



**SINIFLANDIRMA PROBLEMLERİNDE ÖZELLİK SEÇİMİ İÇİN
GELİŞTİRİLEN MODİFİYE EDİLMİŞ PARÇACIK SÜRÜ
OPTİMİZASYONU ALGORİTMASI**

Emre KOÇAK

**DOKTORA TEZİ
İSTATİSTİK ANA BİLİM DALI**

**GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

TEMMUZ 2024

ETİK BEYAN

Gazi Üniversitesi Fen Bilimleri Enstitüsü Tez Yazım Kurallarına uygun olarak hazırladığım bu tez çalışmada;

- Tez içinde sunduğum verileri, bilgileri ve dokümanları akademik ve etik kurallar çerçevesinde elde ettiğimi,
- Tüm bilgi, belge, değerlendirme ve sonuçları bilimsel etik ve ahlak kurallarına uygun olarak sunduğumu,
- Tez çalışmada yararlandığım eserlerin tümüne uygun atıfta bulunarak kaynak gösterdiğimi,
- Kullanılan verilerde herhangi bir değişiklik yapmadığımı,
- Bu tezde sunduğum çalışmanın özgün olduğunu,

bildirir, aksi bir durumda aleyhime doğabilecek tüm hak kayıplarını kabullendiğimi beyan ederim.

Emre KOÇAK

10/07/2024

SINIFLANDIRMA PROBLEMLERİNDE ÖZELLİK SEÇİMİ İÇİN GELİŞTİRİLEN MODİFİYE EDİLMİŞ PARÇACIK SÜRÜ OPTİMİZASYONU ALGORİTMASI

(Doktora Tezi)

Emre KOÇAK

GAZİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

Temmuz 2024

ÖZET

Teknolojinin gelişmesi ile hemen hemen çalışma alanlarının tamamında büyük miktarda veri toplanmaya başlanmış ve bu durum büyük boyutlu veri kümelerinin hızla büyümesine neden olmuştur. Bu veri kümelerinde yer alan alakasız ve gereksiz birçok özellik, veri madenciliği çalışmalarında kullanılan öğrenme algoritmalarının performansında ciddi sorunlara yol açmıştır. Bu sorunların giderilmesi amacıyla çeşitli özellik seçimi yaklaşımları önerilmiştir. Özellik seçimi üzerine yapılan çalışmalar, maksimum doğrulukta bir öğrenme modeli oluşturmak için bir dizi optimal özellik alt kümesi belirlenmesinin karmaşık bir sorun olduğunu göstermiştir. Bu tür karmaşık ve zorlu optimizasyon problemlerinin üstesinden gelebilmek için çeşitli meta-sezgisel yöntemler önerilmiştir. Bu tez çalışmasında ele alınan ve meta-sezgisel yöntemlerden olan İkili Parçacık Sürü Optimizasyonu (BPSO) algoritması, optimal özellik alt kümesinin belirlenmesinde önemli bir rol oynamaktadır. BPSO basit bir yapıya sahip olmasına rağmen araştırma özelliği zayıf olduğundan yerel optimuma sıkışabilir. Bu durum ise özellik seçme problemlerinde farklı özelliklerin seçimini engelleyerek sınıflandırma yönteminin performansını etkileyebilir. Literatürde bu sorunun giderilmesi amacıyla BPSO algoritmasının parametrelerini konu alan çalışmalar mevcut olmasına rağmen bu parametrelerin birlikte değerlendirildiği çalışmalar henüz yapılmamıştır. Bu nedenle bu tez çalışmasında önemli BPSO algoritma parametrelerinin özellik seçimindeki etkilerini ve birbirleriyle olan etkileşimlerini inceleyebilmek amacıyla faktöriyel tasarım modeli oluşturulmuştur. Elde edilen sonuçlar, BPSO algoritmasında farklı parametre değerlerinin kullanılması yerine BPSO algoritmasının çalışma mekanizmasının güncellenmesi gerektiğini göstermiştir. Bu doğrultuda, yerel optimuma takılmayı önlemek, yavaş yakınsamanın önüne geçmek ve BPSO algoritmasının performansını geliştirmek için bu tez çalışmasında yeni bir modifiye edilmiş BPSO (MBPSO) algoritması önerilmiştir. MBPSO algoritması ile farklı özelliklerin seçilebilme olasılığı artırılarak algoritmanın yerel optimuma takılması önlenmiş ve model performansında iyileşmeler meydana gelmiştir. Analizler sonucunda MBPSO algoritması, BPSO algoritmasına göre daha iyi bir yakınsama göstermiş ve daha az özellik ile daha yüksek doğru sınıflandırma oranı sağlanmıştır.

Bilim Kodu : 20517

Anahtar Kelimeler : Özellik seçimi, meta-sezgisel yöntemler, ikili parçacık sürü optimizasyonu, faktöriyel tasarım, sınıflandırma

Sayfa Adedi : 75

Danışman : Prof. Dr. H. Hasan ÖRKÜ

MODIFIED PARTICLE SWARM OPTIMIZATION ALGORITHM DEVELOPED FOR
FEATURE SELECTION IN CLASSIFICATION PROBLEMS

(Ph. D. Thesis)

Emre KOÇAK

GAZİ UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

July 2024

ABSTRACT

With the development of technology, large amounts of data have begun to be collected in almost all fields of study, resulting in the rapid growth of large-scale data sets. Many irrelevant and unnecessary features in these data sets have caused severe problems in the performance of learning algorithms used in data mining studies. Various feature selection approaches have been proposed to address these problems. Studies on feature selection have shown that determining a set of optimal feature subsets to create a learning model with maximum accuracy is a complex problem. Various meta-heuristic methods have been proposed to tackle such complex and challenging optimization problems. The Binary Particle Swarm Optimization (BPSO) algorithm, one of the meta-heuristic methods discussed in this thesis, is essential in determining the optimal feature subset. Although BPSO has a simple structure, its search feature is weak, so that it can get stuck in local optima. This situation can affect the performance of the classification method by preventing the selection of different features in feature selection problems. Although there are studies in the literature on BPSO parameters to solve this problem, studies have yet to be conducted in which these parameters are evaluated together. Therefore, a factorial design model was created in this thesis study to examine the effects and interactions of important BPSO parameters on feature selection. The results showed that the working mechanism of the BPSO algorithm should be updated rather than the parameters of the BPSO algorithm. In this regard, a new modified BPSO (MBPSO) algorithm is proposed in this thesis study to prevent getting stuck in local optimum, prevent slow convergence, and improve the performance of the BPSO algorithm. By increasing the possibility of selecting different features with the MBPSO algorithm, the algorithm was prevented from getting stuck in local optima, and improvements in model performance occurred. As a result of the analysis, the MBPSO algorithm showed better convergence than the BPSO algorithm and had a higher correct classification rate with fewer features.

Science Code : 20517

Key Words : Feature selection, meta-heuristics, binary particle swarm optimization, factorial design, classification

Page Number : 75

Supervisor : Prof. Dr. H. Hasan ÖRKÇÜ

TEŞEKKÜR

Çalışmalarım boyunca beni yönlendiren, ilgi ve önerilerini paylaşmaktan kaçınmayan değerli danışmanım Prof. Dr. H. Hasan ÖRKÜ'ye, tez çalışması boyunca gösterdikleri ilgi ve sağladıkları değerli katkıları için Prof. Dr. Bülent ÇELİK ve Prof. Dr. Özlem TÜRKŞEN'e, tez savunması sırasında yaptıkları öneriler ile tezimin daha yararlı olmasını sağlayan Prof. Dr. Reşat KASAP ve Prof. Dr. Diyar AKAY'a, Gazi Üniversitesi İstatistik Bölümümün değerli öğretim üyelerine, desteği ve yardımlarıyla her zaman yanımda olan değerli çalışma arkadaşlarıma teşekkür ederim. Çalışmalarım boyunca destekleriyle beni hiçbir zaman yalnız bırakmayan en değerli varlığım olan eşim ve oğluma sonsuz teşekkür ederim.

İÇİNDEKİLER

	Sayfa
ÖZET	iv
ABSTRACT.....	v
TEŞEKKÜR.....	vi
İÇİNDEKİLER	vii
ÇİZELGELERİN LİSTESİ.....	ix
ŞEKİLLERİN LİSTESİ.....	x
SİMGELER VE KISALTMALAR.....	xi
1. GİRİŞ.....	1
2. ÖZELLİK SEÇİM YÖNTEMLERİ	7
3. PARÇACIK SÜRÜ OPTİMİZASYONU.....	13
3.1. İkili Parçacık Sürü Optimizasyonu.....	16
3.1.1. Kaotik haritalar.....	19
3.1.2. Transfer fonksiyonları.....	21
3.2. Modifiye Edilmiş İkili Parçacık Sürü Optimizasyonu	23
4. FAKTÖRİYEL TASARIM.....	29
5. ÖZELLİK SEÇİM YÖNTEMLERİNİN DEĞERLENDİRİLMESİ	33
5.1. Sınıflandırma Algoritması	33
5.2. Ayrıştırma Türü	34
5.3. Karmaşıklık Matrisi.....	34
6. DENEYSEL SONUÇLAR.....	37
6.1. Problemin Tanımı	37
6.2. Uygulama	38
7. SONUÇ VE ÖNERİLER.....	59

	Sayfa
KAYNAKLAR	63
EKLER.....	69
EK-1. Farklı faktör seviyeleri için ACC sonuçları ve seçilen özelliklerin sayısı	70
EK-2. $(\beta; \gamma) = (0,9; 0,1)$ durumu için MBPSO algoritma sonuçları	71
EK-3. $(\beta; \gamma) = (0,5; 0,5)$ durumu için MBPSO algoritma sonuçları	72
EK-4. $(\beta; \gamma) = (0,1; 0,9)$ durumu için MBPSO algoritma sonuçları	73
ÖZGEÇMİŞ	74

ÇİZELGELERİN LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Eylemsizlik ağırlığı türleri ve formülleri	16
Çizelge 3.2. Kaotik harita türleri ve matematiksel formülleri	20
Çizelge 3.3. Transfer fonksiyon türleri ve matematiksel formülleri.....	22
Çizelge 3.4. Sınırlı fonksiyon türleri ve matematiksel formülleri	26
Çizelge 4.1. Varyans analizi tablosu.....	30
Çizelge 5.1. Karmaşıklık matrisi	34
Çizelge 5.2. Performans ölçüsü türleri ve matematiksel formülleri	35
Çizelge 6.1. Faktör ve düzeyleri	39
Çizelge 6.2. 4 ⁴ faktöriyel tasarım sonuçları	40
Çizelge 6.3. BPSO ve MBPSO algoritmalarına ilişkin karşılaştırmalı sonuçlar	44
Çizelge 6.4. BPSO ve MBPSO algoritmalarına ilişkin farklı parametreler açısından karşılaştırmalı sonuçlar	46
Çizelge 6.5. Amaç fonksiyonu ağırlıkları açısından ortalama ACC, ortalama seçilen özellik sayısı ve test istatistiği	47
Çizelge 6.6. Eşik değeri açısından ortalama ACC, ortalama seçilen özellik sayısı ve test istatistiği	48
Çizelge 6.7. Sınırlı fonksiyon türü açısından ortalama ACC, ortalama seçilen özellik sayısı ve test istatistiği	51
Çizelge 6.8. Eylemsizlik ağırlığı türü açısından ortalama ACC, ortalama seçilen özellik sayısı ve test istatistiği	52
Çizelge 6.9. Parçacık sayısı açısından ortalama ACC, ortalama seçilen özellik sayısı ve test istatistiği	54
Çizelge 6.10. Veri setlerinin yapılarına ilişkin bilgiler.....	55
Çizelge 6.11. Farklı performans ölçülerine göre lineer azalan eylemsizlik türü için BPSO ve MBPSO sonuçlarının karşılaştırılması.....	56
Çizelge 6.12. Farklı performans ölçülerine göre lojistik eylemsizlik türü için BPSO ve MBPSO sonuçlarının karşılaştırılması.....	57

ŞEKİLLERİN LİSTESİ

Şekil	Sayfa
Şekil 2.1. Özellik seçim yöntemleri	10
Şekil 3.1. BPSO akış diyagramı.....	18
Şekil 3.2. Kaotik haritaların görselleştirilmesi	21
Şekil 3.3. Transfer fonksiyonlarının görselleştirilmesi.....	23
Şekil 3.4. BPSO algoritması için amaç fonksiyon değerlerinin (pbest) değişimi.....	24
Şekil 3.5. BPSO algoritması için seçilen özellik sayılarının değişimi.....	25
Şekil 3.6. Sınırlı fonksiyonların görselleştirilmesi	27
Şekil 3.7. MBPSO akış diyagramı	28
Şekil 6.1. Eşik değeri ve transfer fonksiyonu açısından ACC ve seçilen özellik sayısı.	41
Şekil 6.2. BPSO ve MBPSO algoritmalarına ilişkin amaç fonksiyon değeri (pbest) değişimlerinin karşılaştırılması.....	43
Şekil 6.3. BPSO ve MBPSO algoritmalarına ilişkin seçilen özellik sayısı değişimlerinin karşılaştırılması.....	43
Şekil 6.4. Algoritmalar ve eşik değeri açısından seçilen özellik sayısı	45
Şekil 6.5. Algoritmalar ve eşik değeri açısından ACC değerleri.....	45
Şekil 6.6. Amaç fonksiyonu ağırlıkları açısından ACC ve seçilen özellik sayısı.....	47
Şekil 6.7. Eşik değeri açısından ACC ve seçilen özellik sayısı	49
Şekil 6.8. Eşik değeri açısından ortalama seçilen özellik sayısı	49
Şekil 6.9. Eşik değerini açısından ACC değerleri.....	50
Şekil 6.10. Sınırlı fonksiyon türü açısından ACC ve seçilen özellik sayısı.....	51
Şekil 6.11. Eylemsizlik ağırlığı türü açısından ACC ve seçilen özellik sayısı.....	53
Şekil 6.12. Parçacık sayısı açısından ACC ve seçilen özellik sayısı	55

SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklamalar
ACC	Doğru sınıflandırma oranı
BPSO	İkili parçacık sürü optimizasyonu
ER	Hata oranı
FN	Yanlış negatiflerin sayısı
FP	Yanlış pozitiflerin sayısı
F1	F1 ölçüsü
KO	Kareler ortalaması
KT	Kareler toplamı
MBPSO	Modifiye edilmiş ikili parçacık sürü optimizasyonu
MCC	Matthews korelasyon katsayısı
PSO	Parçacık sürü optimizasyonu
SF	Seçilen özellik sayısı
SS	Standart sapma
SVM	Destek vektör makineleri
TF	Toplam özellik sayısı
TN	Doğru negatiflerin sayısı
TNR	Özgüllük ölçüsü
TP	Doğru pozitiflerin sayısı
TPR	Duyarlılık ölçüsü

1. GİRİŞ

Günümüzde, bilgisayar ve veri tabanı teknolojilerindeki hızlı büyüme büyük boyutlu veri kümelerinin de hızla büyümesine yol açmıştır. Buna bağlı olarak, yüksek hız ve doğruluk gerektiren büyük boyutlu veri setlerine sahip veri madenciliği uygulamaları da hızla artmaktadır. Veri boyutlarındaki bu artış sistemin hesaplama maliyetini artırmakta ve modellerde yer alan alakasız ve gereksiz birçok özellik modellerin doğruluk oranının düşmesine neden olmaktadır. Ayrıca özellik sayısı arttıkça daha fazla veri örneğine olan ihtiyaç da artmakta ve bu durum problemin zamansal ve mekansal zorluklarını artırmaktadır.

Verilerin modellenmesi ya da sınıflandırılmasının önemli hedeflerinden biri, veri kümelerindeki eğitim verilerine ve mevcut özelliklere dayanarak tahminde bulunmaktır. Veri kümelerinde çok sayıda alakasız ve gereksiz özelliğin var olması, öğrenilen modellerin doğruluğunu önemli ölçüde etkileyeceği gibi modellerin öğrenme hızını da düşürecektir. Bu nedenle bu türlü büyük boyutlara sahip verilerin mevcut olması, öğrenme algoritmalarının performansında ciddi sorunlara neden olabilmektedir.

Büyük boyutlu veri kümelerinde yer alan ve verinin içinde yatan bilgiyi bulmak açısından önemli olmayan bir takım alakasız ve gereksiz özellikler modelleme yapılmadan önce veri kümelerinden çıkartılabilir. Bu tür özelliklerin veri kümesinden çıkarılması işlemine literatürde boyut indirgeme ya da boyut azaltma adı verilmiştir. Boyut indirgeme işlemi, makine öğrenimi algoritmasının performansını artırabileceği gibi alakasız ve gereksiz özellikleri kaldırarak hesaplama zorluğunu azaltabilir [1, 2]. Bu nedenle birçok çalışma alanında veri boyutunun indirgenmesi önemli konuların başında gelmektedir.

Yıllar içerisinde yapılan çalışmalar incelendiğinde, boyut indirgeme için özellik çıkarımı ve özellik seçimi olmak üzere iki farklı yaklaşım sunulmuştur. Özellik çıkarımında, başlangıç veri kümesindeki özelliklerin doğrusal veya doğrusal olmayan bir dönüşüm ile birleştirilmesi sonucunda özellik alt kümesi oluşturularak bu özelliklerin birincil özelliklerde yer alan bilgilerin tamamını ya da birçoğunu içermesi sağlanır. Buna karşın özellik seçiminde, alakasız ve gereksiz özellikler çıkarılarak, modelleme için yeterli bilgiyi içinde barındıran ve başlangıç veri kümesindeki özelliklere sahip bir alt kümesi seçilir.

Özellik seçimi, veri madenciliği ve örüntü tanıma istatistik alanı açısından aktif bir araştırma alanı olmuştur. Özellik seçiminin ana fikri, hiç öngörü bilgisi içermeyen ya da çok az içeren özellikleri ve aynı zamanda birbirleriyle güçlü bir şekilde ilişkili olan gereksiz özellikleri eleyerek mevcut özelliklerin bir alt kümesini seçmektir [3]. Öğrenme modelleri ile ilişkisi bakımından filtre, sarmal ve gömülü olmak üzere üç ana başlıkta toplanan özellik seçim yöntemlerine ek olarak hem filtre hem de sarmal modellerin birlikte oluşturduğu hibrit yöntemlerde dahil edilebilir.

Filtre yönteminde özelliklerin ilişkisini hesaplamak için verideki bilgi içeriğinin ve entropinin ölçülmesi sağlayan bilgi teorisi ölçümleri kullanılarak değerlendirilip sıralanır ve ardından en yüksek sıralamaya sahip olan özellikler seçilir [4]. Ayrıca bu yöntem özellik seçim sürecini öğrenme algoritmalarından bağımsız olarak gerçekleştirdiğinden dolayı diğer yöntemlere göre daha hızlı çalışabilmektedir. Sarmal yöntemler, seçilen özellik alt kümesinin verimliliğini ölçmek için bir sınıflandırma algoritması kullanırlar. En yüksek sınıflandırma doğruluğuna sahip bir özellik alt kümesini seçmek amacıyla arama sürecinde bir özellik alt kümesini değerlendirmek için belirli bir öğrenme modeli kullanılır. Bu durum daha iyi bir özellik alt kümesi seçilebilmesine olanak sağlamasına karşın çalışma süreleri açısından uzun zaman alabilir. Gömülü yöntemde ise özellik seçme süreci, öğrenme algoritmasının bir parçası olarak kabul edilir ve eş zamanlı olarak gerçekleşir [5]. Hibrit yöntemlerde hem filtre hem de sarmal modellerden yararlanılarak, filtre modellerinin hesaplama verimliliği ile sarmal modellerin doğruluğunu dengeleyen bir model sağlanmaya çalışılır.

Orijinal özellik sayısı n olmak üzere, en alakalı ve önemli özellikleri bulmak için tüm olası alt kümeler dahil toplam 2^n kadar özellik alt kümesi oluşturulması gerekmektedir. Uygun özelliklerin bulunmasını sağlayan bu şekildeki kapsamlı bir arama orta büyüklükteki veri kümeleri için bile hesaplama açısından mümkün değildir. Tüm olası alt kümelerin değerlendirilmesi çok zaman alacağından dolayı hem kalite açısından faydalı hem de sayısal olarak uygulanabilir bir çözüm aranmalıdır [6, 7].

Çözülmesi ve doğrulanması zor optimizasyon problemleri çözmeye yönelik olan meta-sezgisel algoritmalar, optimal çözümü bulmak yerine kabul edilebilir bir sürede tatmin edici yaklaşık çözümler bulabilen yaklaşımlardır [8]. Bu algoritmalar, yerel optimumlardan kaçış

stratejilerine sahip olan ve çok çeşitli optimizasyon problemlerinde kullanılabilen optimizasyon algoritmalarının kategorilerinden biridir.

Birçok özellik seçme yöntemi, büyük boyutlu veri setinde artan hesaplama zorluğunu önlemek için meta-sezgisel yöntemler kullanır. Bu algoritmalar, bir optimizasyon problemini çözmek ve bir dizi yinelemede en uygun çözümü aramak için farklı mekanizmaları ve işlemleri kullanır. Genellikle rastgele çözümler içeren bir popülasyonla başlayan bu algoritmalar, her yineleme adımında bu çözümlerin optimalliğini artırmaya çalışır. Meta-sezgisel algoritmaların çoğunun başlangıcında, bir dizi başlangıç çözümü rastgele üretilir ve daha sonra oluşturulan popülasyonun bireysel çözümlerinin optimalliğini hesaplamak için bir uygunluk fonksiyonu kullanılır. Sonlandırma kriterlerinden hiçbiri karşılanmazsa yeni nesil üretime başlanır. Bu döngü, sonlandırma kriterlerinden biri karşılanıncaya kadar tekrarlanır [9, 10].

Parçacık Sürü Optimizasyonu (PSO), Kennedy ve Eberhart tarafından 1995 yılında geliştirilen güçlü bir sürü zekası tabanlı optimizasyon yöntemidir [11]. Kuşların ve balıkların kolektif davranışlarından ilham alan bu optimizasyon yöntemi, son özellik alt kümesinin seçimini optimize etmek için araştırmalarda sıklıkla kullanılmaktadır.

