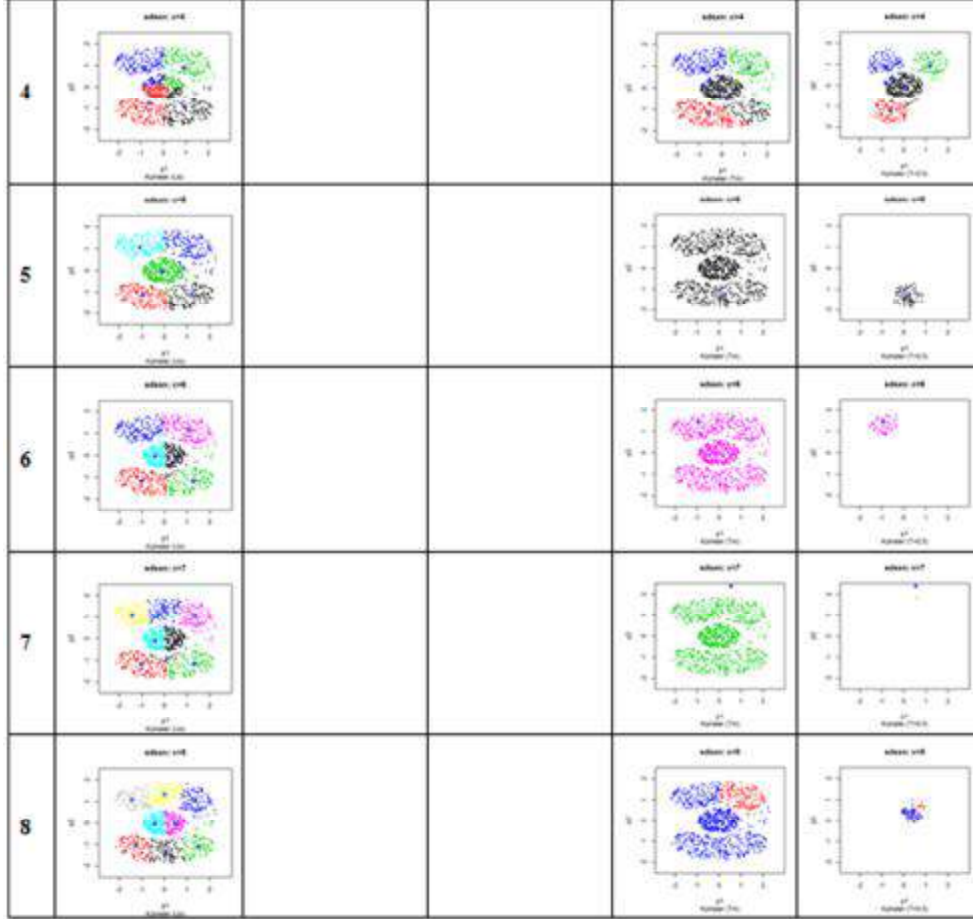
Ek 5.7. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (SDSX):



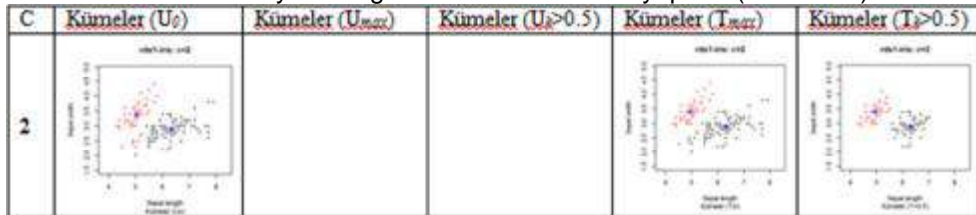
Ek 5.8. İOCO küme sayılarına göre kümeler ve küme yapıları (SDSXN):



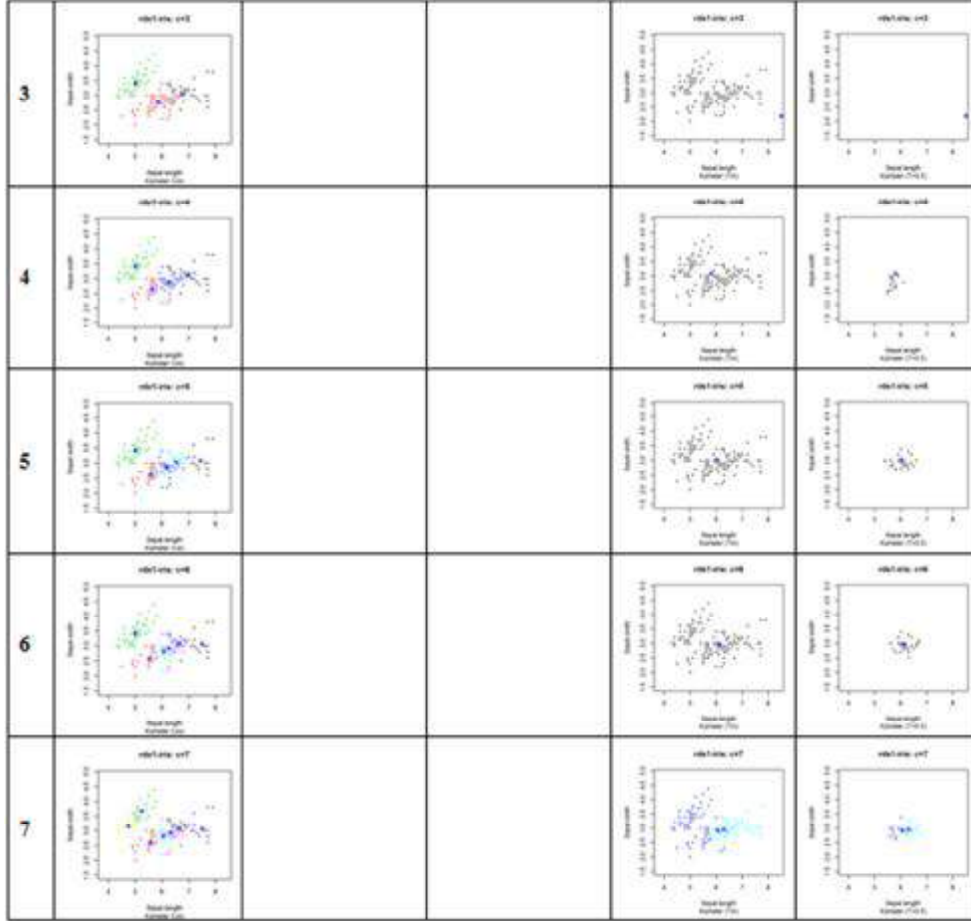
Ek 5.8. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (SDSXN):



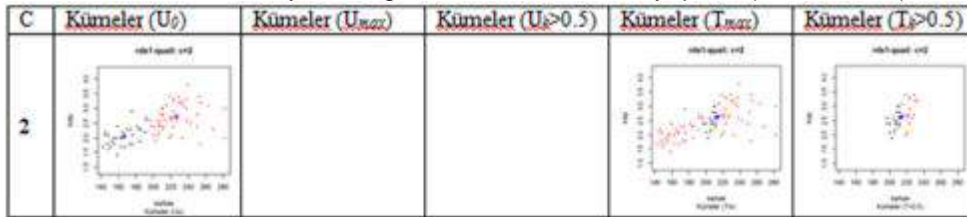
Ek 5.9. İOCO küme sayılarına göre kümeler ve küme yapıları (RDS1-Iris):