Chuang ve diğerleri, arama sürecinde herhangi bir gelişme olmadığında arama alanına yeni parçacıkların dahil edilmesine dayalı olarak ikili PSO'nun (BPSO) performansını artırmak için bir yayın balığı stratejisi uyguladı [12]. Xue ve diğerleri, özellik alt kümesi seçiminde optimizasyon görevlerinin performansını artırmak için yeni başlatma yöntemleri ve en iyi parçacık güncelleme stratejileri geliştirdi [13]. Banka ve Dara, sınıflandırma görevinde belirlenen son özellik alt kümesini seçmek için BPSO algoritması ve Hamming mesafesinin entegrasyon modelini önermiştir. Bu yöntemde, bir BPSO algoritmasında hız güncelleme sürecini iyileştirmek için bir hamming mesafesi kullanılır ve daha sonra bu geliştirilmiş BPSO, özellik seçimi problemlerinde kullanılır [14]. Moradi ve Gholampour, filtre ve sarmalayıcı yaklaşımların birleşimi ile etkili PSO tabanlı özellik seçim yöntemi önermiştir. Önerilen yöntem, ilgili olmayan ve ilgili özelliklerin alt kümesini seçmek için yeni bir yerel arama başlatır [15]. PSO'da durgunluğu önlemek ve erken yakınsamanın üstesinden gelmek için Mistry ve diğerleri, yeni bir popülasyon yapısı kullanarak küçük popülasyonlu ikincil sürü stratejisini kullandı [16].

PSO algoritmasında yer alan eylemsizlik ağırlığı, küresel arama ve yerel arama yeteneği arasındaki dengeyi kontrol eder. Bu nedenle eylemsizlik ağırlığının uygun şekilde ayarlanması önemlidir. PSO algoritmasının arama süreci sıklıkla yerel bir optimumda parçacık sıkışmasıyla karşılaşır ve bu durum ise erken yakınsamaya neden olur. Erken yakınsamanın önlenmesi ve bu iki arama arasındaki dengenin uygun bir şekilde sağlanabilmesi için kaotik haritalar, zamanla değişen eylemsizlik ağırlıkları, sabit ve rastgele eylemsizlik ağırlıkları dahil olmak üzere zaman içinde çeşitli süreçler önerilmiştir [17–20].

Sürekli arama alanı, BPSO algoritmasının özelliklerine göre ayrık ikili uzaya dönüştürülür. Amaç, konum vektörünün elemanının 0'dan 1'e gitme olasılığını temsil etmek olduğundan transfer fonksiyonu $[0,1]$ aralığıyla sınırlandırılmalıdır. Orijinal BPSO algoritmasının, Sigmoid transfer fonksiyonunun sınırlamaları nedeniyle keşif ve kullanım arasında iyi bir denge sağlayamadığı iddia edilmiştir [21–24]. Bunun üstesinden gelmek için literatürde V şekilli, U şekilli, Z şekilli ve Q şekilli gibi farklı transfer fonksiyonlarının yanı sıra zamana bağlı transfer fonksiyonları BPSO algoritması ile kullanılmıştır [21, 25–28].

Orijinal BPSO algoritmasında Sigmoid fonksiyonu yardımıyla $[0,1]$ aralığına dönüştürülen parçacık değeri, ikili çözüme dönüştürüldüğü ikilileştirme kuralını kullanmaktadır. Bu kurala göre dönüştürülen parçacık değeri yineleme anında üretilen herhangi bir rastgele sayıdan küçükse 0, büyükse 1 değerini almaktadır. Yapılan çalışmaların büyük bir çoğunluğunda orijinal BPSO'da kullanılan ikilileştirme kuralı kullanılmasına rağmen, statik olasılık, elitist, elitist rulet ve tamamlayıcı gibi farklı kurallar da BPSO algoritması ile kullanılmıştır [29–33].

Yerel optimuma takılmayı önlemek ve yavaş yakınsamanın önüne geçilebilmesi için literatürde farklı yaklaşımlar önerilmiştir. Buna rağmen, bilindiği kadarıyla önerilen bu yaklaşımların BPSO algoritması üzerindeki etkileri ve birbirleri ile olan etkileşimlerinin istatistiksel olarak incelendiği çalışma bulunmamaktadır. Bu tez çalışmasında, diğer çalışmalardan farklı olarak önemli BPSO algoritma parametrelerinin özellik seçimindeki etkilerini ve birbirleriyle olan etkileşimlerini inceleyebilmek amacıyla faktöriyel tasarım modeli oluşturulmuştur. Elde edilen sonuçlar doğrultusunda, yerel optimuma takılmayı önlemek, yavaş yakınsamanın önüne geçmek ve BPSO algoritmasının performansının geliştirilebilmesi için BPSO algoritmasında farklı parametre değerlerinin kullanılması yerine BPSO algoritma yapısında bir değişiklik meydana gelmesi gerektiği sonucuna

varılmış olup, yeni bir modifiye edilmiş BPSO (MBPSO) algoritması önerilmiştir. MBPSO algoritması ile farklı özelliklerin seçilebilme olasılığı artırılarak algoritmanın yerel optimuma takılmasının önlenmesi ve model performansında iyileşmeler meydana getirmesi amaçlanmıştır.

Bu tez çalışması yedi bölümden oluşmaktadır. İkinci bölümde, özellik seçim yöntemleri hakkında bilgi verilmiştir. Üçüncü bölümde PSO, BPSO ve bu algoritmalarda kullanılan parametrelere ilişkin bilgiler ile birlikte önerilen MBPSO algoritması detaylı bir şekilde verilmiştir. Dördüncü bölümde faktöriyel tasarım yönteminden bahsedilmiştir. Ardından beşinci bölümde özellik seçim yöntemlerinin değerlendirilmesi ve altıncı bölümde deneysel sonuçlar yer almıştır. Son olarak, yedinci bölümde sonuç ve öneriler bölümü yer almaktadır.



2. ÖZELLİK SEÇİM YÖNTEMLERİ

Son yıllarda bilim ve teknolojinin ilerlemesiyle birlikte veri boyutunun hızla artması veri madenciliği, örüntü tanıma ve makine öğrenimi çalışmalarında optimizasyon sürecinin kalitesini etkileyen en büyük sorunlardan birisi olmuştur. Çok sayıda alakasız ve gereksiz özellik, öğrenilen modellerin doğruluğunu önemli ölçüde bozabileceği ve modellerin öğrenme hızını da düşürebileceği için bu türlü büyük boyutlara sahip veriler öğrenme algoritmalarının performansında ciddi sorunlara neden olabilmektedir. Literatürde bu soruna boyutsallık laneti adı verilmiştir ve bu sorun modellerin hesaplama zamanını önemli bir şekilde artırmaktadır [34].

Boyutsallık, ihtiyaç duyduğu yüksek hesaplama zamanları nedeniyle veri madenciliği ya da makine öğrenimi sürecini etkileyebilecek en büyük sorunlardan biridir. Bir sınıflandırma modelini eğitmek için büyük boyutlu veri kümelerinin kullanılması, öğrenilen modelin performansını azaltabilir. Bu soruna ek olarak, bu tür veri kümeleri yüksek hesaplama zamanı ve bellek maliyetlerine neden olur. Orijinal özellik sayısı n olmak üzere, en alakalı ve önemli özellikleri bulmak için tüm olası alt kümeler dahil toplam 2^n kadar özellik alt kümesi oluşturulması gerekmektedir. Örneğin, bu tez çalışmasında kullanılan “*Ionesphere*” veri seti 34 özellikten meydana gelmektedir. Bu veri setine ait en alakalı ve önemli özellikleri bulmak için $2^{34} = 17\,179\,869\,184$ tane özellik alt kümesi oluşturulması gerekir.

Yüksek hesaplama ve bellek maliyetlerine engel olabilmek amacıyla boyut indirgeme, en bilgilendirici özellikleri kullanarak veri kümelerinin boyutsallığının azaltılması ve aynı zamanda sınıflandırma doğruluğunun en üst düzeye çıkarılması için öğrenme süreci öncesi zorunlu bir aşama haline gelmiştir. Ayrıca, makine öğrenimi algoritmasının performansını artırabilir ve gereksiz özellikleri kaldırarak hesaplama zorluğunu azaltabilir [2, 35]. Bu dezavantajlı durumu gidermek için yıllar içerisinde yapılan çalışmalarda özellik çıkarma ve özellik seçimi olmak üzere iki farklı boyut indirgeme tekniği önerilmiştir.

Özellik çıkarma tekniği, veri depolama ve model performansını artırmak için en bilgilendirici, ayırt edici ve azaltılmış özellikler kümesini bulma sorunuyla ilgilenir. Özellik çıkarma, orijinal özellik kümesinden daha düşük boyutlu yeni bir özellik kümesine ulaşmak için doğrusal veya doğrusal olmayan bir dönüşümü içerir. Mevcut özellikler bir araya

getirilerek daha az sayıda bir özellik alt kümesi oluşturulur ve böylece bu özellik alt kümeleri birincil özelliklerde yer alan bilgilerin birçoğunu ya da tümünü içerir. Temel bileşenler analizi ve diskriminant analizi yaygın olarak bilinen ve en sık kullanılan özellik çıkarma yöntemlerinin başında gelmektedir [36, 37].

Özellik seçiminde diğer özelliklere göre alakasız ve gereksiz özellikler kaldırılarak orijinal bir özellik veri kümesinden bilgilendirici ve ilgili özelliklerden oluşan yeni bir özellik alt kümesini seçmek amaçlanır. Özellik seçiminin arkasındaki ana fikir, öğrenme sürecine büyük ölçüde etkisi olmayan özelliklerin yanı sıra birbirleriyle güçlü bir şekilde ilişkiye sahip olan özellikleri eleyerek, model performansını artıracak önemli özelliklerin bir alt kümesini seçmektir. Özellik sayısındaki bu azalma sonucunda öğrenme süreci hızlandırılır, basit ve anlaşılır bir tahmin modeli elde edilir ve aşırı uyum önlenir [36–39].

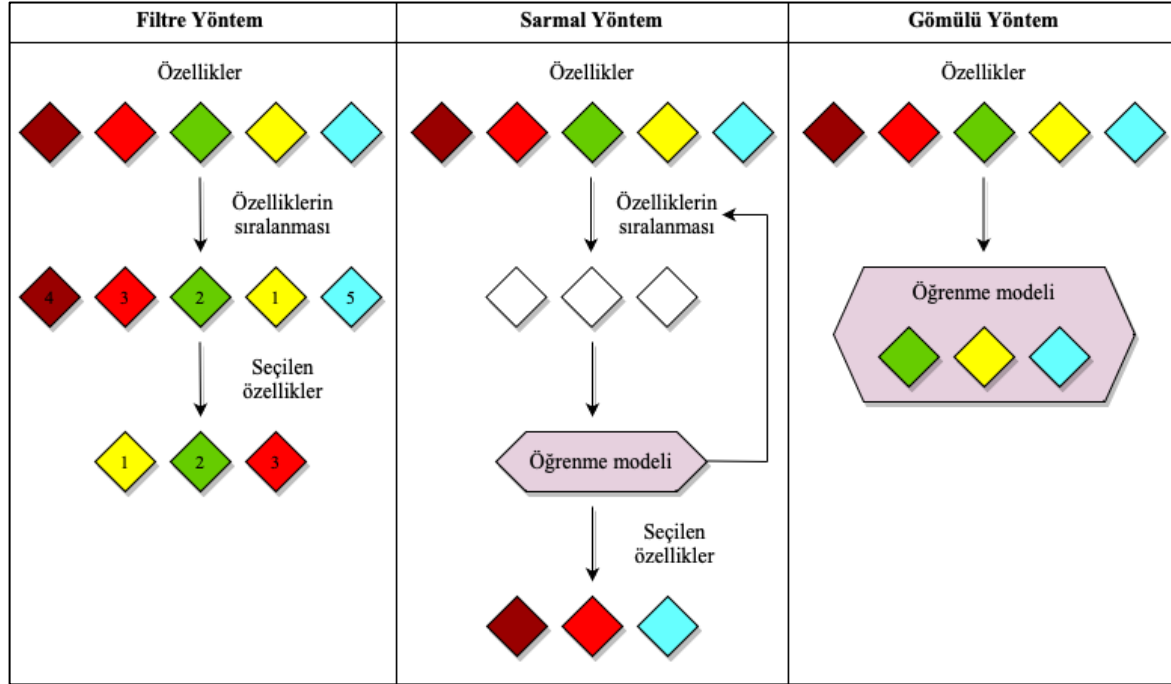
Veri kümelerindeki alakasız ve gereksiz özellikler kaldırılması ve yorumlanabilirliğin artırılabilmesi için bu tez çalışmasında ele alınan özellik seçim sürecinde özellik alt kümenin oluşturulması, özellik alt kümesinin değerlendirilmesi, durdurma kriterleri ve sonuçların doğrulanması şeklinde dört ana aşamadan oluşur. En iyi özellik alt kümesinin belirlenmesi sürecinin her yinelenmesinde, orijinal özelliklerden aday özellik kümesinin bir alt kümesi oluşturulur ve bunun uygunluğu bir değerlendirme kriteri ile ölçülür. Alt küme oluşturma süreci ve değerlendirmesi, önceden belirlenmiş bir durdurma kriterine ulaşılan kadar tekrarlanır. Bu işlem sonunda seçilen özelliğin en iyi alt kümesi test veri seti üzerinde doğrulanır [40, 41].

Öğrenme modelleri ile ilişkisi bakımından filtre, sarmal ve gömülü olmak üzere üç ana başlıkta toplanan özellik seçim yöntemlerine ek olarak hem filtre hem de sarmal modellerin birlikte oluşturduğu hibrit yöntemler de dahil edilebilir. Filtre yönteminde, özelliklerin ilişkisini hesaplamak için Gini İndeks, Laplasiyen Skoru, Karşılıklı Bilgi ve Hilbert-Schmidt Bağımsızlık Kriteri gibi bilgi teorisi ölçümleri kullanılarak değerlendirilip sıralanır ve ardından en yüksek sıralamaya sahip olan özellikler kullanılarak değerlendirilip sıralanır ve ardından en yüksek sıralamaya sahip olan özellikler seçilir [4, 42–45]. Ayrıca bu yöntem özellik seçim sürecini öğrenme algoritmalarından bağımsız olarak gerçekleştirdiğinden dolayı diğer yöntemlere göre daha hızlı çalışabilmektedir. Filtre yöntemi özelliklerin değerlendirilmesi açısından incelendiğinde, tek değişkenli ve çok değişkenli olmak üzere iki kategoriye ayrılır. Tek değişkenli yöntemlerde, özellikler birbirinden bağımsız olduğu ve

özellikler arasındaki olası bağımlılıkların dikkate alınmadığı varsayılır. Fakat özellikler arasındaki bağımlılığı göz ardı ederek, nihai özellik setinde benzer özelliklerin sunulmasına ve aynı zamanda zayıf ve karmaşık öğrenme modellerinin oluşumuna neden olmasından dolayı çok değişkenli filtre yöntemler önerilmiştir [36]. Çok değişkenli yöntemlerde, özelliklerin uygunluğu karşılıklı bağımlılıklarına göre değerlendirilir. Bundan dolayı çok değişkenli yöntemler tek değişkenli yöntemlerden daha fazla hesaplama zorluğuna sahip olmasına rağmen daha yüksek performansa sahiptirler.

Sarmal yöntemler, seçilen özellik alt kümesinin verimliliğini ölçmek için bir sınıflandırma algoritması kullanırlar. En yüksek sınıflandırma doğruluğuna sahip bir özellik alt kümesini seçmek amacıyla arama sürecinde belirli bir öğrenme modeli kullanılır. Özellik seçim sürecinin her aşamasında, bir özellik alt kümesi oluşturulur ve bir öğrenme modeli ile seçilen özellik alt kümesi değerlendirilir. Son olarak, en iyi değerlendirmeye sahip özellik alt kümesi, nihai özellik alt kümesi olarak belirlenir. Sarmal yöntemlerin çoğu, çözüm popülasyonunu en iyi çözüme doğru yönlendirmek için kullanıldığı yinelemeli özellik seçim süreçlerini kullanır. Ancak sarmal yaklaşımın arama sürecinde bir öğrenme modelinin yer alması daha iyi bir özellik alt kümesi seçilebilmesine olanak sağlamasına karşın çalışma süreleri uzun zaman alabilir. Öğrenme algoritmaları ile birlikte kullanılarak filtre yöntemine göre daha iyi bir performans göstermesi ve özellikler arasındaki etkileşimleri yakalayabilmesi nedeniyle bu tez çalışmasında sarmal yöntemler ele alınmıştır.

Gömülü yöntemde ise özellik seçme süreci, öğrenme algoritmasının bir parçası olarak kabul edilir ve eş zamanlı olarak gerçekleşir [5]. Hibrit yöntemde hem filtre hem de sarmal modellerden yararlanmaya ve filtre modelinin hesaplama verimliliği ile sarmal modelin doğruluğunu dengeleyen bir model sağlamaya çalışılır. Aslında hibrit yöntemin amacı hem verimlilik hem de etkinlik sağlayan bir yöntem sağlamaktır. Filtre, sarmal ve gömülü yöntemlerin özellik seçme süreçleri Şekil 2.1’de görselleştirilmiştir.



Şekil 2.1. Özellik seçim yöntemleri

Özellik seçimi ile ilgili yapılan çalışmalar göstermiştir ki, maksimum doğrulukla bir öğrenme modeli oluşturmak için bir dizi optimal özellik alt kümesini belirlemek zor bir problemdir. En iyi alt küme, olası tüm alt kümelerin kapsamlı bir arama stratejisi kullanılarak değerlendirilmesiyle bulunabilir. Her ne kadar bu yöntem optimal bir özellik alt kümesini garanti etse de optimal çözümü bulmak çok zaman alır. Bu durum orta büyüklükteki veri kümeleri için bile pratik olmayacaktır. Olası tüm alt kümeleri değerlendirmek çok maliyetli olduğundan hem hesaplama zorluğu hem de uygunluk açısından kabul edilebilir bir özellik alt kümesinin araştırılması gerekir [46, 47].

Bu tarz çözülmesi ve doğrulanması zor optimizasyon problemlerin üstesinden gelmek için öğrenme temellerine dayalı yöntemlere ek olarak bazı meta-sezgisel yöntemler önerilmiştir ve optimele yakın bir özellik kümesi bulmaya çalışılmıştır. Meta-sezgisel algoritmalar, optimal çözümü bulmak yerine kabul edilebilir bir sürede optimele yakın çözümler bulabilen yaklaşımlardır [48]. Yerel optimumdan kaçma stratejilerine sahip olan meta-sezgisel algoritmalar, birçok farklı optimizasyon probleminde kullanılabilen algoritmalarlardır. Uygun özelliklerin bulunmasını sağlayan kapsamlı bir arama orta büyüklükteki veri kümeleri için bile hesaplama açısından mümkün olmaması nedeniyle birçok özellik seçme yöntemi, büyük boyutlu veri setinde artan hesaplama zorluğunu önlemek için meta-sezgisel algoritmalar kullanır. Meta-sezgisel algoritmaların çoğunun başlangıcında, bir dizi başlangıç çözümü

rastgele oluşturulur ve daha sonra oluşturulan popülasyonun bireysel çözümlerinin optimalliğini hesaplamak için bir uygunluk fonksiyonu kullanılır. Her yineleme adımı bu çözümlerin optimalliğini iyileştirilmeye çalışır. Belirlenmiş sonlandırma kriterleri sağlanıncaya kadar döngü tekrarlanır [49, 50].

Meta-sezgisel yaklaşımlar sürü zekası, evrimsel algoritmalar, fizik tabanlı algoritmalar ve insan tabanlı algoritmalar gibi çeşitli kategorilere ayrılmaktadır. Sürü zekası algoritmaları, literatürde en çok kullanılan ve en önemli meta-sezgisel yaklaşımlardır. Sürü zekası algoritmaları bireysel olmayan ve kendi kendini organize eden sistemlerde toplu davranışlara dayanan bir optimizasyon türüdür. Genellikle yerel olarak ve çevreleriyle etkileşime giren basit aktörlerden oluşan bir popülasyondan oluşur. Bu kavram genellikle doğadan ilham alır ve her ajan kolay bir iş yapar, ancak bu ajanlar arasındaki yerel etkileşimler ve kısmen rastgele etkileşimler, bireysel ajanlar tarafından küresel davranışların ortaya çıkmasına neden olur. Parçacık sürü optimizasyonu, karınca kolonisi optimizasyonu, yapay arı kolonisi gibi birçok sürü zekası algoritması özellik seçiminde kullanılmaktadır.



3. PARÇACIK SÜRÜ OPTİMİZASYONU

Sürü kelimesi, düzensiz hareket eden bireyler veya doğada birlikte yaşayan böcekler, kuşlar ve balıklar gibi nesneler topluluğu anlamına gelir. Bir sürüyü etkileşen halinde olan bireyler topluluğu olarak düşünen araştırmacılar, bu bireylerin yiyecek arama davranışlarını modelleyerek ve simüle ederek çeşitli meta-sezgisel algoritmalar geliştirdiler.

Sürü zekası; kuş sürüleri, balık sürüleri, karınca kolonileri, hayvan sürüleri, bakteriyel büyüme ve mikrobiyal zeka sürüleri gibi sosyal varlıkların bireysel olmayan, kolektif ve kendi kendine organize işbirlikçi davranışlarına dayalı olarak koordine olan, birçok bireyden oluşan, doğal ve yapay sistemlerle ilgilenen bir yapay zeka optimizasyon dalıdır. Bir sürünün üyeleri aktif, dinamik, basit ve sürü halinde organize edilmiş basit etkileşimli ajanlardan oluşmalıdır. Sürü içinde bu işbirlikçi davranış nedeniyle, rastgele aramadan daha iyi bir arama stratejisi ortaya çıkar. Bu şekilde elde edilen akıllı arama stratejisi, genel olarak sürü zekası olarak adlandırılabilir [51, 52].