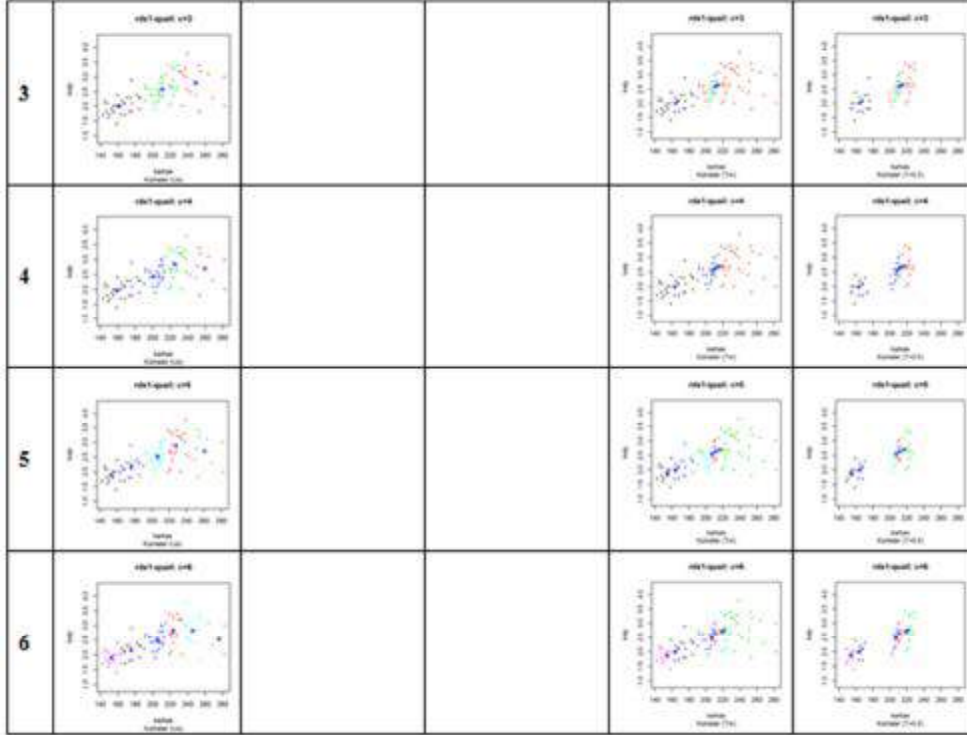
Ek 5.9. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (RDS1-Iris):



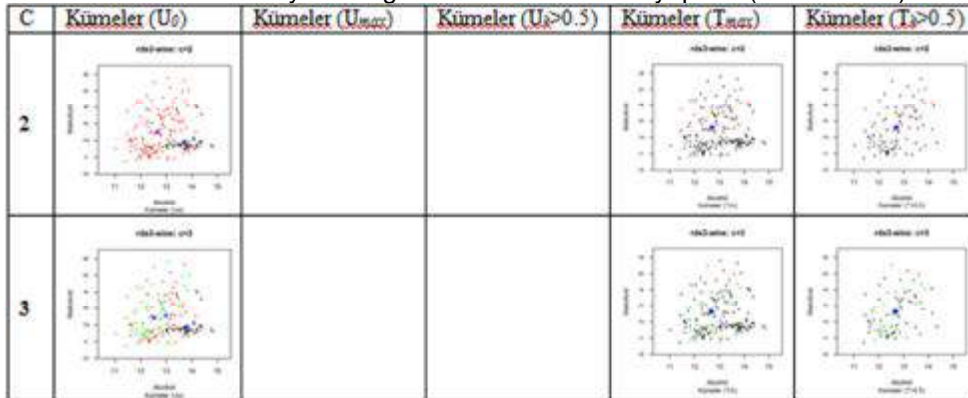
Ek 5.10. İOCO küme sayılarına göre kümeler ve küme yapıları (RDS2-Quail):



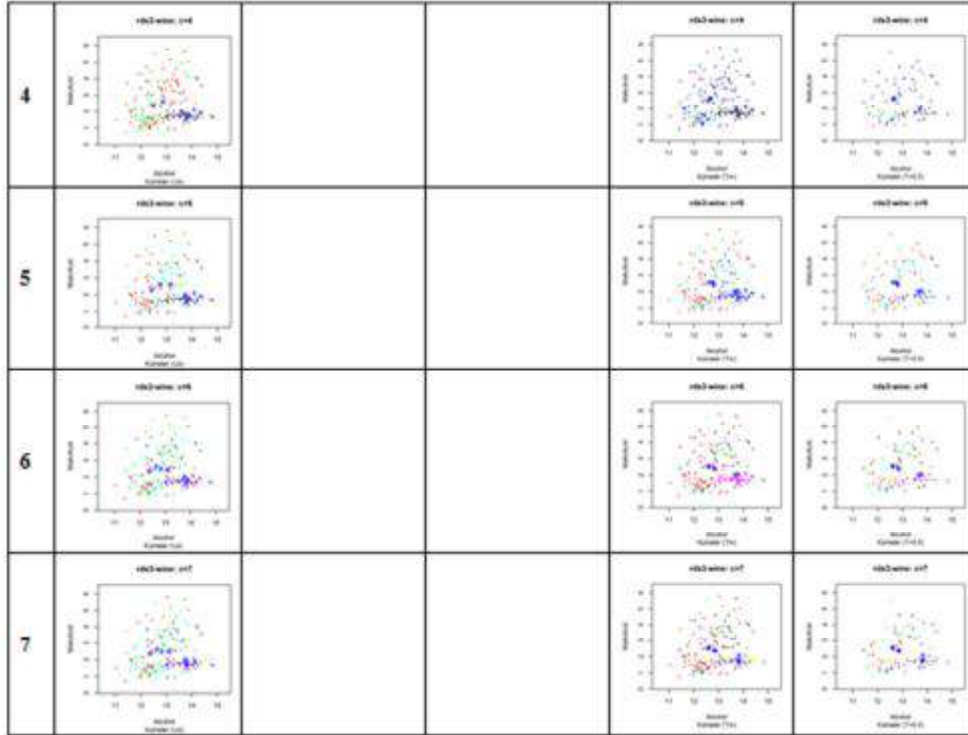
Ek 5.10. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (RDS2-Quail):



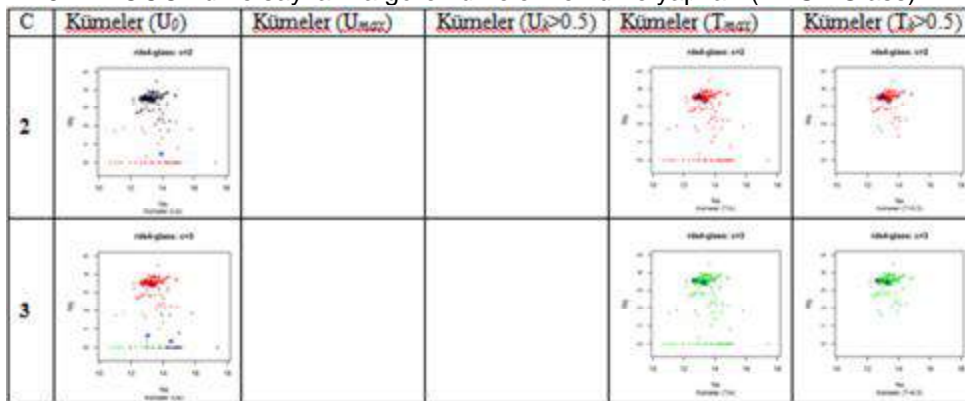
Ek 5.11. İOCO küme sayılarına göre kümeler ve küme yapıları (RDS3-Wine):



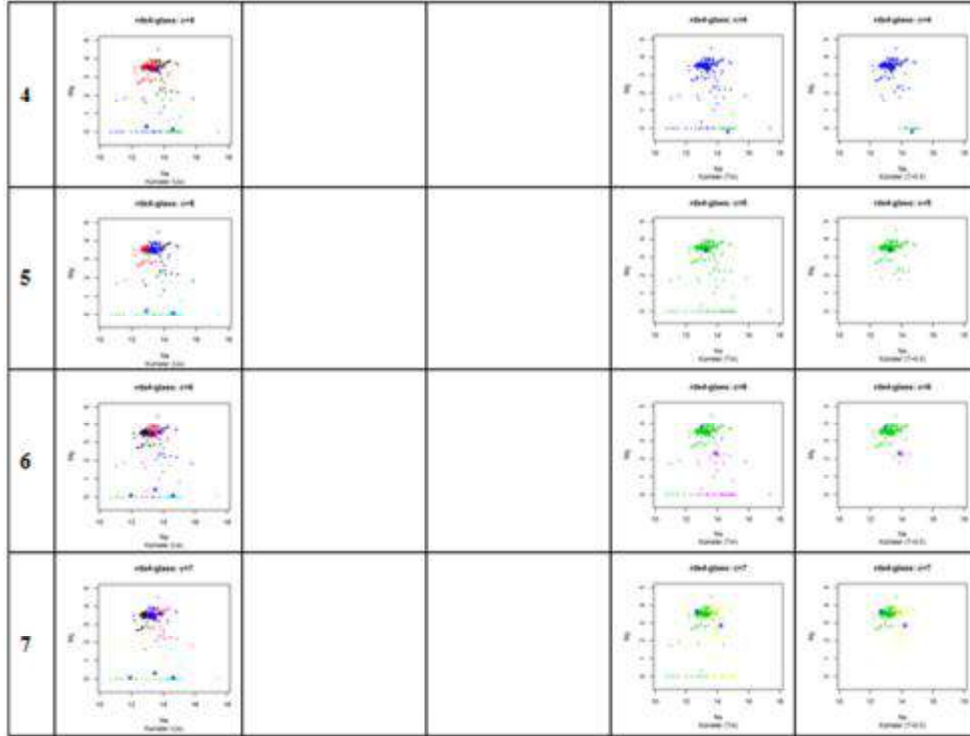
Ek 5.11. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (RDS3-Wine):



Ek 5.12. İOCO küme sayılarına göre kümeler ve küme yapıları (RDS4-Glass):

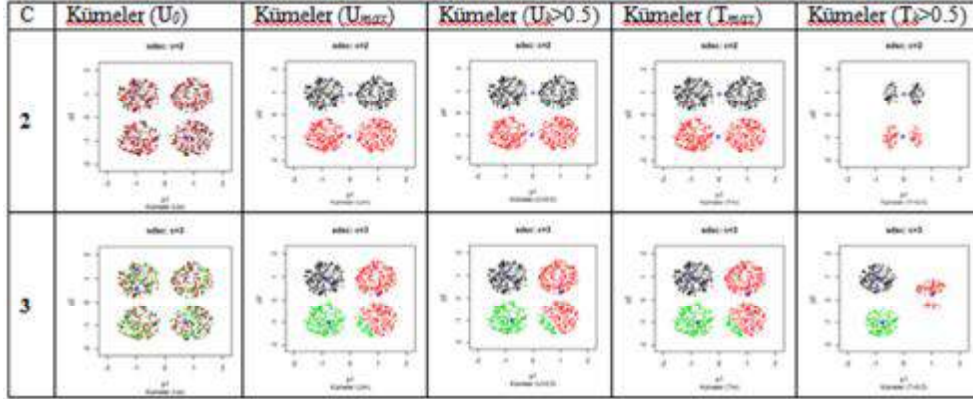


Ek 5.12. (Devamı) İOCO küme sayılarına göre kümeler ve küme yapıları (RDS4-Glass):



EK 6

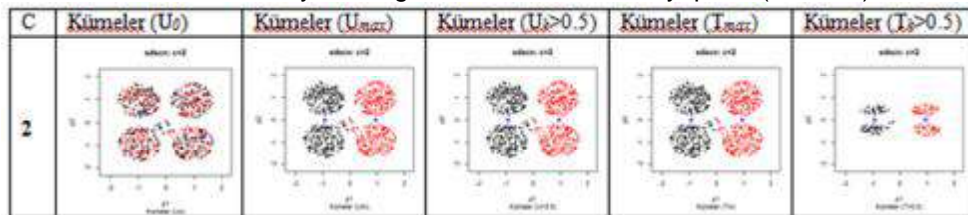
Ek 6.1. UOBCO küme saylarına göre kümeler ve küme yapıları (SDSC):



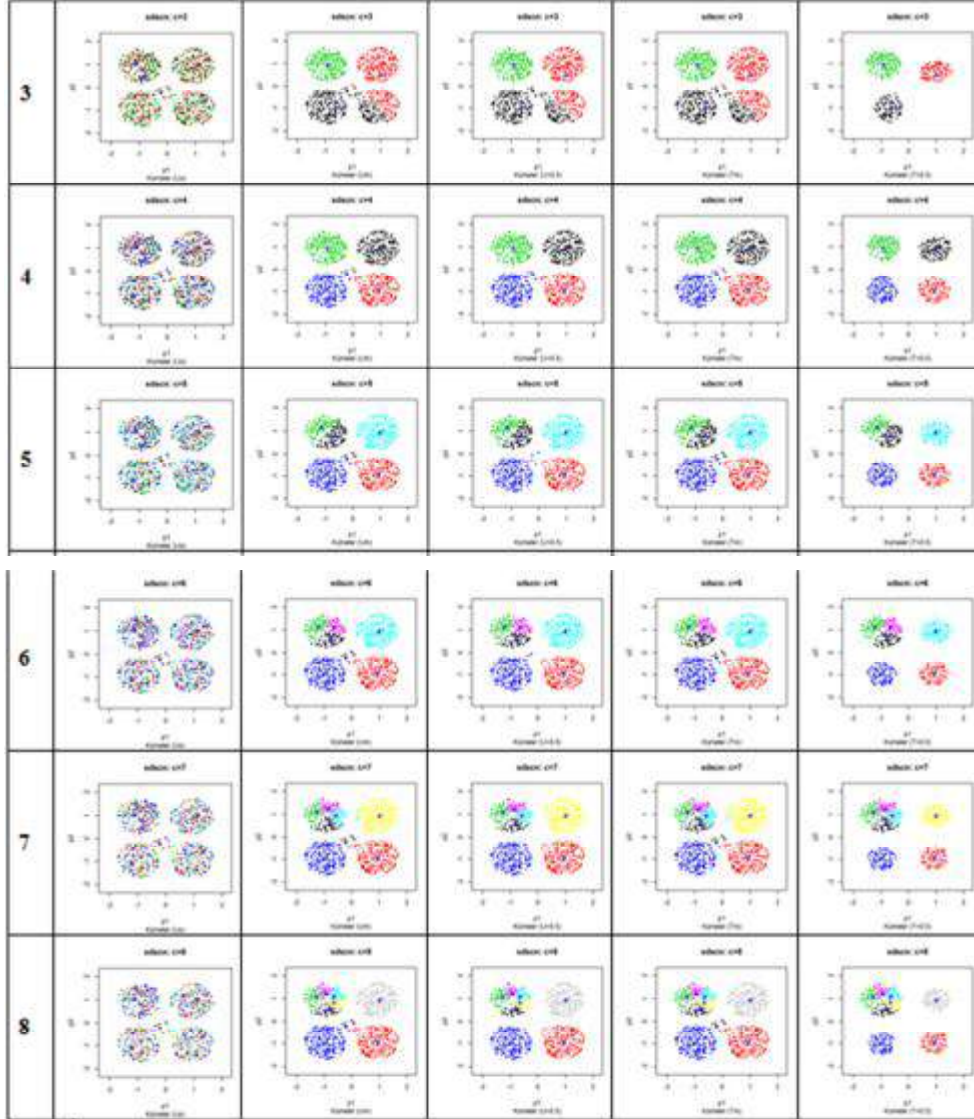
Ek 6.1. (Devamı) UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSC):