1990'lı yılların ortalarına kadar sürü zekası yaklaşımı; popülasyon kullanımı, stokastik doğası ve uygulama alanları açısından benzerlikleri nedeniyle evrimsel hesaplama yaklaşımları altında değerlendirilmiştir. Ancak günümüzde, sürü zekası ve evrimsel hesaplamanın altında yatan felsefeler arasındaki bazı doğal farklılıklar nedeniyle sürü zekası kendi kimliğine kavuşmuştur. Evrimsel hesaplama, biyolojik evrimden ilham alırken sürü zekası, basit ajanların kolektif ve sinerjik davranışlarını taklit etmeye çalışır. Gerçek dünya problemlerini çözmedeki basitliği ve etkinliği nedeniyle sürü zekası, bir optimizasyon algoritmaları sınıfı olarak oldukça popüler hale gelmiştir.

Sürü zekası algoritmaları, temel olarak varyasyon aşaması ve seçim aşaması olmak üzere iki aşamadan oluşmaktadır. Bu aşamalar, araştırma ve yoğunlaşma arasındaki dengeyi korumaktan ve tüm sürüyü zorlamaktan sorumludur. Varyasyon aşaması, arama uzayının farklı alanlarını araştırırken seçim aşaması, önceki deneyimlerin kullanımı için çalışır [51].

Sürü zekası algoritmalarından biri olan PSO algoritmasında diğer algoritmalara göre daha az parametreye sahip olması temel bir avantaj olarak görülebilir. Ayrıca yapısında yer alan parçacıkların birbirleri ile olan etkileşiminden dolayı daha iyi sonuçlar elde edilebilir. Buna

rağmen, özellik seçimi gibi büyük boyutlu arama alanı gerektiren problemlerde optimuma doğru yavaş bir yakınsama gösterebilir. Üstelik büyük veri kümeleri söz konusu olduğunda doğruluğu az olan sonuçlar verir [53]. Bu dezavantajlı durumların giderilmesi ve algoritmada ele alınan parametrelerini etki ve etkileşimlerin incelenmesi için bu tez çalışmasında PSO algoritması ele alınmıştır.

Kuş, balık ve hayvan sürüleri gibi doğada sosyal olarak organize olmuş popülasyonları yöneten kavram ve kurallara dayanan PSO, Kennedy ve Eberhart tarafından simülasyon modellerine dayalı stokastik bir optimizasyon algoritması olarak geliştirilmiştir [11]. Bu algoritmada, arama uzayında stokastik olarak hareket eden bir arama noktası popülasyonu kullanılır. Aynı zamanda, deneyim olarak da adlandırılan, her bireyin eriştiği en iyi konum hafızada tutulur. Bu deneyim daha sonra popülasyonun bir kısmına veya tamamına iletilir ve bireylerin hareketi şimdiye kadar tespit edilen en iyi bölgelere doğru yönlendirilir. İletişim, algoritmanın yakınsama özellikleri üzerinde çok önemli bir rol oynayan sabit veya uyarlanabilir bir sosyal ağ tarafından belirlenir.

Bir sürü zekası algoritması olan PSO’da çözüm, sürü zekası ile donatılmış rastgele bir arama yoluyla elde edilir. Bu arama, rastgele oluşturulmuş bir dizi potansiyel çözüm tarafından yapılır. Her bir potansiyel çözüm parçacık olarak adlandırılır. Parçacıkların oluşturduğu topluluğa ise sürü adı verilir. Her parçacık bir dizi parametreden oluşur ve çok boyutlu uzayda bir noktayı temsil eder. Sürüdeki tüm parçacıklar, aynı yönetim ilkesi altında ayrı ayrı hareket eder. Sürüde bulunan parçacıklar hareket davranışlarını gerçekleştirirken, ilgili parçacıkların hızları ayarlanır ve ayarlanan hızlar önceki konumlara eklenerek parçacıklar hareket ettirilir [54]. Parçacıkların mevcut konum değerleri sürekli kontrol edilerek en iyi kişisel ve en iyi genel konuma doğru yönlendirilerek arama yapılması sağlanır. Ayrıca her parçacığın bir önceki en iyi konumunu takip etmesine yardımcı olan bir hafızası bulunur.

PSO’da arama, sosyal öğrenme ve bilişsel öğrenme olarak adlandırılan iki tür öğrenmeden etkilenir. Bu öğrenmeler esnasında parçacık, sosyal öğrenmenin bir sonucu olarak “*gbest*” denilen sürünün herhangi bir parçacığının ziyaret ettiği en iyi çözümü hafızasında saklar. Bilişsel öğrenmenin bir sonucu olarak parçacık, kendi başına o ana kadar ziyaret edilen ve “*pbest*” denilen en iyi çözümü belleğinde saklar [51].

D boyutlu bir arama uzayı için, t . adımında sürünün i . parçacık için konum ve hız vektörleri sırasıyla $x_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{iD}^t)^T$ ve $v_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{iD}^t)^T$ şeklinde temsil edilsin. t . adımda ziyaret edilen en iyi konum $p_i^t = (p_{i1}^t, p_{i2}^t, \dots, p_{iD}^t)^T$ iken g sürüdeki en iyi parçacığın indeksidir. i . parçacığın hızı Eş. 3.1'deki hız güncelleme denklemi kullanılarak güncellenir.

$$v_{id}^{t+1} = \omega v_{id}^t + c_1 r_1 (p_{id}^t - x_{id}^t) + c_2 r_2 (p_{gd}^t - x_{id}^t) \quad (3.1)$$

Konum ise Eş. 3.2'deki konum güncelleme denklemi kullanılarak güncellenir.

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad (3.2)$$

Burada, $d = 1, 2, \dots, D$ iken $i = 1, 2, \dots, N$ şeklinde ifade edilir. N , sürünün boyutudur ve c_1 ve c_2 parametrelerine hızlandırma parametreleri adı verilir. Arama sürecine rastgelelik katan r_1 ve r_2 katsayıları, değerleri her iterasyonda yenilenen ve $[0,1]$ aralığında düzgün dağılım ile rastgele olarak üretilen sayılardır. ω parametresine ise eylemsizlik ağırlığı adı verilmektedir.

Eş. 3.1'de verilmiş olan bir parçacığın hızının üç bileşene göre değiştiği gösterilmektedir. Birinci bileşen, bir eylemsizlik ağırlığı ile ölçeklenen eylemsizlik hareketidir. Bu bileşen genellikle alışkanlık davranışı olarak da bilinir. İkinci bileşen, genellikle hafıza veya kendini tanıma olarak bilinen, parçacığın kendi tecrübelerinden faydalananarak yaptığı bilişsel öğrenme hareketi ve üçüncü bileşen ise genellikle ekip çalışması veya sosyal bilgi olarak bilinen, parçacığın sürü ile bilgi paylaşımından faydalananarak yaptığı sosyal öğrenme hareketi şeklindedir.

Kişisel ve sosyal öğrenme faktörleri ya da hızlanma parametreleri olarak adlandırılan c_1 ve c_2 sabitleri, bir parçacığı sırasıyla $pbest$ ve $gbest$ 'e doğru iten stokastik hız terimlerinin ağırlıklarını temsil eder. Küçük değerler bir parçacığın hedef bölgelerden uzakta dolaşmasına izin verirken, büyük değerler parçacıkların hedef bölgelere doğru ani hareketine neden olur. c_1 ve c_2 sabitleri, pozitif değerli katsayılar olup, genellikle 0 ile 4 arasında değerler alır [55].

PSO algoritmasında, küresel araştırma ve yerel yoğunlaşmanın uygun şekilde kontrol edilmesi çok önemli bir konudur. Shi ve Eberhart tarafından evrim sürecinde yerel ve küresel aramaları dengelemek için PSO'nun orijinal versiyonuna eylemsizlik ağırlığı kavramı tanıtıldı [56]. Eylemsizlik ağırlığı, bir parçacığın önceki hızının mevcut zaman adımındaki hızına katkı oranını belirler. Uygun eylemsizlik ağırlığı seçimi, yerel ve küresel keşif arasında bir denge sağlayarak, optimal bir çözümün bulunması için algoritmanın çalışmasında daha az yinleme ya da başka bir deyişle iterasyon gerektirir ve zaman maliyetinin düşürülmesinde önemli bir rol oynar. Literatürde lineer azalan, rastgele ağırlıklandırma ve tavlama benzetimi algoritmasında yer alan soğutma fonksiyonu gibi eylemsizlik ağırlığı için çok sayıda öneri mevcuttur. Bu ağırlıkların hesaplanma formülleri Çizelge 3.1'de verilmiştir [56–58].

Çizelge 3.1. Eylemsizlik ağırlığı türleri ve formülleri

Eylemsizlik Ağırlığı	Formül
Lineer azalan	$\omega = \omega_{maks} - \frac{\omega_{maks} - \omega_{min}}{\text{maksimum iterasyon sayısı}} * \text{iterasyon}$
Rastgele ağırlıklandırma	$\omega = 0,5 + \frac{rand()}{2}$
Soğutma fonksiyonu	$\omega = \omega_{min} + (\omega_{maks} - \omega_{min})\lambda^{\text{iterasyon}-1}, \lambda = 0,95$

PSO algoritmasında öncelikle parametreler belirlenerek başlangıç sürüsünün pozisyonları ve hızları oluşturulur. Daha sonra tüm parçacık değerlerine ilişkin uygunluk kriterleri hesaplanır. Daha önceki iterasyonlarda elde edilen sonuçlar ile güncel sonuçlar karşılaştırılır ve güncel sonuçlar daha iyiye önceki sonuçlar ile yer değiştirilir. En iyi yerel değerler kendi arasında karşılaştırılır ve içlerindeki en iyi yerel değer, küresel değer olarak atanır. Daha sonra pozisyon ve hız değerleri güncellenir ve durdurma kriteri sağlanıncaya kadar bahsedilen işlemler tekrarlanır.

3.1. İkili Parçacık Sürü Optimizasyonu

PSO algoritması başlangıçta sürekli arama uzaylarındaki problemleri çözmek için önerildi. Ancak birçok optimizasyon probleminde, değişkenler arasında ve değişken seviyeleri arasında ayrık ya da niteliksel ayrımlar içeren bir alan ortaya çıkabilir. Özellik seçimi gibi

birçok optimizasyon problemi, ayrı bir arama uzayında meydana gelir. Bu nedenle Kennedy ve Eberhart tarafından ayrık problemler için ikili parçacık sürü optimizasyonu (BPSO) geliştirildi [59]. BPSO'nun işleyiş algoritması klasik PSO ile aynı olmasına rağmen, temel farklılığı Eş. 3.1 ve Eş. 3.2'de bulunan x_{id} , p_{id} ve p_{gd} değerlerinin 0 ya da 1 olarak kısıtlandırılmış olmasıdır. Ayrıca her bir parçacığın 1 değerini alma olasılığını belirlemek amacıyla, Eş. 3.1'de hesaplanan v_{id} hız değerlerini (0,1) aralığına dönüştürmek için Eş. 3.3'te verilen Sigmoid fonksiyonu kullanılır.

$$S(v_{id}) = \frac{1}{1 + e^{-v_{id}}} \quad (3.3)$$

(0,1) aralığına dönüştürülen hız değerleri yardımıyla, x_{id} parçacık değerlerinin 0 ya da 1 değerleri açısından güncellemesi Eş. 3.4'te ifade edilmiş olan koşula göre yapılır.

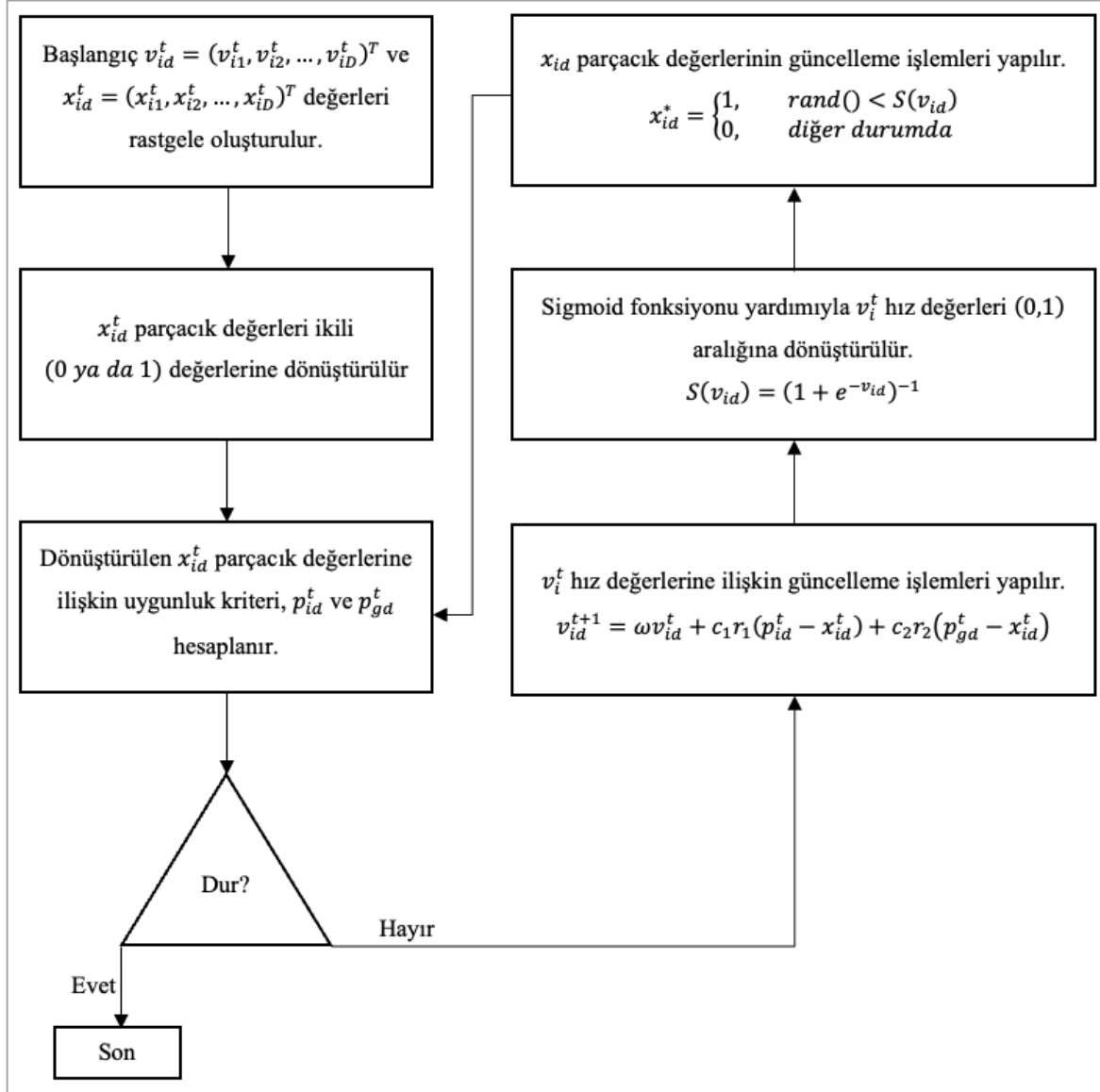
$$x_{id} = \begin{cases} 1, & rand() < S(v_{id}) \\ 0, & \text{diğer durumda} \end{cases} \quad (3.4)$$

Burada, $rand()$ ifadesi [0,1] aralığında düzgün dağılım ile rastgele olarak üretilen bir sayıdır [59].

PSO ve BPSO algoritmasının etkinliğini belirleyen araştırma (*exploration*) ve yoğunlaşma (*exploitation*) yetenekleridir. Bu iki durumun algoritma içerisinde dengelenmesi gerekmektedir. Büyük hız değerleri küresel araştırma yapmayı sağlarken küçük hız değerleri yerel yoğunlaşmayı sağlar. Ayrıca büyük hız değerleri iyi çözümlere ulaşmayı engelleyebilirken küçük hız değerleri en iyi çözüme ulaşmak için geçen süreyi arttırabilmektedir. Bu nedenlerden dolayı hız değerlerinin güncellenerek Eş. 3.5'te verilen belirli bir aralığa göre sınırlandırılması gerekmektedir [60].

$$v_{id}^{t+1} = \begin{cases} v_{max}, & v_{id}^{t+1} > v_{max} \\ v_{min}, & v_{id}^{t+1} < v_{min} \end{cases} \quad (3.5)$$

Burada, v_{min} ve v_{max} değerleri sırasıyla -6 ve 6 şeklindedir [59]. BPSO'ya ait akış diyagramı Şekil 3.1'de verilmiştir.



Şekil 3.1. BPSO akış diyagramı

PSO ve BPSO algoritmasında yer alan eylemsizlik ağırlığı küresel arama ve yerel arama yeteneği arasındaki dengeyi kontrol eder. Bu nedenle eylemsizlik ağırlığının uygun şekilde ayarlanması önemlidir. Ayrıca sürekli arama alanı, BPSO algoritmasının özelliklerine göre ayrık ikili uzaya dönüştürülür. Amaç, konum vektörünün elemanının 0'dan 1'e gitme olasılığını temsil etmek olduğundan transfer fonksiyonu [0,1] aralığıyla sınırlandırılmalıdır. Orijinal BPSO algoritmasının, Sigmoid transfer fonksiyonunun sınırlamaları nedeniyle uygun bir transfer fonksiyonun da seçilmesi önemlidir. Küresel arama ve yerel arama yeteneği arasındaki dengeyi kontrol etmek ve Sigmoid fonksiyonunun dezavantajlarını giderebilmek amacıyla bu tez çalışmasında sırasıyla farklı kaotik haritalar ve transfer fonksiyonları ele alınmıştır.

3.1.1. Kaotik haritalar

Matematikte zamana bağımlı dinamik bir sistem olarak tanımlanan kaos, bir düzensizlik durumu anlamına gelir. Kaotik sistem, başlangıç koşullarına karşı yüksek düzeyde bir duyarlılığa sahip olup, başlangıç koşullarındaki basit bir değişiklik bile sonuçlarda büyük değişikliklere yol açarak öngörülemez davranışlar sergiler. Çoğu doğrusal olmayan dinamik sistemlerden türetilen kaotik sistem; sınırlı, düzensiz ve başlangıç koşullarına duyarlı olması nedeniyle her zaman tahmin edilebilir değildir [61–63].

Kaos teorisi, deterministik yasaları takip eden ancak rastgele ve öngörülemez görünen sistemlerin, yani dinamik sistemlerin yürütülmesine odaklanır. Ayrık zamanlı dinamik bir sistem olan kaotik bir harita Eş. 3.6'daki gibi ifade edilebilir [64].

$$x_{n+1} = f(x_n), \quad n = 0, 1, 2, \dots \quad (3.6)$$

Kaotik haritalama, basit bir deterministik sistem tarafından üretilen rastgele kaotik dizileri oluşturmak için kullanılan bir mekanizmadır. Bu diziler doğrusal olmama, tekrarlanamazlık ve rastgelelik özelliklerine sahiptir [65]. Bu nedenle kaotik diziler, arama ajanlarının bir arama alanını daha eksiksiz keşfetmesine, algoritmanın yerel optimal çözümden kaçmasına ve popülasyonun çeşitliliğini artırmasına yardımcı olur.

Kaotik haritalar doğadan ilham alan algoritmaları geliştirmek içinde kullanılmıştır. PSO algoritmasında yer alan eylemsizlik ağırlığı, küresel arama ve yerel arama yeteneği arasındaki dengeyi kontrol eder. Bu nedenle eylemsizlik ağırlığının uygun şekilde ayarlanması önemlidir. Büyük bir eylemsizlik ağırlığı, küresel aramayı kolaylaştırırken küçük bir eylemsizlik ağırlığı, yerel aramayı kolaylaştırır. Yapılan çalışmalarda klasik algoritmalara göre yerel çözümden daha kolay bir şekilde kaçınılması sağlanmaya çalışılmıştır. Bu şekilde küresel yakınsamanın artırılması hedeflenmiştir.