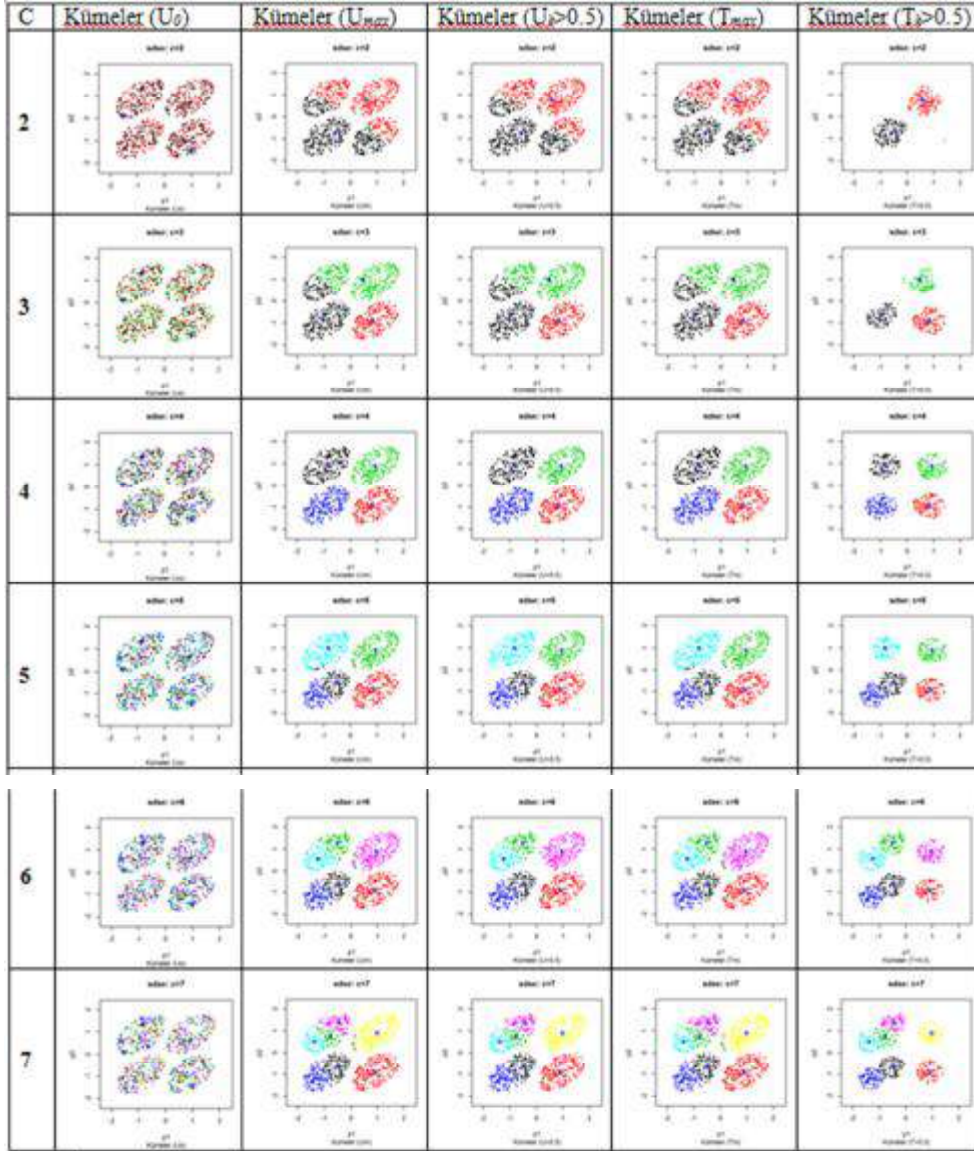
Ek 6.2. UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSCN):



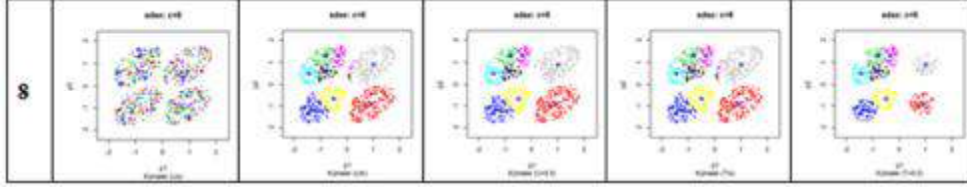
Ek 6.2. (Devamı) UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSCN):