BPSO süreci, yerel bir optimumda sıkışıp kalmakta ve bu durum da erken yakınsamaya neden olmaktadır [66]. Daha önce yapılan çalışmalarda, BPSO algoritmasına lojistik (*logistic*) ve çadır adı altında iki farklı kaotik harita entegre edilmiştir. Eş. 3.7'de lojistik haritanın matematiksel formülü verilmiştir. Burada X_n , mevcut popülasyon boyutunun maksimum popülasyon boyutuna oranı şeklinde temsil edilen 0 ile 1 arasında bir sayıdır ve

μ , 0 ile 4 arasında sabit bir parametre değeridir. Eş. 3.8'de ise Eş. 3.7'de verilen denklem sayesinde eylemsizlik ağırlığının nasıl değiştiği ifade edilmektedir ve burada t , iterasyon sayısını ifade etmektedir [66].

$$X_{n+1} = \mu X_n(1 - X_n) \quad (3.7)$$

$$\omega(t + 1) = 4\omega(t)(1 - \omega(t)) \quad (3.8)$$

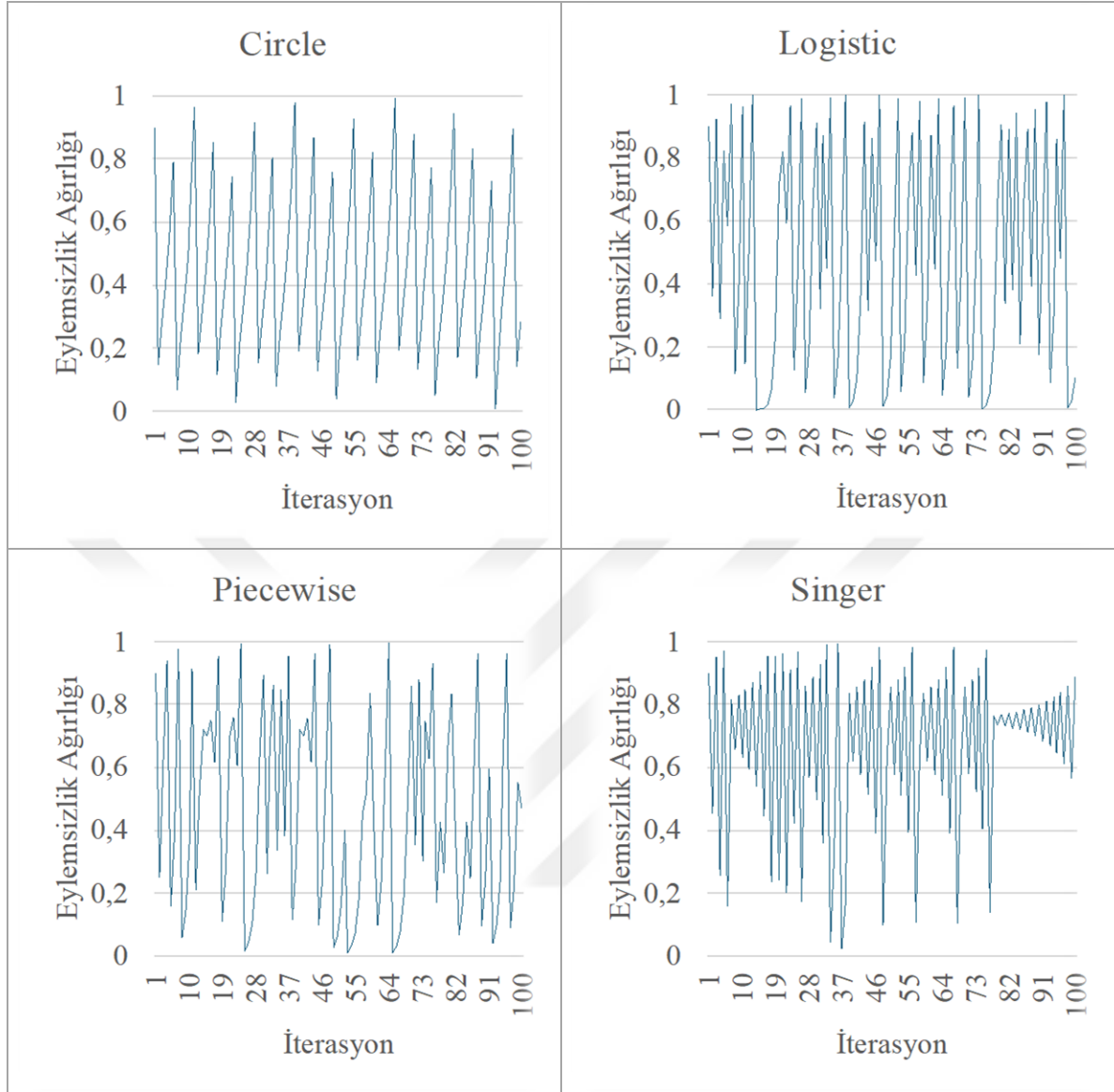
Araştırma yeteneği arama alanını geniş ölçekte araştırarak yeni ve daha iyi çözümler bulmaya odaklanırken, yoğunlaşma yeteneği yerel bölgedeki verileri kullanmaya odaklanır [67]. Uygun eylemsizlik ağırlığı seçimi, küresel ve yerel araştırma arasında bir denge sağladığından dolayı arama uzayındaki aramayı dengelemek için kaos teorisi kullanılmıştır.

Kaotik haritalar, rastgele faktörlerin kullanılmadığı belirli bir biçime sahiptir. Literatürde sıkça kullanılan bazı kaotik haritaların matematiksel formülleri ve değer aralıkları Çizelge 3.2'de belirtilmiştir.

Çizelge 3.2. Kaotik harita türleri ve matematiksel formülleri

Kaotik Harita	Matematiksel Formül	Parametre Değerleri	Değer Aralığı
Circle	$\omega_{t+1} = \text{mod} \left(\omega_t + d - \left(\frac{s}{2\pi} \right) \sin(2\pi\omega_t), 1 \right)$	$s = 0,5$ $d = 0,2$	(0,1)
Logistic	$\omega_{t+1} = c\omega_t(1 - \omega_t)$	$c = 4$	(0,1)
Piecewise	$\omega_{t+1} = \begin{cases} \frac{\omega_t}{l}, & 0 \leq \omega_t < l \\ \frac{\omega_t - l}{0,5 - l}, & l \leq \omega_t < 0,5 \\ \frac{1 - l - \omega_t}{0,5 - l}, & 0,5 \leq \omega_t < 1 - l \\ \frac{1 - \omega_t}{l}, & 1 - l \leq \omega_t < 1 \end{cases}$	$l = 0,4$	(0,1)
Singer	$\omega_{t+1} = \mu(7,86\omega_t - 23,31\omega_t^2 + 28,75\omega_t^3 - 13,302875\omega_t^4)$	$\mu = 1,07$	(0,1)

Burada, t iterasyon sayısını, ω eylemsizlik ağırlığını, ω_t t . iterasyondaki eylemsizlik ağırlığını, c, d ve μ dinamik sistemin kaotik davranışını belirleyen kontrol parametreleridir. Kaotik harita değerleri için bu haritaların görselleştirilmesi Şekil 3.2'te gösterilmektedir.



Şekil 3.2. Kaotik haritaların görselleştirilmesi

3.1.2. Transfer fonksiyonları

Transfer fonksiyonu sürekli arama uzayını ikili arama uzayına dönüştürme sürecinde önemli bir rol oynamaktadır. PSO algoritması da aynı şekilde başlangıçta sürekli arama uzaylarındaki problemleri çözmek için önerildi. Ancak birçok optimizasyon problemi, özellik seçimi, zamanlama ve yönlendirme gibi ayrık alanlarda tanımlanır ve bu nedenle ayrık problemler için BPSO geliştirilmiştir [59]. BPSO’da her bir parçacığın 1 değerini alma olasılığını belirlemek amacıyla v_{id} hız değerlerini $[0,1]$ aralığına dönüştürmek için Sigmoid fonksiyonu veya literatürdeki diğer adı S şekilli olan transfer fonksiyonu kullanılmıştır.

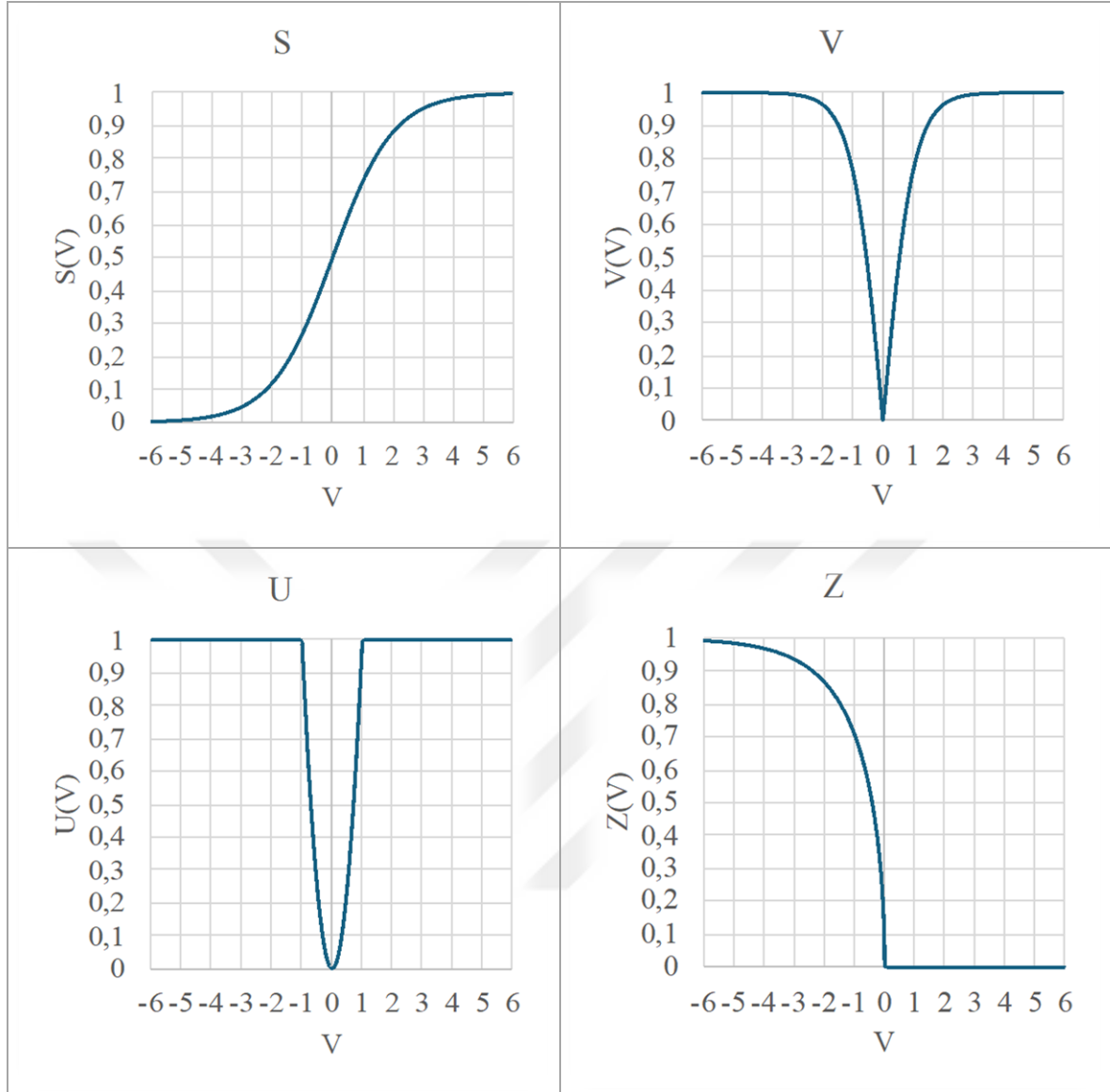
BPSO basit bir yapıya sahip olmasına rağmen bazı dezavantajlara sahiptir. Dezavantajların bir kısmı doğrudan PSO'nun eksikliklerine bağlı iken diğerleri ise transfer fonksiyonuyla ilgilidir. PSO'nun araştırma özelliği zayıftır ve bu nedenle yerel optimuma sıkışabilir. Aynı şekilde BPSO, PSO'nun hızını kullandığından bazen yerel optimuma takılabilir veya yavaş yakınsama oranı gösterebilir [68].

BPSO'nun diğer bir dezavantajı Sigmoid fonksiyonundan kaynaklanmaktadır. PSO'da pozitif veya negatif yöndeki büyük hız değerleri arasında fark yoktur. Hızın mutlak değerinin büyük olması, mevcut parçacığın konumunun uygun olmadığını ve optimum konuma ulaşmak için büyük bir hareket gerektiğini gösterir. Ayrıca hızın küçük bir mutlak değeri, mevcut parçacığın konumunun optimum çözüme yakın olduğunu ve optimum konuma ulaşmak için küçük bir mesafeye ihtiyaç duyulduğunu gösterir. BPSO'da pozitif yöndeki bir değer daha büyük bir olasılık oluşturur ve negatif yöndeki bir değer bir sonraki parçacığın konumu için olasılığı sıfır yapar [68].

S şeklindeki transfer fonksiyonlarının neden olduğu yakınsama problemlerini çözmek için Çizelge 3.3'te matematiksel formüllerinde yer verilen V şekilli, U şekilli ve Z şekilli gibi çeşitli transfer fonksiyonları önerilmiştir [26, 28, 69]. Farklı hız değerleri için transfer fonksiyonlarının görselleştirilmesi Şekil 3.3'te gösterilmektedir.

Çizelge 3.3. Transfer fonksiyon türleri ve matematiksel formülleri

Transfer Fonksiyonu	Matematiksel Formül
S	$S(V_i^{t+1}) = \frac{1}{1 + e^{-V_i^{t+1}}}$
V	$V(V_i^{t+1}) = \tanh(V_i^{t+1}) $
U	$U(V_i^{t+1}) = \alpha (V_i^{t+1})^\beta , \quad \alpha = 1, \beta = 2$
Z	$Z(V_i^{t+1}) = \sqrt{(1 - \alpha^{V_i^{t+1}})}, \quad \alpha = 2$



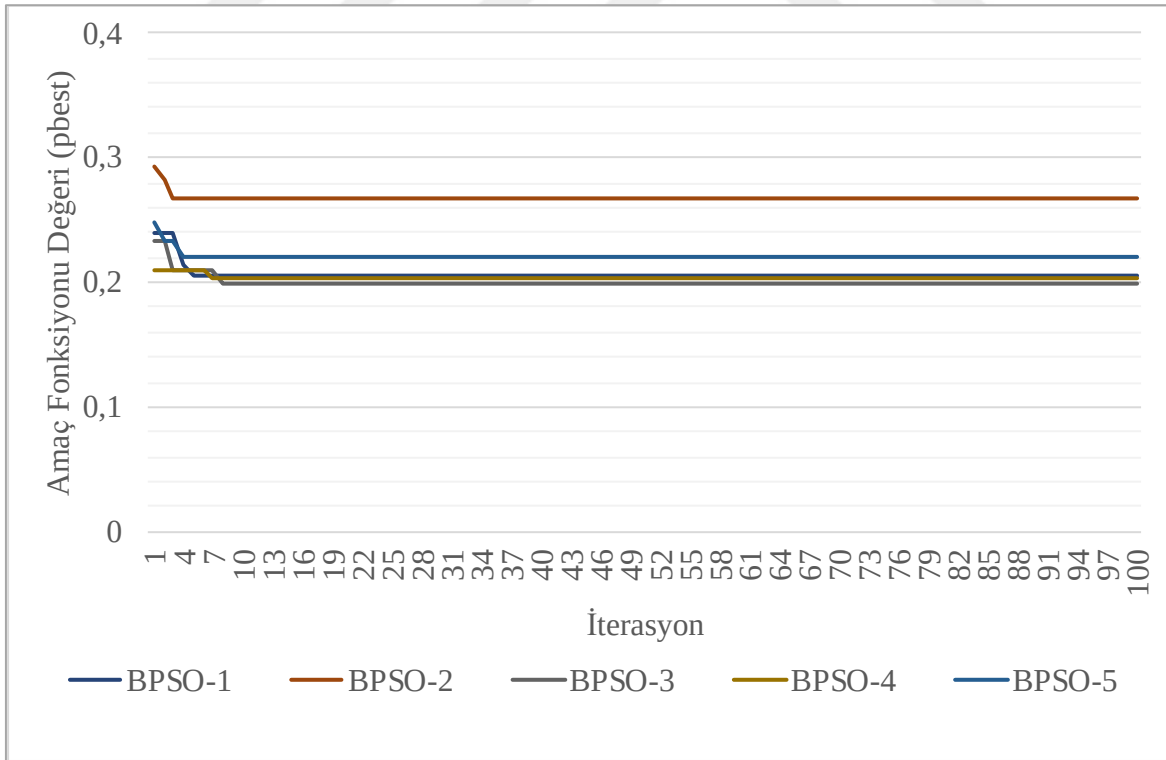
Şekil 3.3. Transfer fonksiyonlarının görselleştirilmesi

3.2. Modifiye Edilmiş İkili Parçacık Sürü Optimizasyonu

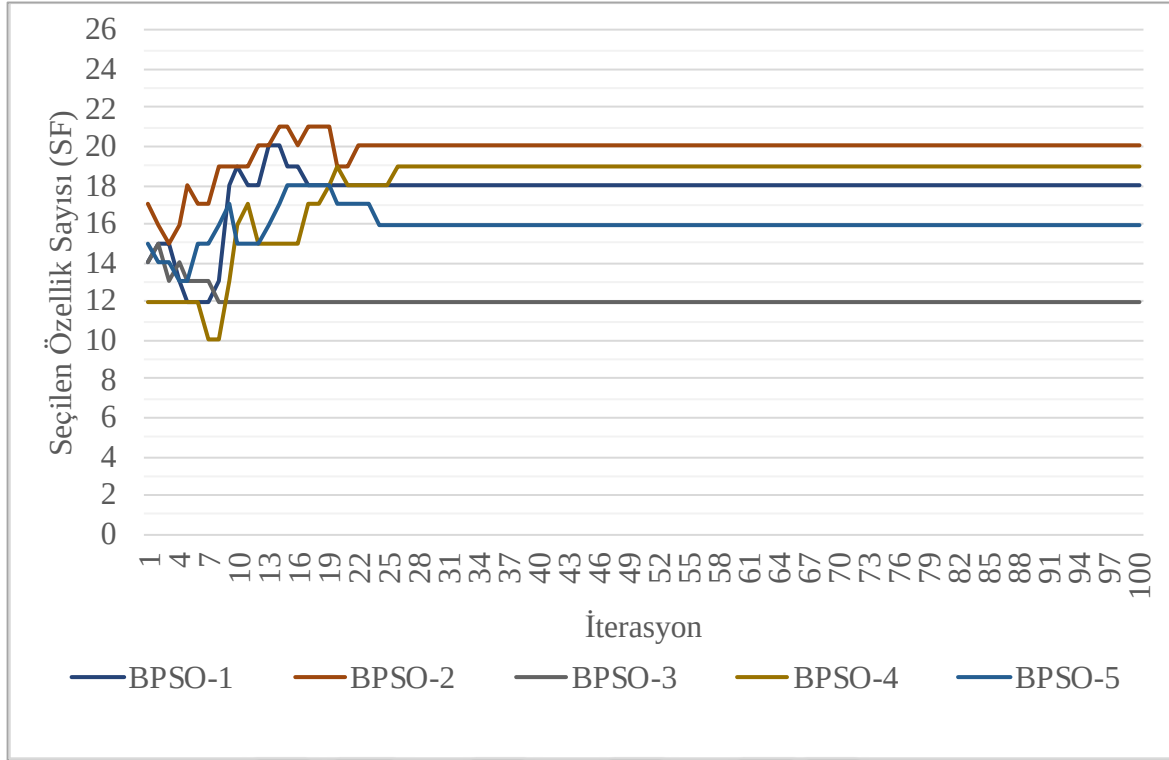
BPSO basit bir yapıya sahip olmasına rağmen araştırma özelliği zayıftır ve bu nedenle yerel optimuma sıkışabilir. Aynı şekilde BPSO, PSO'nun hızını kullandığından bazen yerel optimuma takılabilir veya yavaş yakınsama oranı gösterebilir [68]. Bu durum özellik seçme problemlerinde farklı özelliklerin seçimini engelleyerek sınıflandırma yönteminin performansını etkileyebilir. Bu durum bir örnek üzerinde gösterilirse, Şekil 3.4'te klasik BPSO algoritmasının kullanıldığı bir özellik seçme probleminde rastgele seçilmiş olan beş parçacığa ilişkin amaç fonksiyon değerlerinin değişim grafiği verilmiştir. Parçacıklar, neredeyse ilk 10 iterasyon içerisinde yakınsama sağlayıp geri kalan iterasyon da herhangi

bir iyileşme gösterememiştir. Şekil 3.5'te ise BPSO algoritmasının PSO'nun hızını kullanmasından dolayı seçilen özellik sayısının değişim grafiği verilmiştir. BPSO algoritmasında parçacıklara ilişkin hız değerleri çok büyümesi ya da çok küçülmesi farklı seçilmiş özellik kombinasyonları meydana gelmesini engeller. Bu durum sürekli aynı özelliklerin seçilip modeli daha iyi sınıflandırabilen farklı özelliklerin seçilememesine ve buna bağlı olarak model performansının gelişmemesine neden olabilmektedir.

Yerel optimuma takılmayı önlemek ve yavaş yakınsamanın önüne geçilebilmesi için farklı kaotik haritalar ve transfer fonksiyonları önerilmiştir. Buna rağmen, bilindiği kadarıyla önerilen bu yaklaşımların BPSO algoritması üzerindeki etkileri ve birbirleri ile olan etkileşimlerinin istatistiksel olarak incelendiği çalışma bulunmamaktadır. Bu doğrultuda parametrelerin 4^4 faktöriyel tasarım yardımıyla etki ve etkileşimleri incelenmiştir. Elde edilen sonuçlar doğrultusunda, BPSO algoritmasının performansının geliştirilebilmesi için BPSO algoritma yapısında bir değişiklik meydana gelmesi gerektiği sonucuna varılmış olup, yeni bir modifiye edilmiş BPSO (MBPSO) algoritması önerilmiştir.



Şekil 3.4. BPSO algoritması için amaç fonksiyon değerlerinin (pbest) değişimi



Şekil 3.5. BPSO algoritması için seçilen özellik sayılarının değişimi

Önerilen MBPSO algoritmasında BPSO algoritmasına göre en önemli farklılık, özellik seçimi aşamasında PSO'nun hızının kullanılmamasıdır. MBPSO'da hız değerleri yerine Eş. 3.9'da verilen parçacıkların güncel konum değerleri kullanılmaktadır.

$$x_{id}^{t+1} = x_{id}^t + SF(v_{id}^{t+1}) \quad (3.9)$$

Güncel konum değerleri kullanılmasıyla farklı özelliklerin seçilebilme olasılığını artırılıp, yerel optimuma takılmanın önlenmesi ve model performansının iyileştirilmesi amaçlanmıştır. Farklı özelliklerin seçilmesi sonucunda birbirinden farklı seçilmiş özellik kombinasyonları meydana gelecektir. Bu doğrultuda farklı seçilmiş özelliklerin bulunduğu sınıflandırma modelleri oluşturulacak ve birbirinden farklı birçok durum değerlendirilebilecektir.

MBPSO algoritmasında Eş. 3.1'de verilen denklem yardımıyla parçacıklara ilişkin güncel hız değerleri her iterasyonda hesaplanmaktadır. Hesaplanan v_{id} hız değerlerini BPSO algoritmasından farklı olarak (0,1) aralığına dönüştürmek yerine MBPSO algoritmasında (-1,1) aralığına dönüştürülür. Sigmoid fonksiyonu hız değerlerini (0,1) aralığına

dönüştürürken MBPSO algoritmasında kullanılan sınırlı fonksiyonun hız fonksiyonunu $(-1,1)$ aralığına dönüştürmesi gerekmektedir. Eğer hız değerleri $(0,1)$ aralığına dönüştürülürse konum değerlerinin sürekli büyüyeceği Eş. 3.9'da açıkça gözükmektedir. Büyük konum değerleri özelliklerin sürekli seçilmesine neden olacaktır. Tam tersi durum göz önüne alındığında, eğer hız değerleri $(-1,0)$ aralığına dönüştürülse konum değerleri sürekli küçülecektir. Küçük konum değerleri özelliklerin seçilmemesine neden olacaktır. Bu dengenin sağlanabilmesi için uygun sınırlı fonksiyonlar yardımıyla hız fonksiyonunun $(-1,1)$ aralığına dönüştürülmesi gerekmektedir.