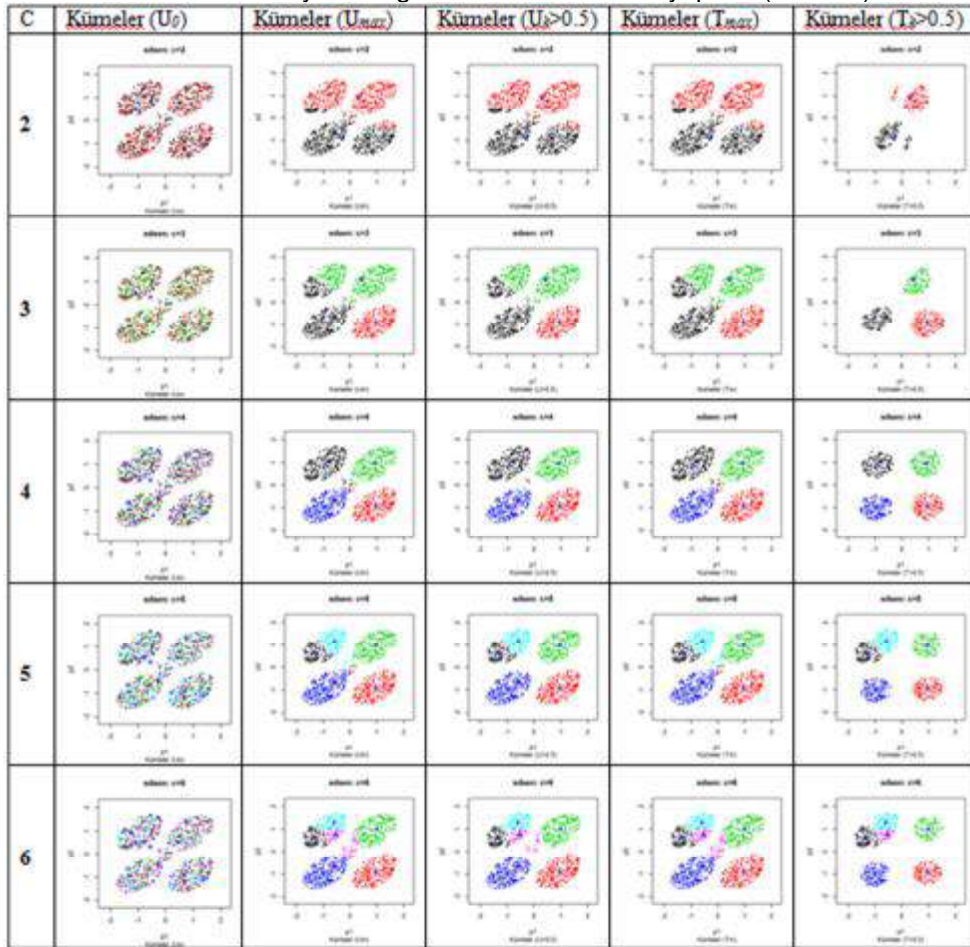
Ek 6.3. UOBKO küme saylarına göre kümeler ve küme yapıları (SDSE):



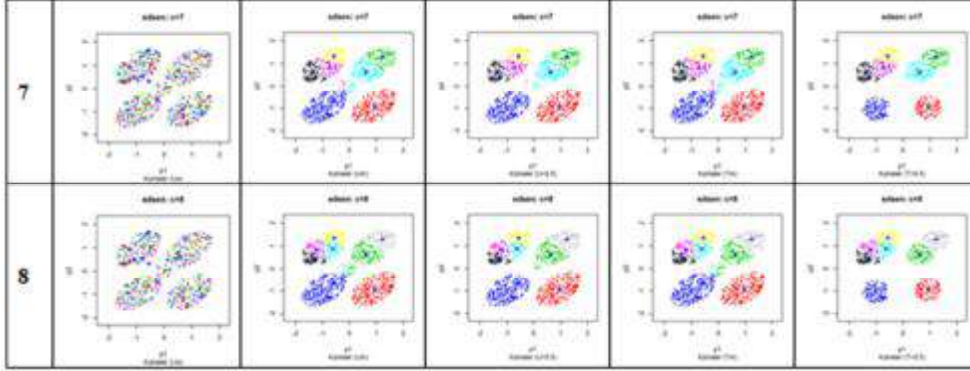
Ek 6.3. (Devamı) UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSE):



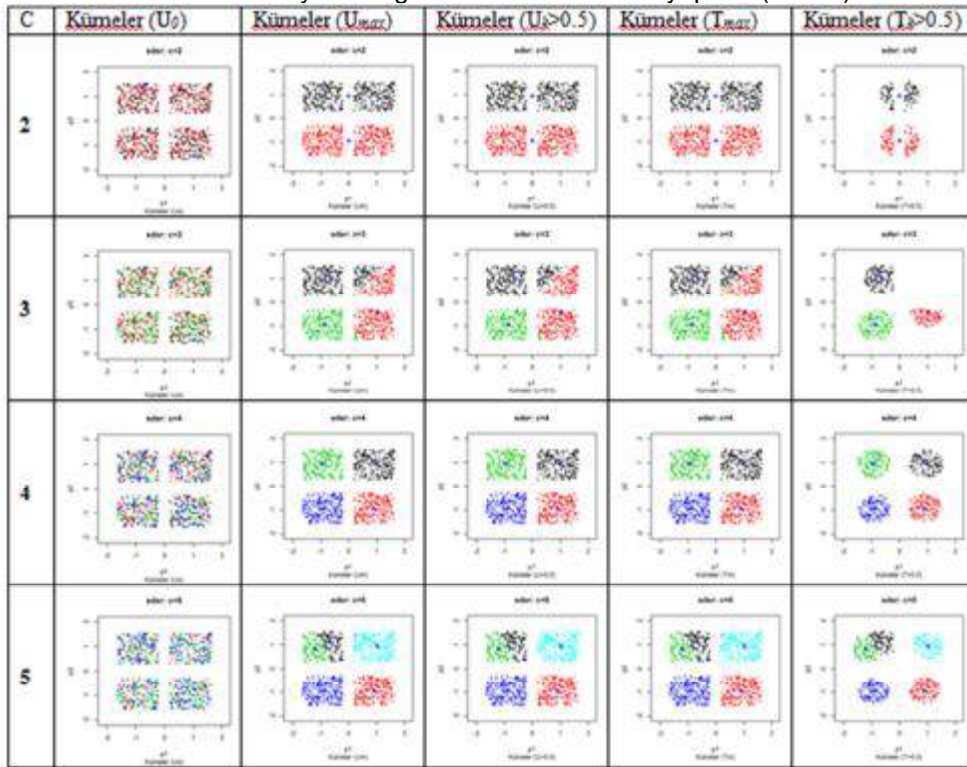
Ek 6.4. UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSEN):



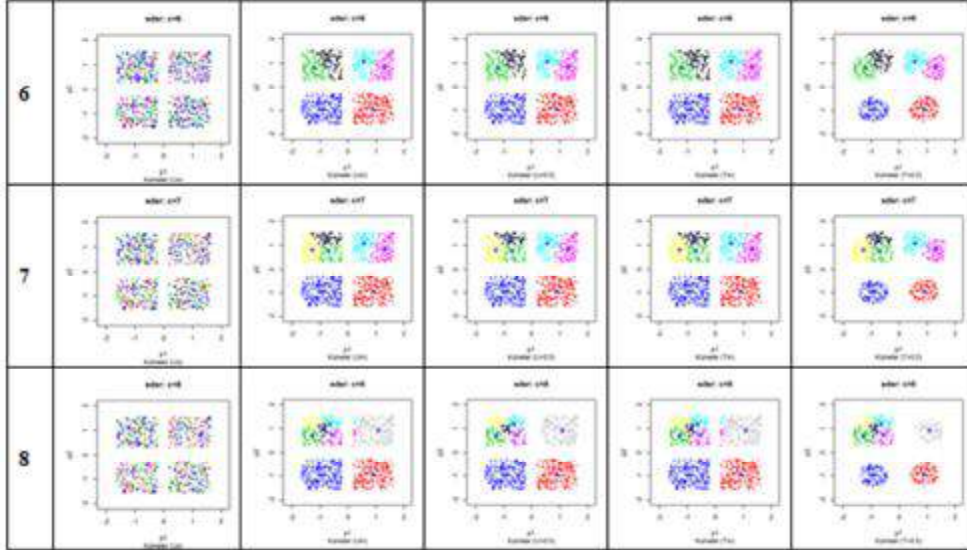
Ek 6.4. (Devamı) UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSEN):