BPSO algoritmasında kullanılan ve Eş. 3.3'te verilen Sigmoid fonksiyonu yerine MBPSO algoritmasında literatürde sınırlı fonksiyonlar olarak tanımlanan ve hız değerlerini $(-1,1)$ aralığına dönüştüren $\tanh$, $\sin$ ve $\cos$ gibi uygulayıcının belirleyeceği farklı fonksiyonlar kullanılabilir. Bu tez çalışmasında ele alınan bazı sınırlı fonksiyonların matematiksel formülleri ve görselleştirilmeleri Çizelge 3.4 ve Şekil 3.6'da verilmiştir.

Çizelge 3.4. Sınırlı fonksiyon türleri ve matematiksel formülleri

Sınırlı Fonksiyon	Matematiksel Formül
SF1	$f(x) = \tanh(x)$
SF2	$f(x) = 2 * \left \operatorname{erf} \left(\frac{\sqrt{\pi}}{4} * x \right) \right - 1$
SF3	$f(x) = \sin(x)$

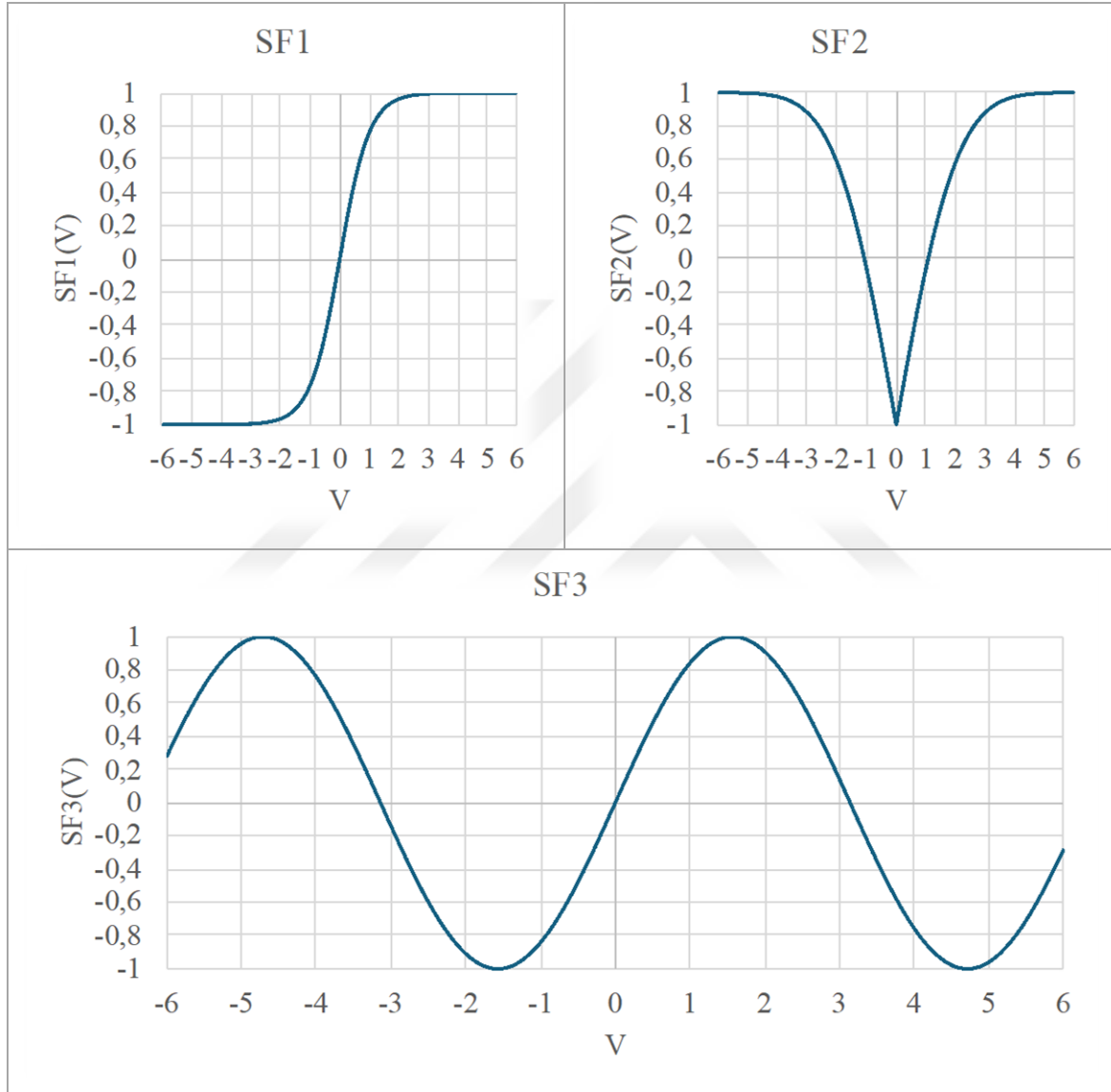
Konum değerlerinin çok büyümesini ve küçülmesini engellemek amacıyla BPSO algoritmasında var olan kısıtlamaya benzer şekilde konum değerlerinin güncellenerek Eş. 3.10'da verilen belirli bir aralığa göre sınırlandırılması gerekmektedir.

$$x_{id}^{t+1} = \begin{cases} x_{max}, & x_{id}^{t+1} > x_{max} \\ x_{min}, & x_{id}^{t+1} < x_{min} \end{cases} \quad (3.10)$$

Burada, x_{min} ve x_{max} değerleri sırasıyla 0 ve 1 şeklindedir. x_{id} konum değerlerinin 0 ya da 1 değerleri açısından güncellemesi Eş. 3.11'de ifade edilmiş olan koşula göre yapılır.

$$x_{id}^* = \begin{cases} 1, & \text{eşik değeri} < x_{id} \\ 0, & \text{diğer durumda} \end{cases} \quad (3.11)$$

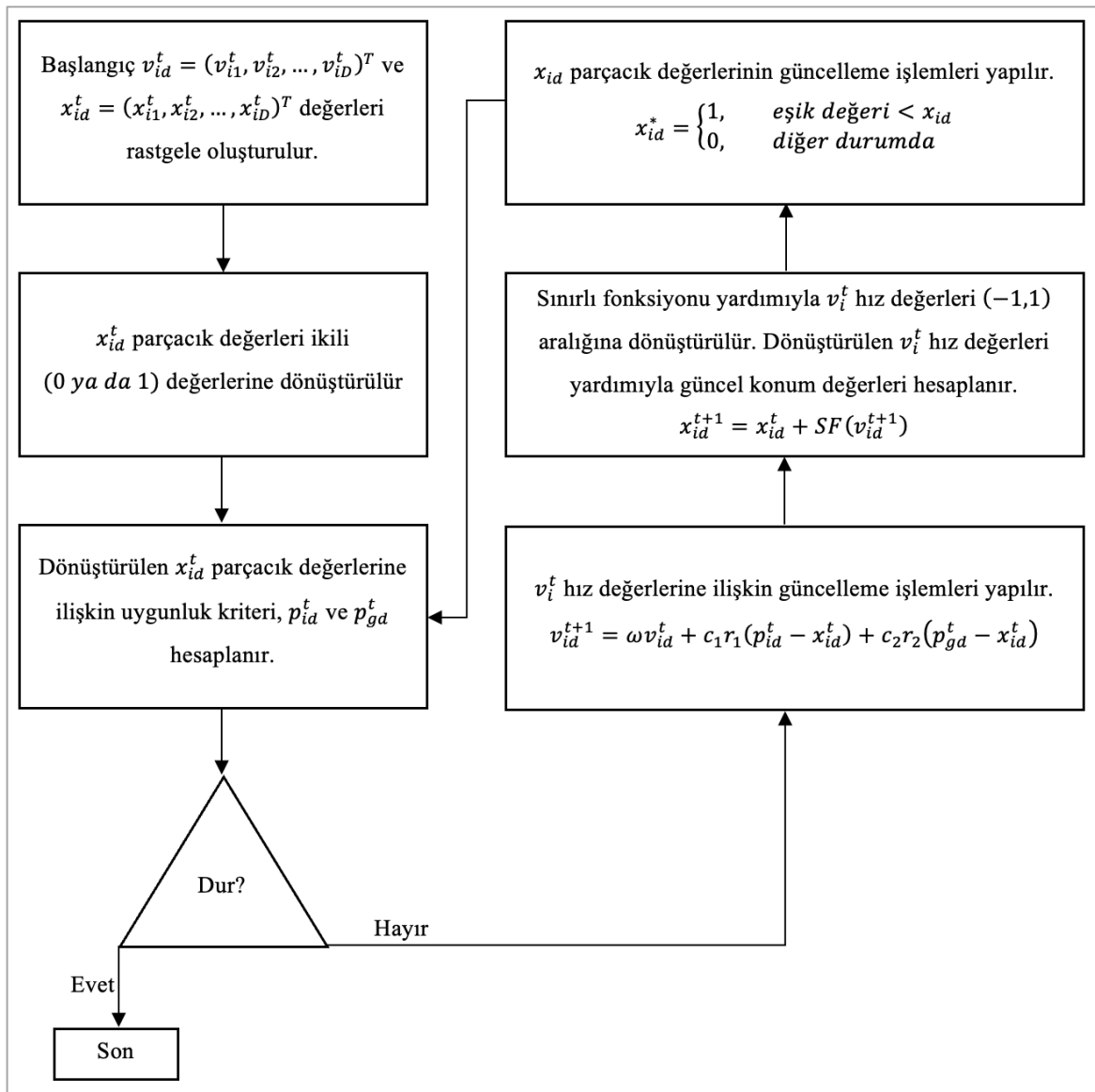
Burada, eşik değeri $[0,1]$ aralığında uygulayıcı tarafından belirlenmiş bir sayıdır.



Şekil 3.6. Sınırlı fonksiyonların görselleştirilmesi

BPSO’da bu eşik değeri, Eş. 3.4’te görüldüğü gibi, $[0,1]$ aralığında düzgün dağılım ile rastgele olarak üretilen bir sayıdır ve $rand()$ ile ifade edilir. Ancak bu rastgele değer çoğu zaman uygun olmayan özelliklerin seçilmesine ve potansiyel özelliklerin seçilememesine yol açar. Bu durumda BPSO algoritmasının yakınsaması gecikecek ve optimal çözüme ulaşma şansı azalacaktır. Örneğin BPSO algoritmasında, $S(v_{id}^{t+1})$ 0,9 gibi yüksek bir değere sahip bir özellik olmasına rağmen, oluşturulan rastgele değer 0,9’dan büyükse modele dahil

edilemez. $S(v_{id}^{t+1})$ değeri 0,05 kadar düşük bir değere sahip özellik olmasına rağmen, üretilen rastgele değer 0,05'ten küçük olması durumunda modele dahil olacaktır. Ayrıca oluşturulan rastgele değer 0,001 gibi çok küçükse, $S(v_{id}^{t+1})$ değerleri 0,005; 0,4 veya 0,99 olan özelliklerin modele dahil olma şansı eşit olacaktır. BPSO algoritmasında yer alan bu dezavantajlardan dolayı problemin amacına göre MBPSO algoritmasında rastgele sayı yerine uygulayıcı tarafından belirlenen eşik değeri, özelliklerin seçilip seçilmeyeceğine karar verir. Ayrıca MBPSO algoritmasında yakınsamanın gecikmesi engellenerek optimal çözüme ulaşma şansı artırılmaktadır. MBPSO'ya ait akış diyagramı Şekil 3.7'de verilmiştir.



Şekil 3.7. MBPSO akış diyagramı

4. FAKTÖRİYEL TASARIM

Fisher [70] ve Yates [71] tarafından önerilen faktöriyel tasarım, iki ya da ikiden fazla faktörün ana etkilerini ve birbirleriyle olan etkileşimleri aynı anda araştırmak için kullanılan bir tasarımdır [72]. Deneylerde birden çok faktörün etkileri aynı anda incelenir ve farklı faktörlerin bütün kombinasyonlarını içerir. Bir faktörün diğer faktörlerden bağımsız olarak ortaya koyduğu sonuçlar ana etki olarak ifade edilirken, iki ya da daha fazla sayıda faktör kombinasyonlarının etkileşimleri ise etkileşim etkisi olarak ifade edilmektedir.

Faktöriyel tasarımın en önemli avantajı, tüm faktörlerin bağımlı değişken üzerindeki etkisi ve faktörlerin birbirleriyle olan etkileşimlerini inceleyen deney tasarım yöntemi olmasıdır. Faktöriyel tasarımlar, birden fazla faktörün etkilerinin aynı anda araştırılmasına olanak sağlar. Bu nedenle zaman ve maliyet açısından tek seferde tek faktör yaklaşımına göre üstünlükleri vardır. Faktöriyel tasarımların bu özelliği, performans testi ve süreç geliştirme gibi mühendislik uygulamalarında araştırmacılar tarafından sıkça kullanılmasına neden olmaktadır [73].

Faktöriyel tasarım, faktör sayısının birden fazla olduğu durumlarda, bu faktörlerin ve tüm etkileşimlerinin incelenmesinde kullanılan ve deney sonucunu etkileyebilecek bir yaklaşımdır. Faktörler arası etkileşim, deney sonucunu etkileyecek önemli bir durumdur. Seçilen herhangi bir faktörün etkisi, diğer faktörlerden biri veya birkaçının seviye değerine bağımlıysa bu faktörler arasında etkileşim olduğu ifade edilir [71].

A, B, C ve D gibi dört faktörün seviyeleri sırasıyla i, j, k ve l olsun. Bu durumda, Eş. 4.1’de verilen 4^4 faktöriyel tasarıma ait matematiksel model,

$$\begin{aligned}
 y_{ijklm} = & \mu + \alpha_i + \beta_j + \gamma_k + \delta_l + \alpha\beta_{ij} + \alpha\gamma_{ik} + \alpha\delta_{il} + \beta\gamma_{jk} + \beta\delta_{jl} + \gamma\delta_{kl} \\
 & + \alpha\beta\gamma_{ijk} + \alpha\beta\delta_{ijl} + \alpha\gamma\delta_{ikl} + \beta\gamma\delta_{jkl} + \alpha\beta\gamma\delta_{ijkl} \\
 & + \varepsilon_{ijklm}, \quad i, j, k, l = 1, 2, 3, 4 \quad m = 1, 2, \dots, N
 \end{aligned} \tag{4.1}$$

şeklinde ifade edilir. Burada y_{ijklm} bağımlı değişkeni, μ genel ortalama, α_i A faktörünün i . seviyesinin etkisi, β_j B faktörünün j . seviyesinin etkisi, γ_k C faktörünün k . seviyesinin etkisi, δ_l D faktörünün l . seviyesinin etkisi ve N birim sayısıdır. Ayrıca $\alpha\beta_{ij}$, $\alpha\gamma_{ik}$, $\alpha\delta_{il}$, $\beta\gamma_{jk}$,

$\beta\delta_{jl}$ ve $\gamma\delta_{kl}$ iki faktörlü etkileşim etkisini; $\alpha\beta\gamma_{ijk}$, $\alpha\beta\delta_{ijl}$, $\alpha\gamma\delta_{ikl}$ ve $\beta\gamma\delta_{jkl}$ üç faktörlü etkileşim etkisini; $\alpha\beta\gamma\delta_{ijkl}$ dört faktörlü etkileşim etkisini ve ε_{ijklm} hata terimini ifade etmektedir.

Faktöriyel tasarımıdaki ana amaç, ele alınan faktörlerin etkilerini ve bu faktörler arasındaki etkileşimlerin istatistiksel olarak anlamlı olup olmadığının test edilmesidir. Bu test için kullanılan hipotez, “ H_0 : Faktörün ana etkisi (Faktörler arasındaki etkileşim) istatistiksel olarak anlamlı değildir.” şeklinde ifade edilir. H_0 , ele alınan faktörün seviyeleri arasında anlamlı bir fark olmadığını (veya ilgili faktörler arasında anlamlı bir etkileşim olmadığını) ima eder. Tüm durumlar göz önüne alındığında, 4^4 faktöriyel tasarımına ait 15 farklı H_0 hipotezi test edilmektedir.

Faktöriyel tasarımda, hata terimlerinin ortalamasının sıfır ve varyansının σ^2 olup bağımsız ve normal dağılımdan geldiği varsayılır. Normallik varsayımı altında, 15 farklı H_0 hipotezini test etmek için Çizelge 4.1’de verilen F testi kullanılır.

Çizelge 4.1. Varyans analizi tablosu

Kaynak	Serbestlik Derecesi	Kareler Toplamı	Kareler Ortalaması	F
A	$a-1$	KT_A	$KO_A = KT_A / (a-1)$	$F_A = KO_A / KO_{Hata}$
B	$b-1$	KT_B	$KO_B = KT_B / (b-1)$	$F_B = KO_B / KO_{Hata}$
C	$c-1$	KT_C	$KO_C = KT_C / (c-1)$	$F_C = KO_C / KO_{Hata}$
D	$d-1$	KT_D	$KO_D = KT_D / (d-1)$	$F_D = KO_D / KO_{Hata}$
A*B	$(a-1)(b-1)$	KT_{AB}	$KO_{AB} = KT_{AB} / ((a-1)(b-1))$	$F_{AB} = KO_{AB} / KO_{Hata}$
A*C	$(a-1)(c-1)$	KT_{AC}	$KO_{AC} = KT_{AC} / ((a-1)(c-1))$	$F_{AC} = KO_{AC} / KO_{Hata}$
A*D	$(a-1)(d-1)$	KT_{AD}	$KO_{AD} = KT_{AD} / ((a-1)(d-1))$	$F_{AD} = KO_{AD} / KO_{Hata}$
B*C	$(b-1)(c-1)$	KT_{BC}	$KO_{BC} = KT_{BC} / ((b-1)(c-1))$	$F_{BC} = KO_{BC} / KO_{Hata}$
B*D	$(b-1)(d-1)$	KT_{BD}	$KO_{BD} = KT_{BD} / ((b-1)(d-1))$	$F_{BD} = KO_{BD} / KO_{Hata}$
C*D	$(c-1)(d-1)$	KT_{CD}	$KO_{CD} = KT_{CD} / ((c-1)(d-1))$	$F_{CD} = KO_{CD} / KO_{Hata}$
A*B*C	$(a-1)(b-1)(c-1)$	KT_{ABC}	$KO_{ABC} = KT_{ABC} / ((a-1)(b-1)(c-1))$	$F_{ABC} = KO_{ABC} / KO_{Hata}$
A*B*D	$(a-1)(b-1)(d-1)$	KT_{ABD}	$KO_{ABD} = KT_{ABD} / ((a-1)(b-1)(d-1))$	$F_{ABD} = KO_{ABD} / KO_{Hata}$
A*C*D	$(a-1)(c-1)(d-1)$	KT_{ACD}	$KO_{ACD} = KT_{ACD} / ((a-1)(c-1)(d-1))$	$F_{ACD} = KO_{ACD} / KO_{Hata}$
B*C*D	$(b-1)(c-1)(d-1)$	KT_{BCD}	$KO_{BCD} = KT_{BCD} / ((b-1)(c-1)(d-1))$	$F_{BCD} = KO_{BCD} / KO_{Hata}$
A*B*C*D	$(a-1)(b-1)(c-1)(d-1)$	KT_{ABCD}	$KO_{ABCD} = KT_{ABCD} / ((a-1)(b-1)(c-1)(d-1))$	$F_{ABCD} = KO_{ABCD} / KO_{Hata}$
Hata	$N-abcd$	KT_{Hata}	$KO_{Hata} = KT_{Hata} / (N-abcd)$	
Toplam	$N-1$	KT_{Toplam}		

Çizelge 4.1’de ifade edilen KT ve KO, sırasıyla kareler toplamını ve kareler toplamının serbestlik derecesine oranı olarak tanımlanan kareler ortalamasını ifade eder. Ayrıca a , b , c ve d ifadelerinin her biri ilgili faktörlerin düzey sayısını belirtir ve 4^4 faktöriyel deney için $a = b = c = d = 4$ ’tür. N , toplam gözlem sayısıdır; 4^4 faktöriyel deney için bu sayı $4^4 r$ ’dir. Burada r tekrar sayısıdır. Tablodaki F testleri v_1 ve v_2 serbestlik derecelerine sahip bir F dağılımına sahiptir. Burada v_1 , ilgilenilen faktörün (veya etkileşimin) serbestlik derecesini temsil ederken $v_2 = N - a * b * c * d$, hatanın serbestlik derecesini temsil eder [74].

F istatistiklerinin hesaplanan değeri, önceden belirlenen anlamlılık seviyesi α için karşılık gelen kritik değerden büyükse, o zaman teste karşılık gelen H_0 hipotezi reddedilir. Benzer şekilde, etkilerin önemini belirlemek için p -değeri de kullanılabilir. Başka bir deyişle, p -değeri önceden belirlenen α ’dan küçükse H_0 hipotezi reddedilir.



5. ÖZELLİK SEÇİM YÖNTEMLERİNİN DEĞERLENDİRİLMESİ

Özellik seçim yöntemlerinin karşılaştırılması için seçilen özelliklere sınıflandırma algoritmaları uygulanıp değerlendirme yapılması gerekmektedir.

5.1. Sınıflandırma Algoritması

Öğrenme algoritmalarına bağımlı olmaları nedeniyle bu tez çalışmasında ele alınan sarmal yöntemler, en yüksek sınıflandırma doğruluğuna sahip bir özellik alt kümesini seçmek amacıyla özellik seçme sürecinde belirli bir öğrenme modeli kullanılır. Seçilen özellik alt kümesinin verimliliğini ölçmek için bir sınıflandırma algoritması kullanılır. Sarmal yöntemler ile birlikte sıklıkla kullanılan destek vektör makineleri (SVM), k-en yakın komşu ve Adaboost gibi farklı sınıflandırıcılar mevcuttur. Bu tez çalışmasında BPSO algoritmasının yapısı ve parametreleri ön planda olduğundan dolayı sadece SVM sınıflandırma algoritması ele alınmıştır.

Vapnik tarafından önerilen SVM, çok geniş özellik uzaylarını ele alma potansiyeline sahip denetimli bir öğrenme algoritmasıdır [75]. SVM, farklı sınıflara ait bir grup nesne arasındaki mesafeyi maksimuma çıkarmak için karar yüzeyi olarak bir hiperdüzlem aramayı amaçlamaktadır. Literatürde bu amaca maksimal marj hiperdüzlemi adı da verilmektedir. Bu şekilde farklı sınıflara ait verilerin birbirinden en uygun şekilde ayrımı yapılır.

$x_1, x_2, \dots, x_n$ bir dizi n eğitim gözlemi ve bu gözlemlere karşılık gelen $y_1, y_2, \dots, y_n$ sınıf değişken değerleri olmak üzere, maksimal marj hiperdüzlemi optimizasyon problemi Eş. 5.1'de verilmiştir.

$$\underset{\beta_0, \beta_1, \dots, \beta_p, M}{\text{Maks}} \quad M$$

$$\sum_{j=1}^p \beta_j^2 = 1 \quad (5.1)$$

$$y_i(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \dots + \beta_p x_{ip}) \geq M, \quad i = 1, 2, \dots, n$$

Burada, p uzay boyutu, β hiperdüzlem parametreleri ve M hiperdüzlemin marjını ifade eder. Uygun β hiperdüzlem parametreleri belirlenerek hiperdüzlemin marjı maksimum olması amaçlanır [76].

5.2. Ayrıştırma Türü

Sarmal yöntemlerde sınıflandırma algoritmaları kullanılmadan önce veri setleri öncelikle eğitim ve test verisi olarak ikiye ayrılması gerekir. Özellik seçim sürecinde eğitim verileri kullanılırken seçilen özellik alt kümesinin uygunluğunun değerlendirilmesi için test verileri kullanılır. Yüzdesel bölme, n-bölme, tekrarlı n-bölme ve 1 eksiltme yöntemleri eğitim ve test verilerinin farklı ayrılmasına yönelik yaklaşımlardır. Bu tez çalışmasında BPSO algoritmasının yapısı ve parametreleri ön planda olduğundan dolayı sadece yüzdesel bölme yöntemi ele alınmıştır.

Özellik seçimi aşamasında öncelikle veriler eğitim ve test verisi olmak üzere iki farklı kümeye ayrılır. Eğitim verileri özellik seçim sürecinde kullanılarak uygun özelliklerin seçimini sağlarken test verileri seçilen bu özelliklerin uygunluğunu değerlendirmek için kullanılır. Sınıflandırma çalışmalarında sıkça kullanılan temel yaklaşımlardan biri olan yüzdesel bölmede ele alınan verinin belirli bir yüzdesi eğitim verisi ve test verisi olarak ayrılmaktadır. Verinin 1/3'ünün test verisi olarak ayrılması gibi genel yaklaşımlar olsa da bu durum değişkenlik gösterebilir.

5.3. Karmaşıklık Matrisi

Karmaşıklık matrisi, gerçek durum ile sınıflandırma yöntemlerinden elde edilen sonuçları gösteren bir matristir. İki sınıf değişkenine sahip olan veri setleri için karmaşıklık matrisinin gösterimi Çizelge 5.1'de ifade edilmiştir.

Çizelge 5.1. Karmaşıklık matrisi

		Gerçek Durum	
		Pozitif	Negatif
Tahmin	Pozitif	TP	FP
	Negatif	FN	TN

Burada, TP (*True Positive*) doğru pozitiflerin sayısı, TN (*True Negative*) doğru negatiflerin sayısı, FP (*False Positive*) yanlış pozitiflerin sayısı ve FN (*False Negative*) yanlış negatiflerin sayısı olmak üzere $N=TP+TN+FP+FN$ şeklinde tanımlanır. Karmaşıklık matrisi yardımıyla, $[0,1]$ aralığında değerler alan doğru sınıflandırma oranı (ACC), hata oranı, F1 ölçüsü, duyarlılık, özgüllük ve Matthews Korelasyon Katsayısı gibi farklı türlerde performans ölçüleri tanımlanmış olup, Çizelge 5.2’de matematiksel formülleri verilmiştir. Bu tez çalışmasında literatürde en yaygın şekilde kullanılan ACC ele alınmıştır.

Çizelge 5.2. Performans ölçüsü türleri ve matematiksel formülleri

Performans Ölçüleri	Matematiksel Formül
Doğru sınıflandırma oranı	$ACC = \frac{TP + TN}{N}$
Hata oranı	$ER = \frac{FP + FN}{N}$
F1 ölçüsü	$F1 = \frac{2TP}{2TP + FP + FN}$
Duyarlılık	$TPR = \frac{TP}{TP + FN}$
Özgüllük	$TNR = \frac{TN}{FP + TN}$
Matthews Korelasyon Katsayısı	$MCC = \frac{TN \times TP - FN \times FP}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}$



6. DENEYSEL SONUÇLAR

Çalışmanın bu bölümünde, BPSO algoritmasını oluşturan parametrelerin 4^4 faktöriyel tasarım yardımıyla etki ve etkileşimleri incelenmiş ve elde edilen sonuçlar doğrultusunda önerilen MBPSO algoritması için farklı veri setleri ve parametre değerleri için uygulama sonuçları karşılaştırılmıştır. Bu amaçla, çalışmalarda sıkça kullanılan “*Ionesphere*” veri seti bu çalışmada ele alınmıştır. Bu veri seti 351 gözlem ve 34 özellikten oluşmaktadır. Sınıf frekans sayısı ise 2’dir. Yapılan analizlerde, modellerin oluşturulması ve görselleştirilmesinde Microsoft Excel, IBM SPSS Statistics ve Python 3.11 programlarından yararlanılmıştır.

6.1. Problemin Tanımı

Verilerin modellenmesi ya da sınıflandırılmasının önemli hedeflerinden biri, eğitim verilerine ve mevcut özelliklere dayanarak tahminde bulunmaktır. Veri kümelerinde çok sayıda alakasız ve gereksiz özelliğin var olması, öğrenilen modellerin doğruluğunu önemli ölçüde etkileyeceği gibi modellerin öğrenme hızını da düşürecektir. Doğruluğun maksimum yapılması ve özellik sayısının minimuma indirilmesi, özellik seçimini konu alan çok amaçlı optimizasyon probleminde iki karşıt amaç olarak kabul edilir. Bu tür problemlerde az sayıda özellik seçimi ile yüksek doğruluğa sahip olunması amaçlanmaktadır. Bu amaçla bu tez çalışmasında ele alınan amaç fonksiyonu ya da uygunluk fonksiyonu Eş. 6.1’de tanımlanmıştır.

$$\text{Min } f = \beta \times (1 - \text{ACC}) + \gamma \times \frac{\text{SF}}{\text{TF}} \quad (6.1)$$

Burada, β ve γ sırasıyla doğru sınıflandırma oranı (ACC), seçilen özellik sayısı (SF) ve toplam özellik sayısı (TF) için tanımlanmış iki ağırlıktır. Ele alınan bu uygunluk fonksiyonu ile en az sayıda seçilen özellik ile yüksek doğruluk sınıflandırma değerleri elde edilmeye çalışılmıştır.

Özellik seçim yöntemlerinden olan sarmal yöntemin kullanıldığı bu tez çalışmasında sınıflandırma yöntemi olarak radyal temel fonksiyon çekirdeğine sahip SVM algoritması ve ayırıştırma türü olarak yüzdesel yöntem (%70 eğitim verisi, %30 test verisi) ele alınmıştır.

6.2. Uygulama

Algoritma parametrelerinin seçimi, meta-sezgisel algoritmaların hesaplama performansını belirler. Çoğu meta-sezgisel algoritma gibi PSO ya da BPSO algoritması da çeşitli alanlarda uygulanır ancak algoritma parametrelerinin belirlenmesine yönelik kesin yönergelere sahip değildirler. Algoritmanın performansı, parametre ayarlarının değiştirilmesinden önemli ölçüde etkilenebilir. Bu bakımdan bu parametrelere uygun değerlerin belirlenmesi önemlidir [77]. PSO ve BPSO üzerine yapılan çalışmalarda, parametreler genellikle deneme yanılma yaklaşımı kullanılarak tanımlanır veya sezgisel olarak belirlenir. Bilindiği kadarıyla önerilen bu yaklaşımların BPSO algoritması üzerindeki etkileri ve birbirleri ile olan etkileşimlerinin istatistiksel olarak incelendiği çalışma bulunmamaktadır. Bu doğrultuda öncelikle BPSO parametrelerinin 4^4 faktöriyel tasarım yardımıyla etki ve etkileşimleri incelenmiştir.

Yerel ve küresel arama yeteneği arasındaki denge eylemsizlik ağırlığı tarafından kontrol edilir. Büyük bir eylemsizlik ağırlığı ile küresel arama, küçük bir eylemsizlik ağırlığı ile de yerel arama daha kolay hale getirilir. Bu nedenle yakınsamanın birincil belirleyicisi olan eylemsizlik ağırlığı, BPSO arama prosedürünü önemli ölçüde etkileyecek olup bu değer doğru ayarlanması oldukça önemlidir.

Transfer fonksiyonunun algoritmanın hesaplama maliyetini etkilememesi, algoritmadan bağımsız olması ve kullanımını kolaylaştırabilmesi nedeniyle uygun fonksiyonun belirlenmesi BPSO algoritmasının performansını optimize etmede çok önemli bir adımdır [21, 69, 78].

BPSO algoritmasında $[0,1]$ aralığında ki rastgele bir sayı olan eşik değeri, özelliklerin seçilip seçilmeyeceğine karar verir. Bu değer çoğu zaman uygun olmayan özelliklerin seçilmesine ve potansiyel özelliklerin reddedilmesine yol açar. Bu durumda BPSO algoritmasının yakınsaması gecikir ve optimal çözüme ulaşma şansı azalır.

Parçacık sayısının problemin özelliklerine göre seçilmesi gerektiği yaygın bir bilgidir [79]. Daha yüksek parçacık sayıları daha yüksek kalitede sonuçlar sağlayabilir. Ancak sürü boyutu, hesaplama süresi ve çözüm kalitesi arasında bir denge vardır [73]. Van den Bergh ve Engelbrecht'e göre, büyük sürü boyutları çözüm kalitesini artırmasına rağmen hesaplama süresini de aynı şekilde artırmaktadır. Bu duruma karşın, eğer az sayıda parçacık kullanılırsa

PSO algoritmasında hesaplama süresi azalmasına rağmen yerel bir minimumla sonuçlanabilir. [80].

Eylemsizlik ağırlığı, transfer fonksiyonu, eşik değeri ve parçacık sayısı BPSO algoritmasının performansının belirlenmesinde önemli parametrelerdir. Bu doğrultuda, özellik seçimi ve sınıflandırma probleminde kullanılan BPSO algoritma parametrelerinin etki ve birbirleri ile olan etkileşimlerini incelemek için 4^4 faktöriyel tasarımı ele alınmıştır. BPSO algoritmasının performansını etkileyen faktör ve bu faktörlerin düzeyleri Çizelge 6.1’de verilmiştir.

Çizelge 6.1. Faktör ve düzeyleri

Faktör	Düzeyler			
A: Kaotik Harita	Circle	Logistic	Piecewise	Singer
B: Transfer Fonksiyonu	S	V	U	Z
C: Eşik Değeri	0,5	0,6	0,7	0,8
D: Parçacık Sayısı	10	25	50	100

Sınıflandırma algoritması olarak SVM ve “*Ionesphere*” veri setinin kullanıldığı bu tasarımda her işlemin üç kez tekrarlanmış ve 4^4 faktöriyel tasarımında yanıt değişkeni olarak ele alınan ACC’nin toplam sayısı $N = 4^4 * 3 = 768$ olmuştur. Algoritmanın 30 kez çalıştırılması ile elde edilen ortalama değer, bir ACC değerine karşılık gelmektedir. Yani 4^4 faktöriyel tasarımında yanıt değişkenlerini elde etmek için algoritma $768 * 30 = 23\ 040$ defa çalıştırılmıştır. Ayrıca yöntemin yakınsama davranışı Şekil 3.4’te görüldüğü üzere hızlı bir yakınsama eğiliminde olduğu gözlemlendiğinden, her çalıştırmadaki iterasyon sayısı 100 olarak alınmıştır. Hızlanma parametreleri ise $c_1 = c_2 = 2$ şeklindedir [55]. Deneylerin farklı faktör düzeylerine göre sonuçları EK-1’de verilmiştir. Çizelge 6.2’de istatistiksel olarak anlamlı ve anlamsız olan faktörler ve etkileşimler, F istatistikleri ve karşılık gelen *p*-değeri ile listelenmektedir.

Çizelge 6.2’de görüldüğü gibi transfer fonksiyonu ve eşik değerinin ana etkilerinin $\alpha = 0,01$ düzeyinde istatistiksel olarak anlamlı olduğu görülmektedir ve BPSO algoritmasının performansını etkileyen başlıca temel parametrelerdir. Eşik değeri, transfer fonksiyonu faktörüne göre daha yüksek bir F değerine sahip olduğundan dolayı bu iki faktör arasında en etkili faktör olduğu söylenebilir.

Çizelge 6.2. 4⁴ faktöriyel tasarım sonuçları

Kaynak	Serbestlik Derecesi	KT	KO	F	p
A	3	0,0001	0,0001	0,5621	0,6402
B	3	0,0014	0,0005	5,2427	0,0014
C	3	0,0394	0,0131	151,2853	0,0001
D	3	0,0005	0,0002	2,1022	0,0990
A*B	9	0,0004	0,0001	0,5682	0,8234
A*C	9	0,0004	0,0001	0,5179	0,8619
A*D	9	0,0013	0,0001	1,7225	0,0811
B*C	9	0,0043	0,0005	5,4969	0,0001
B*D	9	0,0009	0,0001	1,1609	0,3180
C*D	9	0,0008	0,0001	0,9924	0,4452
A*B*C	27	0,0014	0,0001	0,6149	0,9374
A*B*D	27	0,0020	0,0001	0,8358	0,7054
A*C*D	27	0,0029	0,0001	1,2212	0,2062
B*C*D	27	0,0027	0,0001	1,1620	0,2635
A*B*C*D	81	0,0071	0,0001	1,0102	0,4597
Hata	512	0,0444	0,0001		
Toplam	767	0,1101			

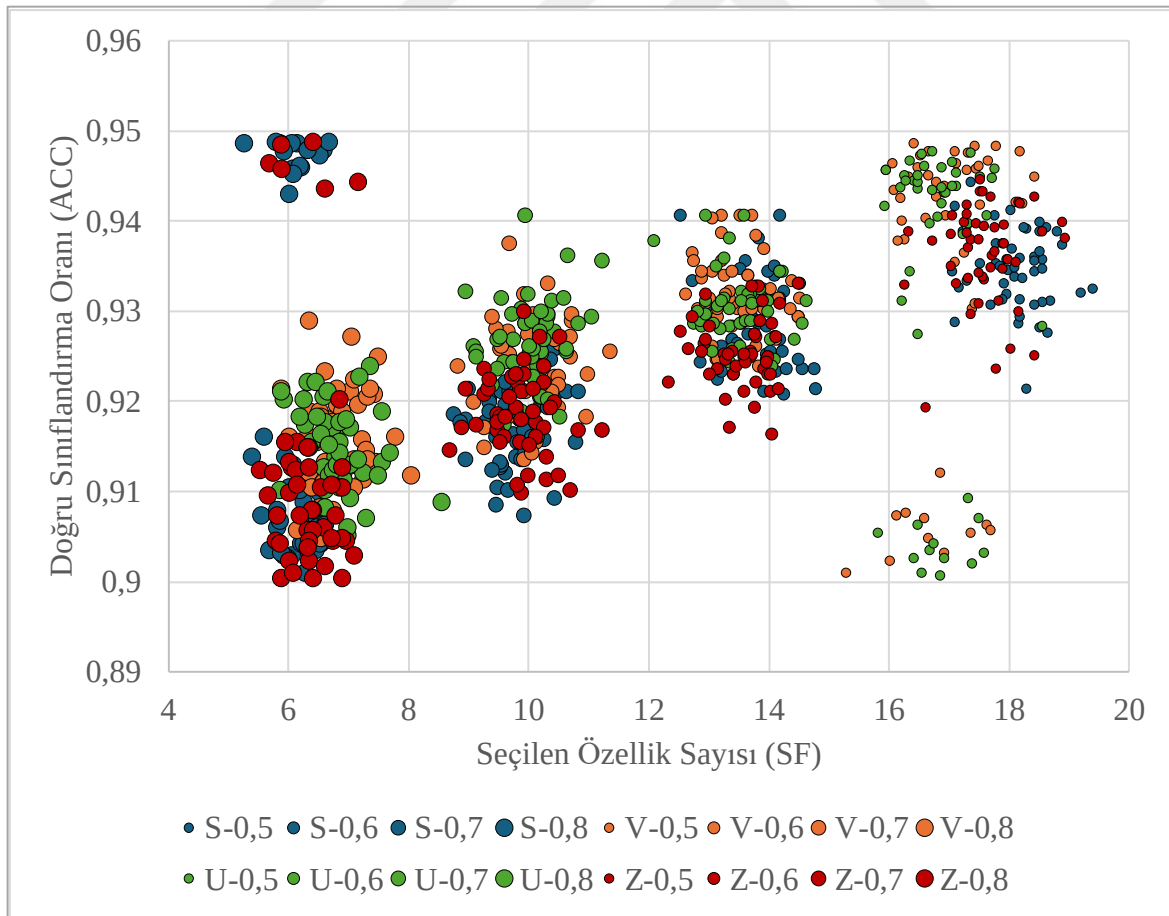
$\alpha = 0,01$ düzeyindeki iki faktörlü etkileşim karşılaştırma sonuçlarına göre, diğer tüm etkileşimlerden yalnızca transfer fonksiyonu ve eşik değeri etkileşiminin etkileri istatistiksel olarak anlamlıdır. Bu sonuçlar ana etkiler için elde edilen sonuçlarla örtüşmektedir. Üç faktörlü ve dört faktörlü etkileşimler arasında $\alpha = 0,01$ düzeyinde istatistiksel olarak anlamlı bir etkileşim bulunmamaktadır.

Yapılan bu çalışmada eşik değerinin, BPSO algoritmasının özellik seçimini ve ACC performansını etkileyen en kritik parametre olduğunu göstermektedir. BPSO algoritmasında eşik değeri rastgele bir sayı olarak kabul edilir. Ancak farklı eşik değerlerinin özellik seçimi ve ACC üzerindeki etkisi bu tez çalışmasında açıkça görülmüştür. Bu nedenle BPSO parametrelerini tanımlarken eşik değerlerine odaklanmak algoritmanın özellik seçimi ve ACC performansını önemli ölçüde etkileyecektir. EK-1’de verilen sonuçlara göre, eşik değeri arttıkça özellik sayısı azalmaktadır. Buna paralel olarak, ACC düşmektedir. Bunun nedeni, modeli tanımlayan özellik sayısının azalması ya da ACC’yi artıracak uygun özelliklerin BPSO algoritması tarafından seçilememesidir. Burada, transfer fonksiyonu da uygun özelliklerin seçilememesinin en önemli nedenlerinden biridir. Uygun olmayan transfer fonksiyonlarının kullanılmasından dolayı özellik seçimi etkilenir. Ayrıca Çizelge

6.2’de verilen sonuçlara göre, transfer fonksiyonunun algoritmanın özellik seçimi ve ACC performansını etkileyen bir faktör olduğu sonucuna varılmıştır.

İstatistiksel olarak anlamsız faktörlerden biri olan kaotik haritanın türü ne olursa olsun ACC’yi etkilememektedir. Bu sonuçtan da anlaşılabacağı üzere ele alınan dört farklı kaotik haritanın birbirlerine göre herhangi bir istatistiksel üstünlüğü bulunmamaktadır. Aynı şekilde parçacık sayısının düşük veya yüksek olmasında da istatistiksel bir etkinin olmadığı gözlemlenmiştir.

Özellik seçimini konu alan çalışmalarda az sayıda özellik seçimi ile yüksek doğruluğa sahip olunması amaçlanmaktadır. Elde edilen sonuçlar incelendiğinde, BPSO algoritmasının performansını etkileyen parametrelerden eşik değeri, seçilen özellik sayısını daha çok etkilerken transfer fonksiyonu ise uygun özelliklerin seçilip ya da seçilememesini daha çok etkilemektedir. Bu durum Şekil 6.1’de ayrıca görselleştirilmiştir.