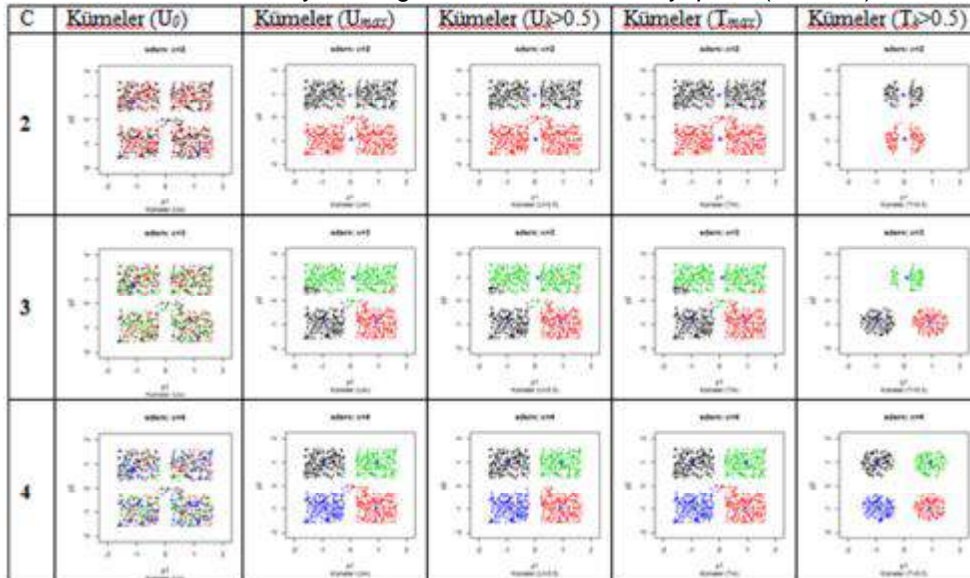
Ek 6.5. UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSR):



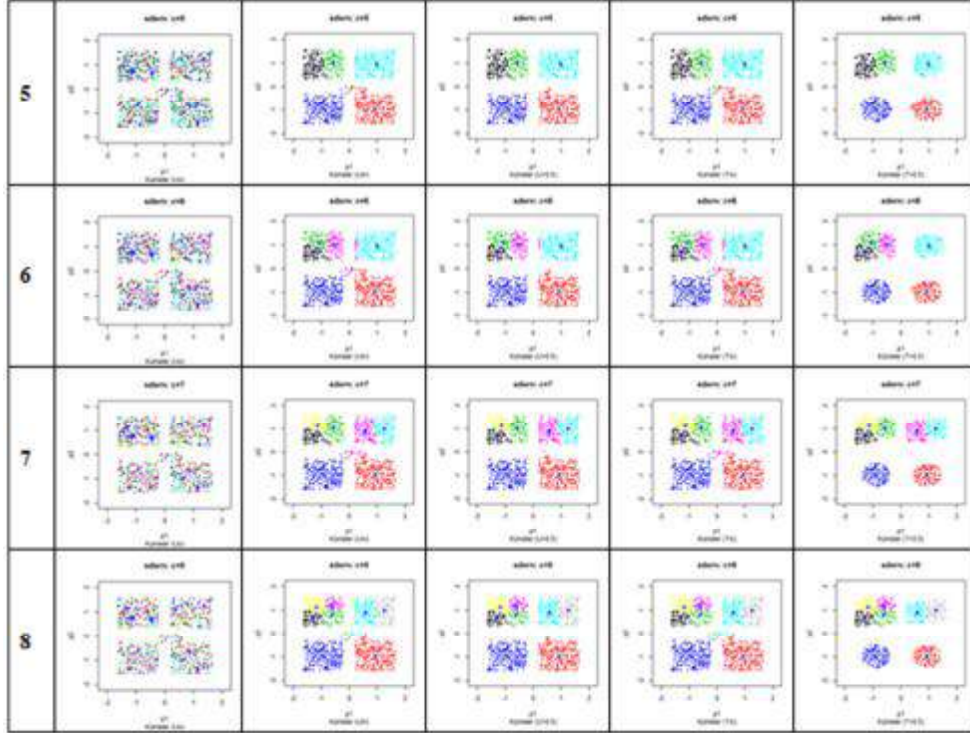
Ek 6.5. (Devamı) UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSR):



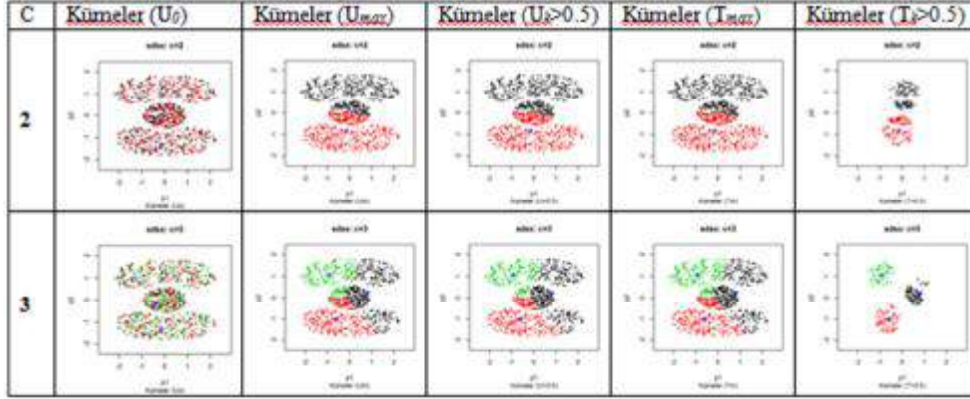
Ek 6.6. UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSRN):



Ek 6.6. (Devamı) UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSRN):



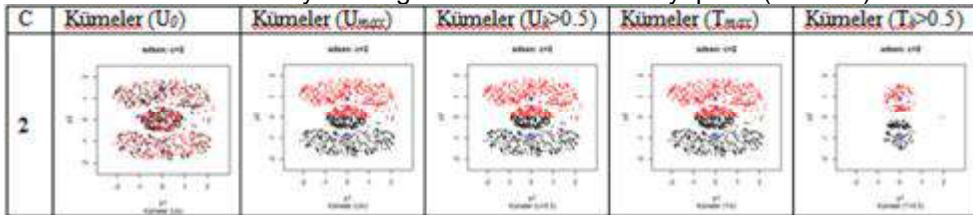
Ek 6.7. UOBKO küme saylarına göre kümeler ve küme yapıları (SDSX):