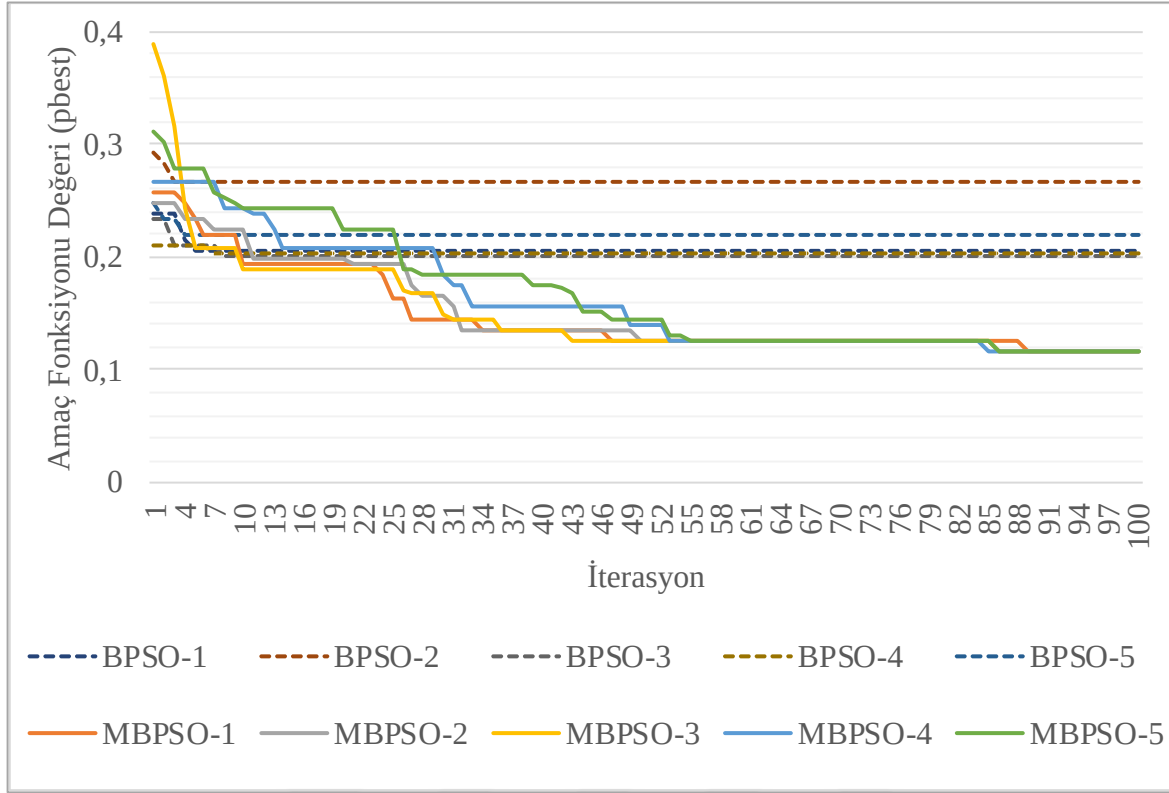


Şekil 6.1. Eşik değeri ve transfer fonksiyonu açısından ACC ve seçilen özellik sayısı

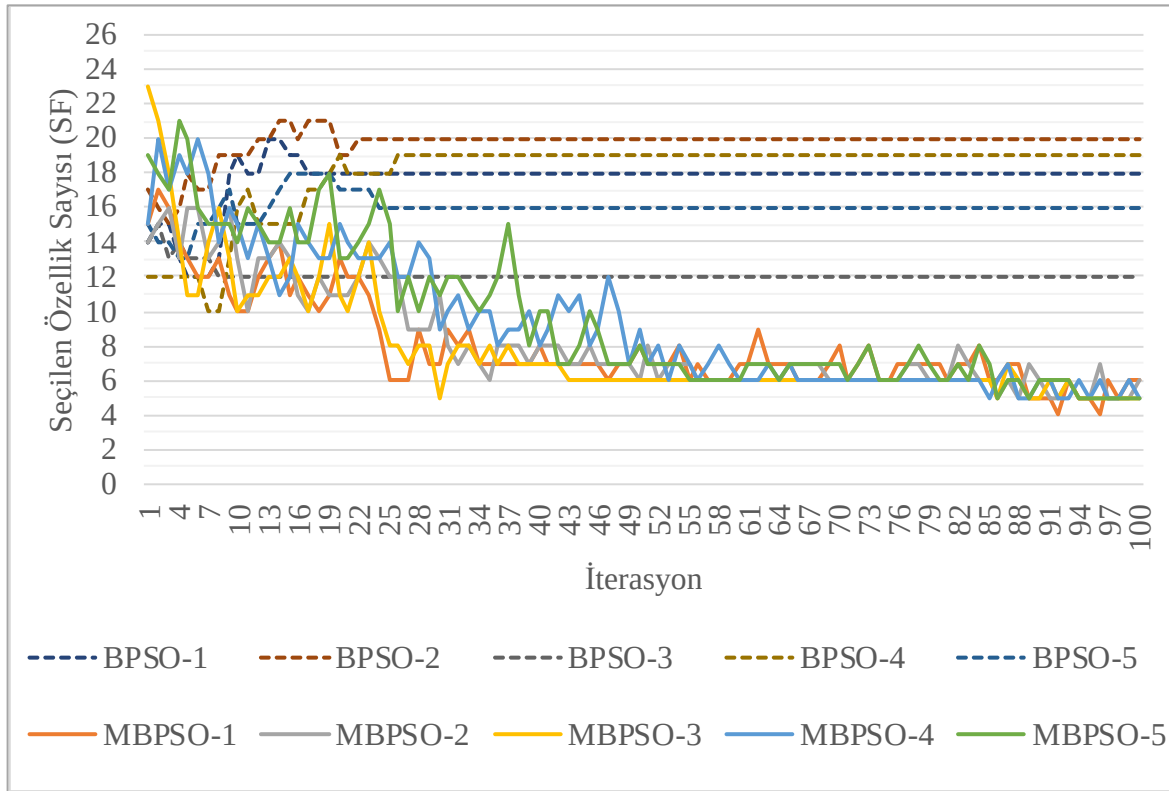
EK-1 ve Şekil 6.1'e göre, eşik değeri arttıkça seçilen özellik sayısının azaldığı net bir şekilde gözükmektedir. Transfer fonksiyonu ise eşik değeri ne olursa olsun seçilen özelliklere bağlı olarak ACC'yi etkilemektedir. Sonuçların çoğunda V şekilli ve U şekilli transfer fonksiyonları eşik değerine bakılmaksızın daha yüksek ACC'ye sahiptir. Buna karşın, eşik değerinin 0,5 olduğu bazı durumlarda V şekilli ve U şekilli fonksiyonlar uygun özellik seçiminde daha az başarılı olmuştur. Eşik değeri 0,8 olduğu bazı durumlarda S şekilli ve Z şekilli fonksiyonlar uygun özellik seçimi ile daha yüksek ACC'ye ulaşmıştır.

BPSO algoritmasını etkileyen parametrelerin 4^4 faktöriyel tasarım yardımıyla etki ve etkileşimleri incelenerek, elde edilen sonuçlar doğrultusunda MBPSO algoritma yaklaşımı önerilmiştir. MBPSO algoritmasında BPSO algoritmasına göre en önemli farklılık, özellik seçimi aşamasında PSO'nun hızının kullanılmamasıdır. Önerilen algoritmada farklı özelliklerin seçilebilme olasılığı artırılarak yerel optimuma takılmanın önlenmesi ve model performansının iyileştirilmesi amaçlanmıştır. BPSO ve MBPSO algoritmalarının rasgele seçilmiş beş parçacığın aynı koşullar altındaki amaç fonksiyon değeri ve seçilen özellik sayısının değişimleri Şekil 6.2 ve Şekil 6.3'te gösterilmiştir. Amaç fonksiyonu değerleri açısından incelendiğinde, BPSO algoritması ilk 10 iterasyondan sonra herhangi bir iyileşme gösteremeyip daha iyi bir sonuca yakınsama gösterememiştir. Bunun aksine MBPSO algoritması 90. iterasyona geldiğinde bile daha iyi bir sonuca yakınsama gösterebilmektedir. Seçilen özellik sayıları açısından incelendiğinde, BPSO algoritmasında başlangıçta seçilen özellik sayısı ya değişmemiştir ya da artmıştır. MBPSO algoritmasındaysa seçilen özellik sayısında azalma meydana gelmiştir. Şekil 6.2 ve Şekil 6.3'te görüleceği üzere, MBPSO algoritması BPSO algoritmasına göre daha iyi bir yakınsama özelliğine sahiptir. Bunun nedeni, farklı özelliklerin seçilmesi sonucunda birbirinden farklı seçilmiş özellik kombinasyonlarının meydana geldiği farklı sınıflandırma modellerinin oluşturulmasıdır.

BPSO ve MBPSO algoritmalarının karşılaştırmalı sonuçlarını incelemek için BPSO algoritmasının temel parametreleri ele alınmıştır. Yani eylemsizlik ağırlık türü lineer azalan ve transfer fonksiyon türü S şekilli olacak şekilde BPSO sonuçları elde edilmiştir. MBPSO algoritmasında farklı olarak S şekilli transfer fonksiyonuna benzemesinden dolayı *tanh* fonksiyonu ele alınmıştır. Amaç fonksiyon ağırlıklandırılması $(\beta; \gamma) = (0,9; 0,1)$ olduğu ve "Ionesphere" verisinin kullanıldığı bu karşılaştırmada farklı parçacık sayısı ve eşik değerlerine göre elde edilen sonuçlar Çizelge 6.3'te verilmiştir.



Şekil 6.2. BPSO ve MBPSO algoritmalarına ilişkin amaç fonksiyon değeri (pbest) değişimlerinin karşılaştırılması

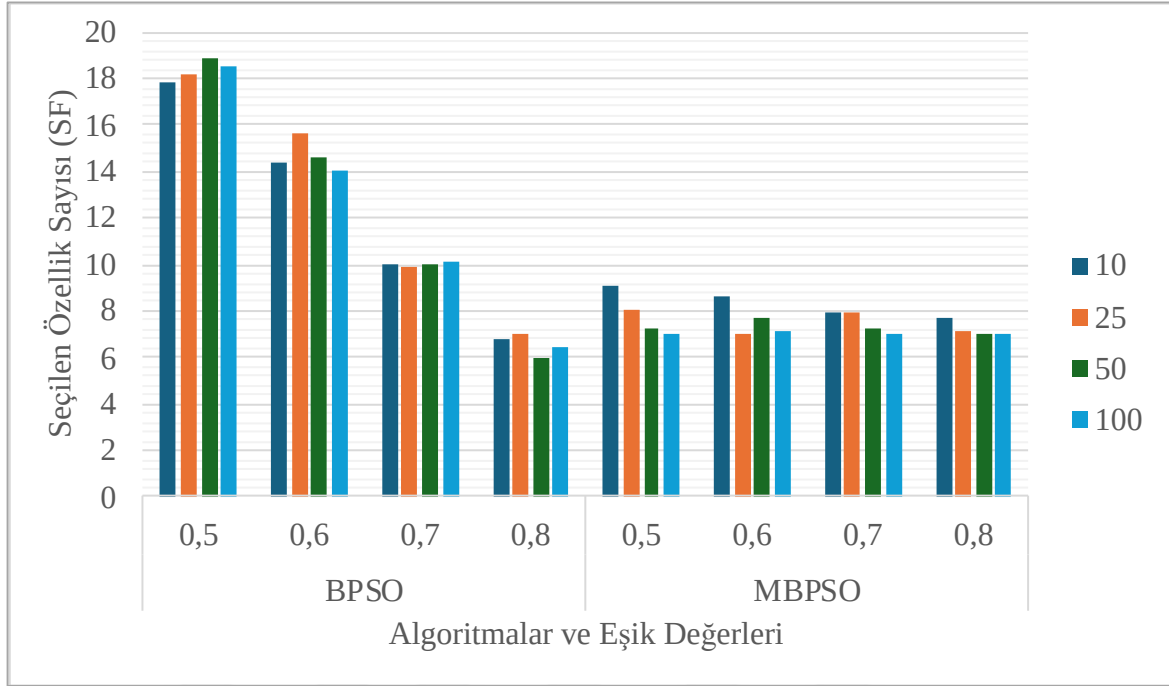


Şekil 6.3. BPSO ve MBPSO algoritmalarına ilişkin seçilen özellik sayısı değişimlerinin karşılaştırılması

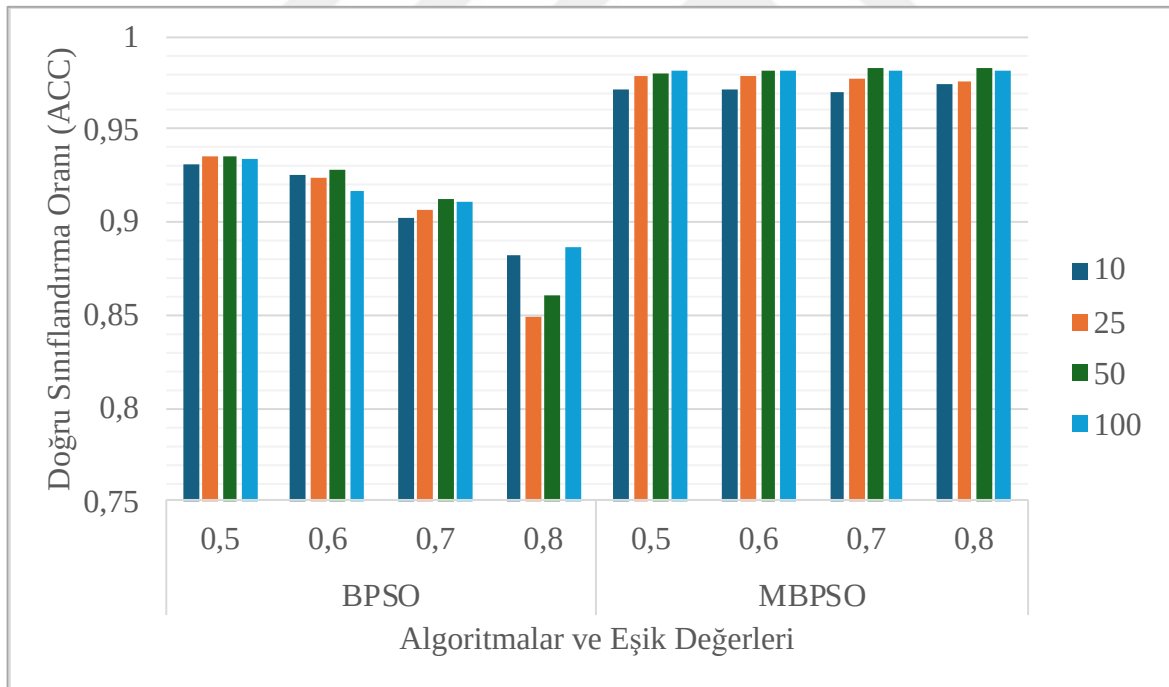
Çizelge 6.3. BPSO ve MBPSO algoritmalarına ilişkin karşılaştırmalı sonuçlar

Algoritma	Eylemsizlik Ağırlığı	Transfer/ Sınırlı Fonksiyon	Parçacık Sayısı	Eşik Değeri							
				0,5		0,6		0,7		0,8	
				ACC	SF	ACC	SF	ACC	SF	ACC	SF
BPSO	Lineer Azalan	S	10	0,9318	17,8667	0,9261	14,4000	0,9031	9,9667	0,8821	6,7667
			25	0,9359	18,2333	0,9245	15,6667	0,9063	9,9333	0,8491	7,0000
			50	0,9359	18,9000	0,9277	14,6667	0,9123	10,0333	0,8613	5,9667
			100	0,9343	18,4667	0,9173	14,0000	0,9113	10,1667	0,8865	6,4667
MBPSO	Lineer Azalan	SF1	10	0,9720	9,0333	0,9711	8,6333	0,9701	7,9333	0,9739	7,7000
			25	0,9786	8,0667	0,9783	7,0667	0,9767	7,9000	0,9755	7,1000
			50	0,9799	7,2667	0,9821	7,6667	0,9827	7,2000	0,9824	6,9667
			100	0,9808	7,0667	0,9818	7,1000	0,9821	7,0333	0,9811	7,0333

Çizelge 6.3'te görüldüğü üzere, MBPSO algoritması eşik değeri 0,8 olduğu durum hariç diğer bütün durumlar için daha az özellik ile daha yüksek ACC sonuçlarına sahip olmuştur. Eşik değeri 0,8 olduğu durumunda seçilen özellik sayısı birbirlerine yakın olmasına rağmen ACC açısından MBPSO algoritması BPSO algoritmasına göre oldukça yüksek çıkmıştır. Yerel optimuma takılması önlenerek BPSO algoritmasına göre daha fazla farklı özellik kombinasyonunu ele alan MBPSO algoritması, daha az özellik sayısı ile modeli daha iyi açıklamıştır. Eşik değeri açısından incelendiğinde, BPSO algoritmasında eşik değeri arttıkça özellik sayısı azalırken buna bağlı olarak ACC'de düşmektedir. Buna karşın, MBPSO algoritmasında eşik değeri arttıkça özellik sayısındaki azalış BPSO algoritmasındaki gibi sert olmamaktadır. Başka bir deyişle, MBPSO algoritmasında farklı eşik değerlerinde seçilen ortalama özellik sayısı birbirine yakinken BPSO algoritmasındaysa bu durum aynı değildir. Bu durum Şekil 6.4'te gösterilmektedir. Ayrıca BPSO algoritmasında gözlemlenen eşik değerinin artışı sonucu düşen ACC, MBPSO algoritmasında gözlemlenmemektedir. Tüm eşik değerleri için birbirine çok yakın ACC sonuçları elde edilmiştir. Bu durum Şekil 6.5'te gösterilmektedir. Parçacık sayısı açısından incelendiğinde, parçacık sayısının artması hem BPSO hem de MBPSO algoritmaları için ACC ve seçilen ortalama özellik sayısını fazla değiştirmemektedir. Önceki analizlerde, BPSO algoritmasını etkileyen parametrelerin 4⁴ faktöriyel tasarım analizi yardımıyla parçacık sayısının düşük veya yüksek olmasında da istatistiksel bir etkinin olmadığı gözlemlenmiştir. Burada da buna paralel bir sonuç elde edilerek sonuçlar desteklenmiştir.



Şekil 6.4. Algoritmalar ve eşik değeri açısından seçilen özellik sayısı



Şekil 6.5. Algoritmaları ve eşik değeri açısından ACC değerleri

EK-1’de verilen farklı kaotik haritalar ve transfer fonksiyonlarına ilişkin parçacık sayısının 100 olduğu durumdaki sonuçlar ile MBPSO algoritmasının sonuçları Çizelge 6.4’te verilmiştir. Önceki sonuçlara paralel olarak, MBPSO algoritması eşik değeri 0,8 olduğu durum hariç diğer bütün durumlar için daha az özellik ile daha yüksek ACC sonuçlarına

sahip olmuştur. Eşik değeri 0,8 olduğu durumda seçilen özellik sayısı birbirlerine yakın olmasına rağmen ACC açısından MBPSO algoritması BPSO algoritmasına göre oldukça yüksek çıkmıştır. Bu sonuçlar doğrultusunda, BPSO algoritmasının performansının geliştirilmesi için literatürde önerilen farklı kaotik haritalar ve transfer fonksiyonları olmasına rağmen MBPSO algoritması bu tez çalışmasında ele alınan yöntemlerden daha iyi sonuçlar vermiştir. Ayrıca “*Ionesphere*” verisine uygulanan $(\beta; \gamma) = (0,9; 0,1)$, $(\beta; \gamma) = (0,5; 0,5)$ ve $(\beta; \gamma) = (0,1; 0,9)$ şeklindeki farklı amaç fonksiyon ağırlıklandırmalarına ait MBPSO algoritma sonuçları sırasıyla EK-2, EK-3 ve EK-4’te verilmiştir.