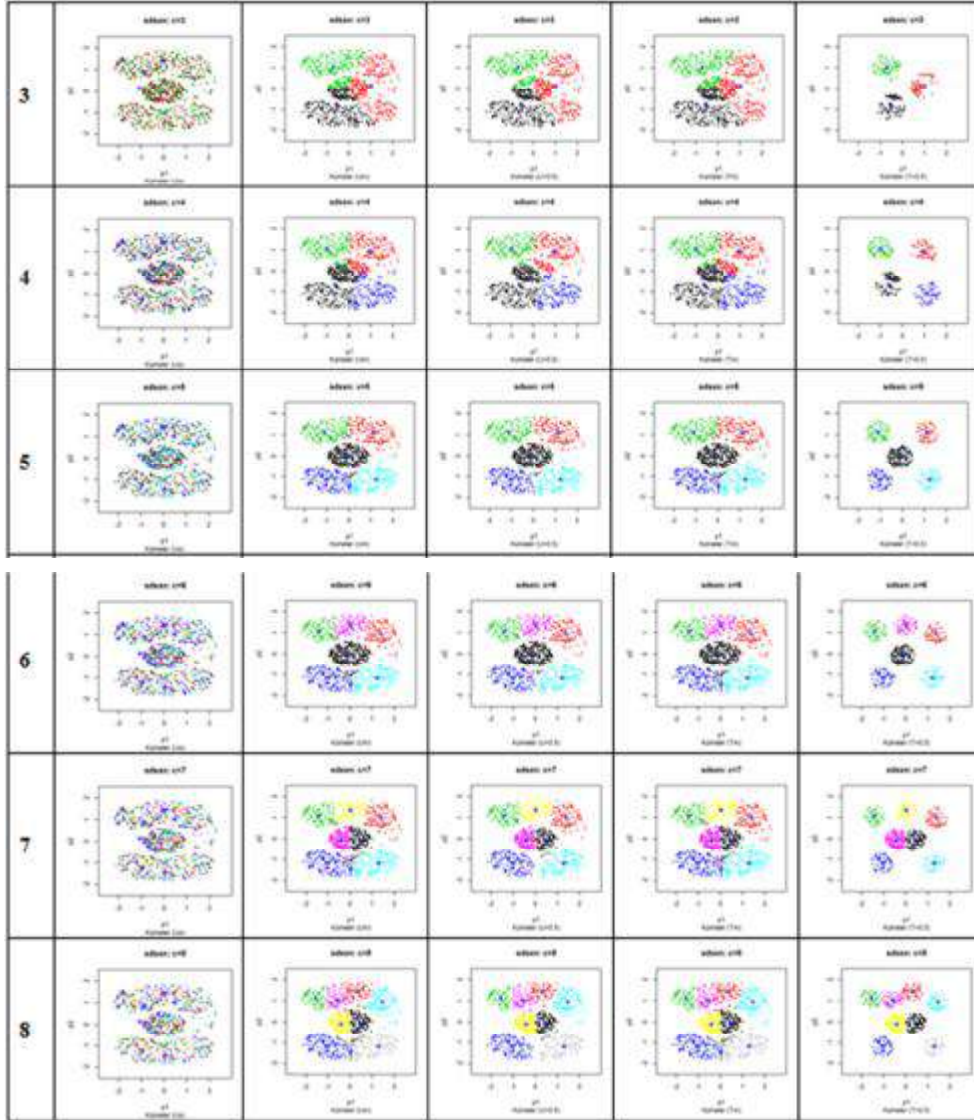
Ek 6.7. (Devamı) UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSX):



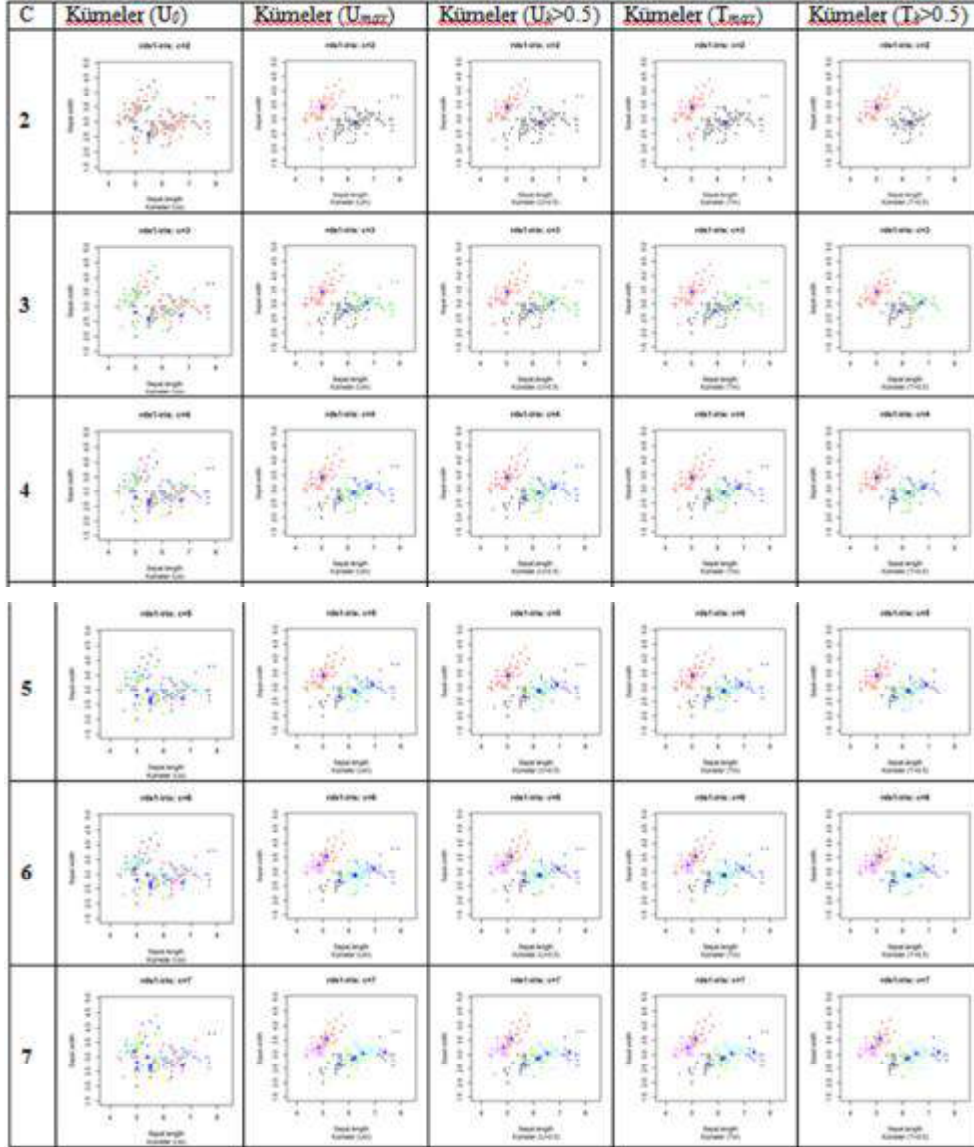
Ek 6.8. UOBCO küme sayılarına göre kümeler ve küme yapıları (SDSXN):



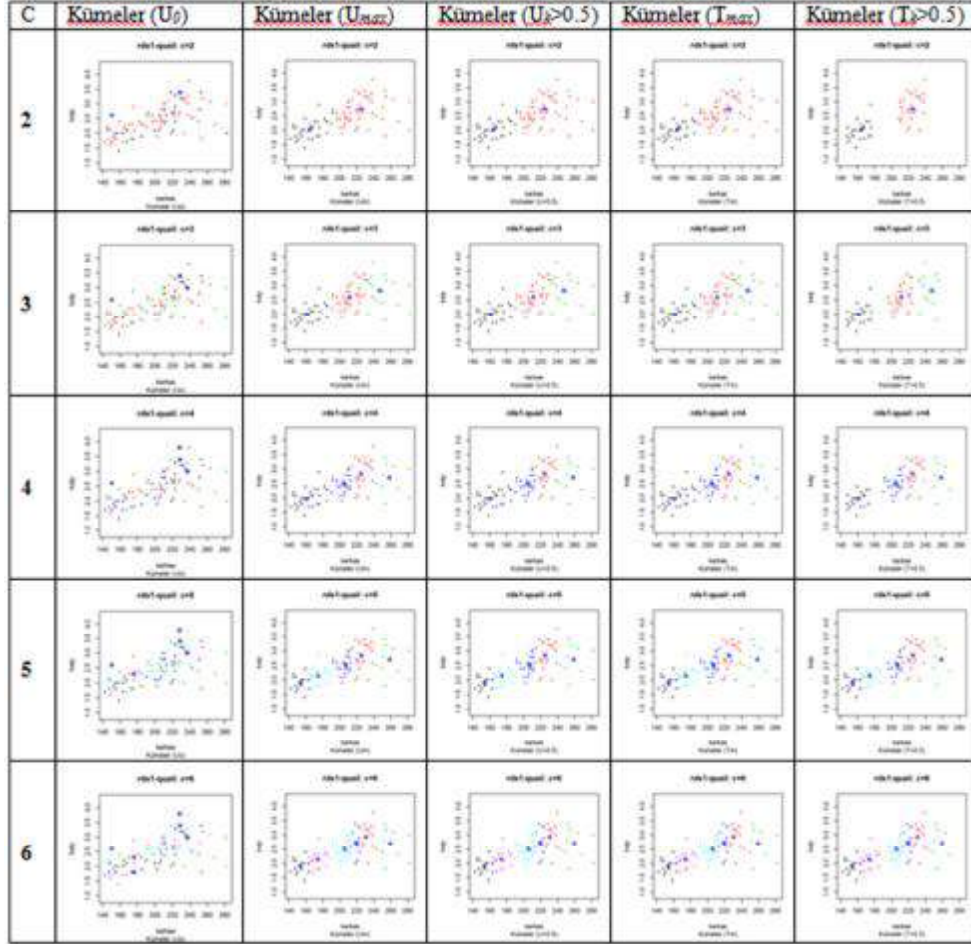
Ek 6.8. (Devamı) UOBKO küme sayılarına göre kümeler ve küme yapıları (SDSXN):



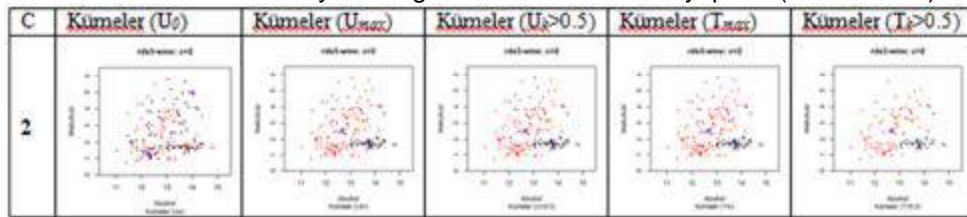
Ek 6.9. UOBOC küme sayılarına göre kümeler ve küme yapıları (RDS1-Iris):



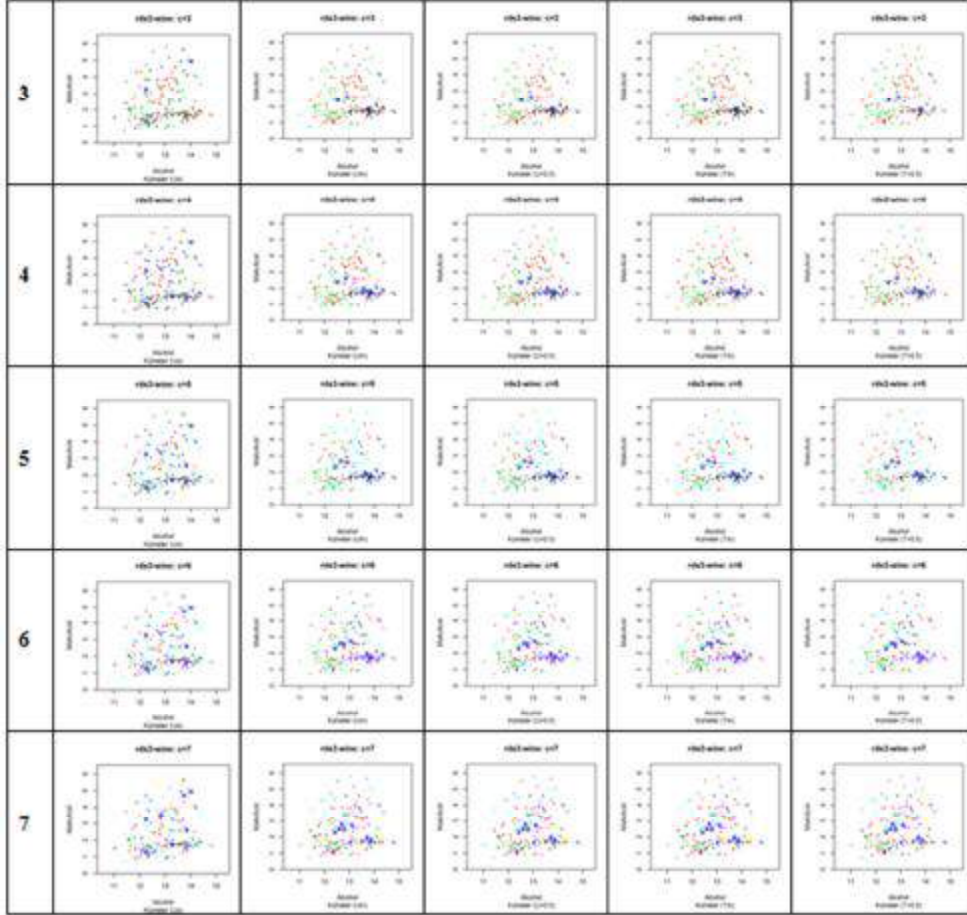
Ek 6.10. UOBKO küme sayılarına göre kümeler ve küme yapıları (RDS2-Quail):



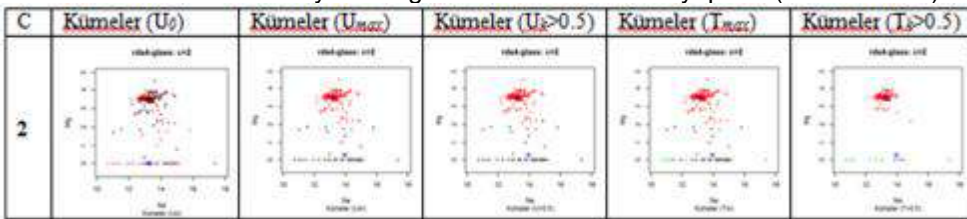
Ek 6.11. UOBKO küme sayılarına göre kümeler ve küme yapıları (RDS3-Wine):



Ek 6.11. (Devamı) UOBCO küme sayılarına göre kümeler ve küme yapıları (RDS3-Wine):



Ek 6.12. UOBCO küme sayılarına göre kümeler ve küme yapıları (RDS4-Glass):



Ek 6.12. (Devamı) UOBCO küme sayılarına göre kümeler ve küme yapıları (RDS4-Glass):