Çizelge 6.4. BPSO ve MBPSO algoritmalarına ilişkin farklı parametreler açısından karşılaştırmalı sonuçlar

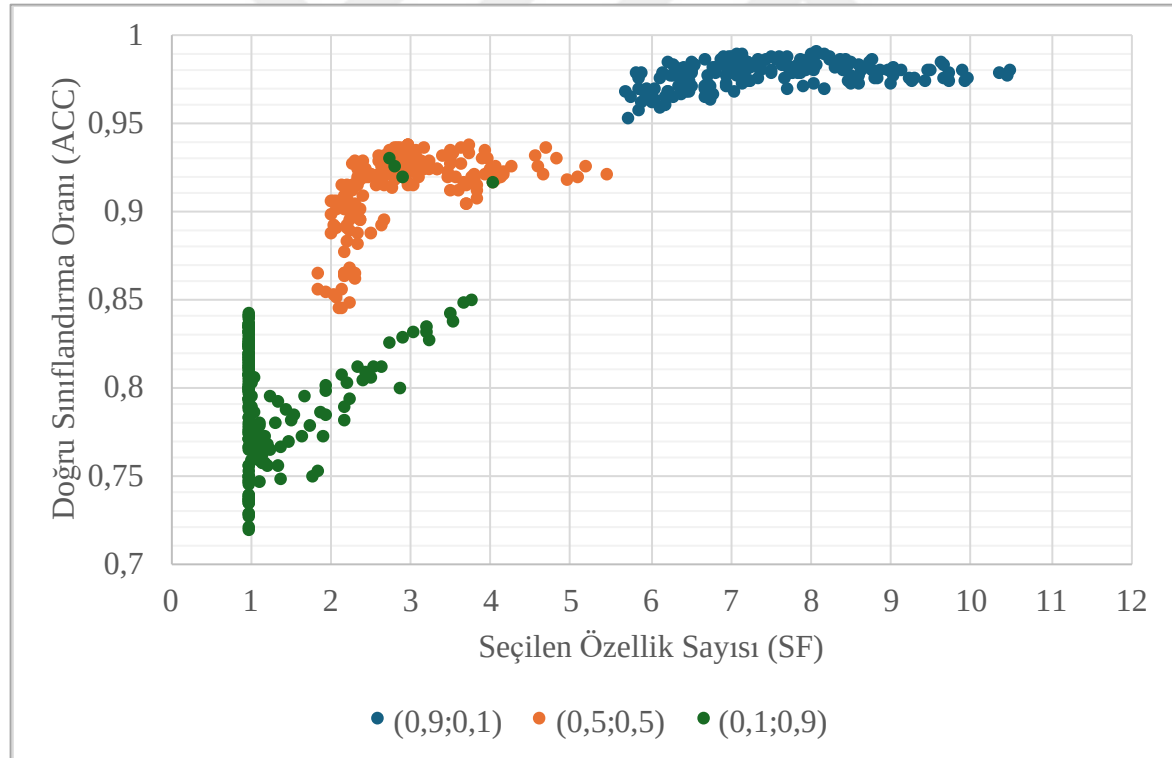
Algoritma	Eylemsizlik Ağırlığı	Transfer/ Sınırlı Fonksiyon	Parçacık Sayısı	Eşik Değeri							
				0,5		0,6		0,7		0,8	
				ACC	SF	ACC	SF	ACC	SF	ACC	SF
BPSO	Circle	S	100	0,9356	18,4666	0,9302	14,0333	0,9148	9,5777	0,9334	5,9444
BPSO	Circle	U	100	0,9306	16,6777	0,9322	13,8333	0,9328	10,0222	0,9156	6,8444
BPSO	Circle	V	100	0,9470	17,6777	0,9299	13,6444	0,9253	10,2222	0,9211	6,8666
BPSO	Circle	Z	100	0,9364	17,6333	0,9258	13,8444	0,9222	9,9888	0,9238	6,2222
BPSO	Logistic	S	100	0,9362	18,1111	0,9273	14,1777	0,9184	9,8333	0,9082	6,3555
BPSO	Logistic	U	100	0,9301	16,8222	0,9336	13,3555	0,9250	9,8888	0,9144	6,8888
BPSO	Logistic	V	100	0,9415	16,8555	0,9318	13,4111	0,9286	10,5888	0,9148	6,6555
BPSO	Logistic	Z	100	0,9380	17,8777	0,9222	13,6444	0,9203	10,0888	0,9053	6,4000
BPSO	Piecewise	S	100	0,9339	18,1777	0,9287	14,2666	0,9193	9,9666	0,9244	6,2333
BPSO	Piecewise	U	100	0,9312	17,2000	0,9284	13,5555	0,9274	10,0000	0,9128	7,1222
BPSO	Piecewise	V	100	0,9455	16,6777	0,9322	13,4444	0,9263	9,7777	0,9089	6,6000
BPSO	Piecewise	Z	100	0,9400	18,1222	0,9252	13,8111	0,9199	10,0777	0,9091	6,2555
BPSO	Singer	S	100	0,9334	18,2888	0,9306	14,2333	0,9171	10,2333	0,9348	5,9000
BPSO	Singer	U	100	0,9304	17,0111	0,9295	13,2111	0,9262	9,7777	0,9156	7,0111
BPSO	Singer	V	100	0,9436	17,1777	0,9374	13,7111	0,9258	9,7888	0,9156	6,5333
BPSO	Singer	Z	100	0,9378	17,8111	0,9244	13,7444	0,9200	10,2111	0,9079	6,1666
MBPSO	Lineer Azalan	SF1	10	0,9720	9,0333	0,9711	8,6333	0,9701	7,9333	0,9739	7,7000
			25	0,9786	8,0667	0,9783	7,0667	0,9767	7,9000	0,9755	7,1000
			50	0,9799	7,2667	0,9821	7,6667	0,9827	7,2000	0,9824	6,9667
			100	0,9808	7,0667	0,9818	7,1000	0,9821	7,0333	0,9811	7,0333
MBPSO	Logistic	SF1	10	0,9730	9,4667	0,9783	8,9000	0,9726	8,5000	0,9686	8,2000
			25	0,9783	8,6667	0,9796	7,9667	0,9799	7,8000	0,9811	7,7667
			50	0,9774	7,7667	0,9786	7,8000	0,9770	7,5333	0,9852	6,9000
			100	0,9818	7,7000	0,9836	7,2333	0,9865	6,9333	0,9843	7,1333

Çizelge 6.5’te verilen sonuçlara göre, amaç fonksiyonu ağırlıkları açısından ACC ve seçilen özelliklerin ortalamaları arasında istatistiksel olarak anlamlı bir farklılık bulunmaktadır ($p < 0,05$). β katsayısı azaldıkça ya da γ katsayısı arttıkça amaç fonksiyonunun yapısından dolayı seçilen özellik sayısının genellikle düştüğü Şekil 6.6’da gösterilmiştir. Çünkü γ

katsayısının artması seçilen özellik sayısının amaç fonksiyonuna etkisini artırmaktadır. Ayrıca özellik sayısının düşmesine bağlı olarak, uygun olmayan özelliklerin seçiminden kaynaklı ACC’de düşüşler meydana gelebilmektedir.

Çizelge 6.5. Amaç fonksiyonu ağırlıkları açısından ortalama ACC, ortalama seçilen özellik sayısı ve test istatistiği

Amaç Fonksiyonu Ağırlıkları	Ortalama (SS.)	Ortalama (SS.)	<i>p</i> -değeri	
	ACC	SF	ACC	SF
$(\beta; \gamma) = (0,9; 0,1)$	0,9770 (0,0072)	7,5100 (1,1042)	<0,0001	<0,0001
$(\beta; \gamma) = (0,5; 0,5)$	0,9135 (0,0212)	2,9646 (0,7273)		
$(\beta; \gamma) = (0,1; 0,9)$	0,7931 (0,0363)	1,3380 (0,6729)		



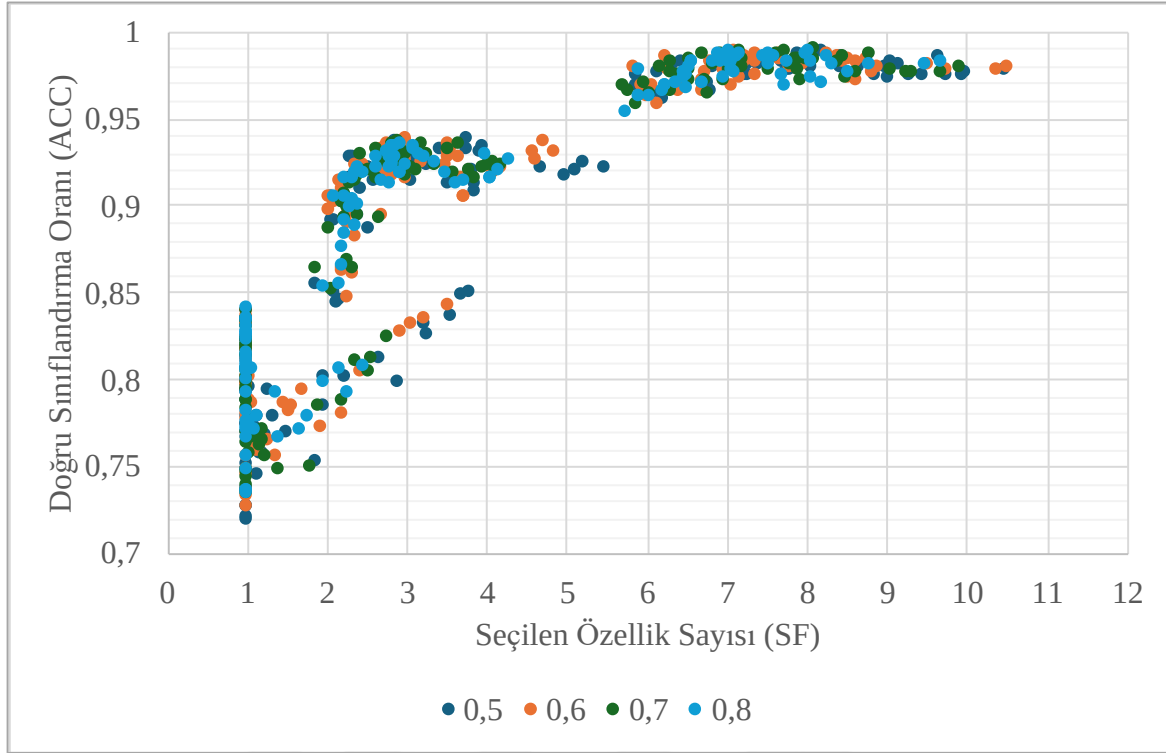
Şekil 6.6. Amaç fonksiyonu ağırlıkları açısından ACC ve seçilen özellik sayısı

Farklı amaç fonksiyonu ağırlıkları için Çizelge 6.6’da ki sonuçlar incelendiğinde, tüm durumlar için eşik değerleri açısından ACC ve seçilen özelliklerin ortalamaları arasında istatistiksel olarak anlamlı bir farklılık bulunmamaktadır ($p>0,05$). Eşik değerinin artması durumunda özelliklerin seçilebilme olasılığı azaltılmaktadır. Ayrıca eşik değerinin BPSO

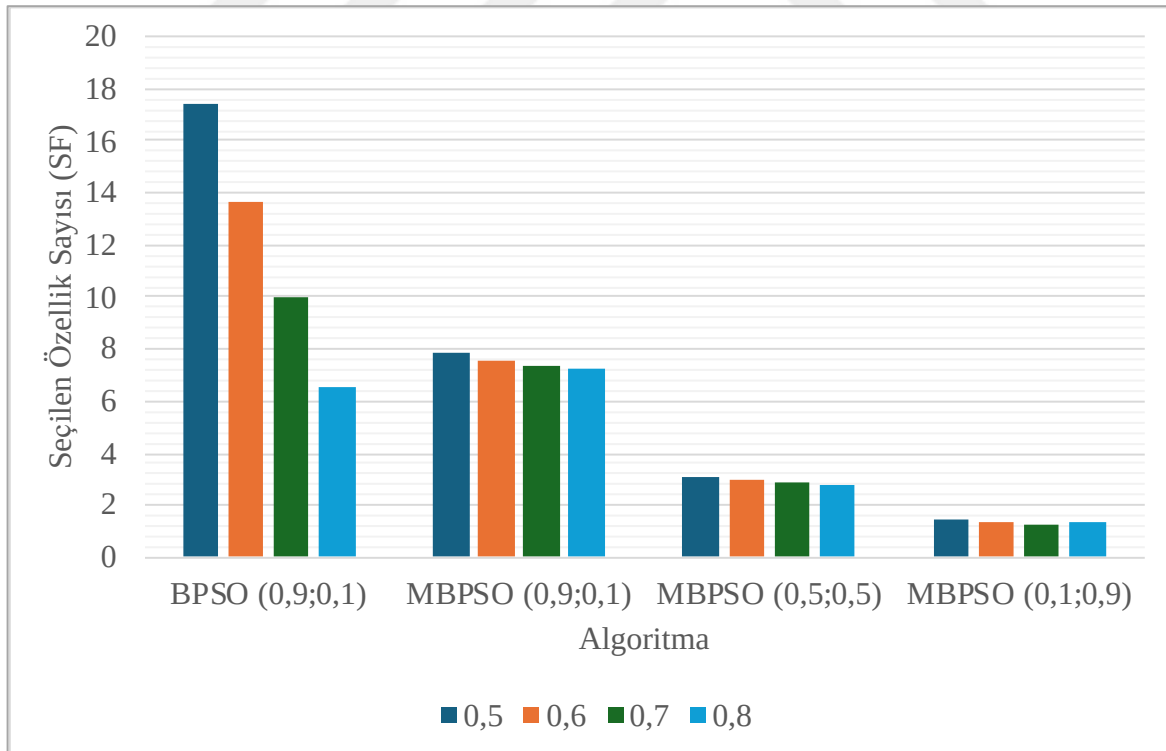
algoritmasının özellik seçimi ve ACC performansını etkileyen en kritik parametre olduğu 4⁴ faktöriyel tasarım sonuçları ile desteklenmiştir. BPSO algoritmasında eşik değerinin artması ile seçilen özelliklerin sayısının azaldığı Şekil 6.1’de görselleştirilmiştir. Önerilen MBPSO algoritmasında, BPSO algoritmasında olduğu gibi eşik değerinin doğrudan seçilen özellik sayısını etkilemediği Şekil 6.7 ve Şekil 6.8’de gösterilmektedir.

Çizelge 6.6. Eşik değeri açısından ortalama ACC, ortalama seçilen özellik sayısı ve test istatistiği

Amaç Fonksiyonu Ağırlıkları	Eşik Değeri	Ortalama (SS.)	Ortalama (SS.)	<i>p</i> -değeri	
		ACC	SF	ACC	SF
$(\beta; \gamma) = (0,9; 0,1)$	0,5	0,9768 (0,0064)	7,8222 (1,2189)	0,8939	0,0526
	0,6	0,9774 (0,0071)	7,6056 (1,1383)		
	0,7	0,9772 (0,0075)	7,4045 (1,0729)		
	0,8	0,9769 (0,0081)	7,2076 (0,8995)		
$(\beta; \gamma) = (0,5; 0,5)$	0,5	0,9128 (0,0232)	3,1146 (0,8989)	0,8834	0,4466
	0,6	0,9138 (0,0220)	3,0264 (0,7605)		
	0,7	0,9142 (0,0202)	2,9056 (0,6167)		
	0,8	0,9131 (0,0200)	2,8118 (0,5736)		
$(\beta; \gamma) = (0,1; 0,9)$	0,5	0,7890 (0,0348)	1,4667 (0,8321)	0,0566	0,3800
	0,6	0,7885 (0,0309)	1,3319 (0,6537)		
	0,7	0,7896 (0,0312)	1,2201 (0,4771)		
	0,8	0,8055 (0,0449)	1,3333 (0,6795)		



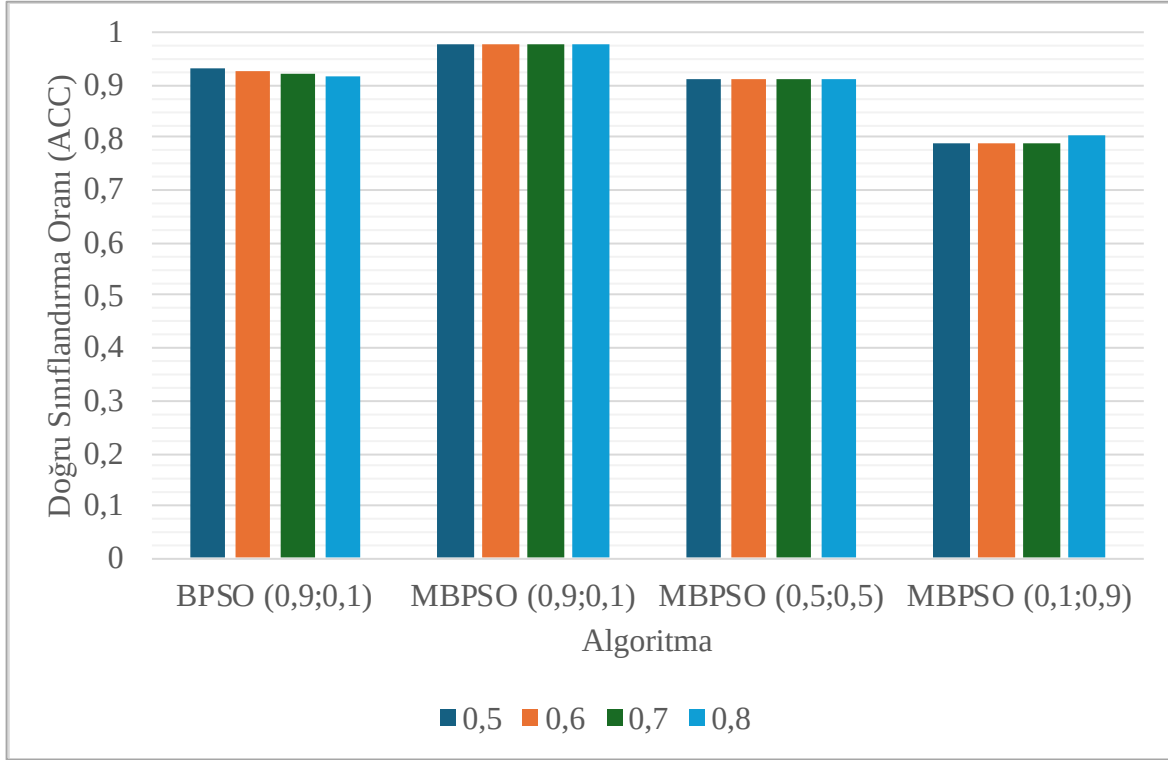
Şekil 6.7. Eşik değeri açısından ACC ve seçilen özellik sayısı



Şekil 6.8. Eşik değeri açısından ortalama seçilen özellik sayısı

Çizelge 6.6’da verilen sonuçlar incelendiğinde, farklı amaç fonksiyonu ağırlıkları için eşik değeri arttıkça ACC’de küçük bir oranda değişimler gözlemlenmektedir. Seçilen ortalama

özellik sayısında ise genellikle küçük bir oranda düşüş gözlemlenmiştir. Eşik değerlerinin artması, BPSO algoritmasında modeli tanımlayan özellik sayısının azalmasına ya da ACC'yi artıracak uygun özelliklerin seçilememesine neden olmaktadır. Buna karşın MBPSO algoritmasında eşik değerlerinden ziyade amaç fonksiyonu ağırlıklarının ACC'yi etkilediği Şekil 6.9'da gösterilmiştir.

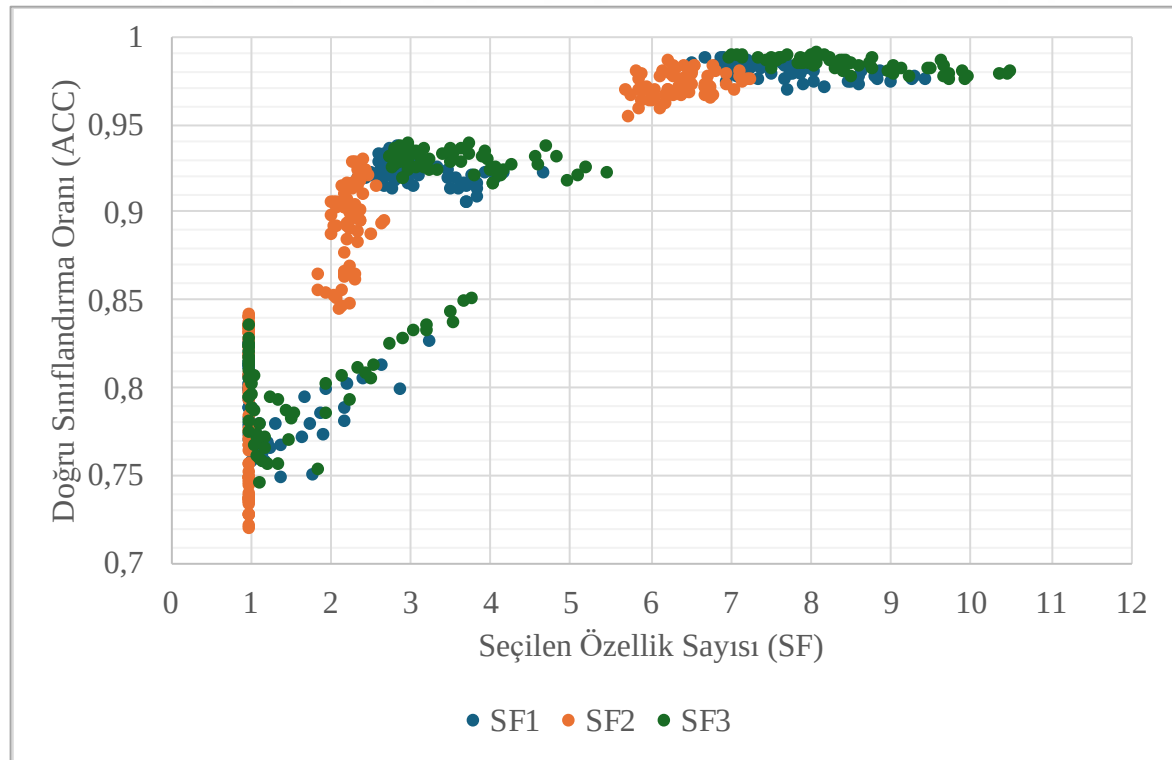


Şekil 6.9. Eşik değerini açısından ACC değerleri

Farklı amaç fonksiyonu ağırlıkları için Çizelge 6.7'de ki sonuçlar incelendiğinde, tüm durumlarda sınırlı fonksiyon türü açısından ACC ve seçilen özelliklerin ortalamaları arasında istatistiksel olarak anlamlı bir farklılık bulunmaktadır ($p < 0,05$). Farklı amaç fonksiyonu ağırlıkları için SF3 fonksiyonu en iyi ACC'yi vermesinin yanında tüm durumlarda seçilen ortalama özellik sayısı diğer türlere göre fazladır. SF2 fonksiyonunda ise bu durumun tam tersi gerçekleşmektedir. Yani SF2 fonksiyonu yardımıyla, tüm durumlarda daha az özellik ile daha düşük ACC elde edilmiştir. Ayrıca $(\beta; \gamma) = (0,1; 0,9)$ durumunda, SF2 fonksiyonu ile yapılan tüm modellerde en iyi çözümü, seçilen bir özellik ile hesaplanan ACC verdiği için seçilen ortalama özellik sayısı 1 olarak elde edilmiştir. Bu durum amaç fonksiyonu ağırlıklandırılmasından kaynaklanmaktadır. Sınırlı fonksiyon türü açısından ACC ve seçilen özellik sayısı sonuçları Şekil 6.10'da görselleştirilmiştir.

Çizelge 6.7. Sınırlı fonksiyon türü açısından ortalama ACC, ortalama seçilen özellik sayısı ve test istatistiği

Amaç Fonksiyonu Ağırlıkları	Sınırlı Fonksiyon	Ortalama (SS.)	Ortalama (SS.)	<i>p</i> -değeri	
		ACC	SF	ACC	SF
$(\beta; \gamma) = (0,9; 0,1)$	SF1	0,9785 (0,0044)	7,6370 (0,7245)	<0,0001	<0,0001
	SF2	0,9704 (0,0070)	6,4117 (0,4020)		
	SF3	0,9822 (0,0040)	8,4813 (0,9001)		
$(\beta; \gamma) = (0,5; 0,5)$	SF1	0,9204 (0,0067)	3,1203 (0,5050)	<0,0001	<0,0001
	SF2	0,8920 (0,0237)	2,2656 (0,1678)		
	SF3	0,9280 (0,0056)	3,5078 (0,7071)		
$(\beta; \gamma) = (0,1; 0,9)$	SF1	0,7900 (0,0210)	1,3021 (0,5289)	0,0028	<0,0001
	SF2	0,7841 (0,0415)	1,0000 (0,0000)		
	SF3	0,8053 (0,0400)	1,7120 (0,9130)		

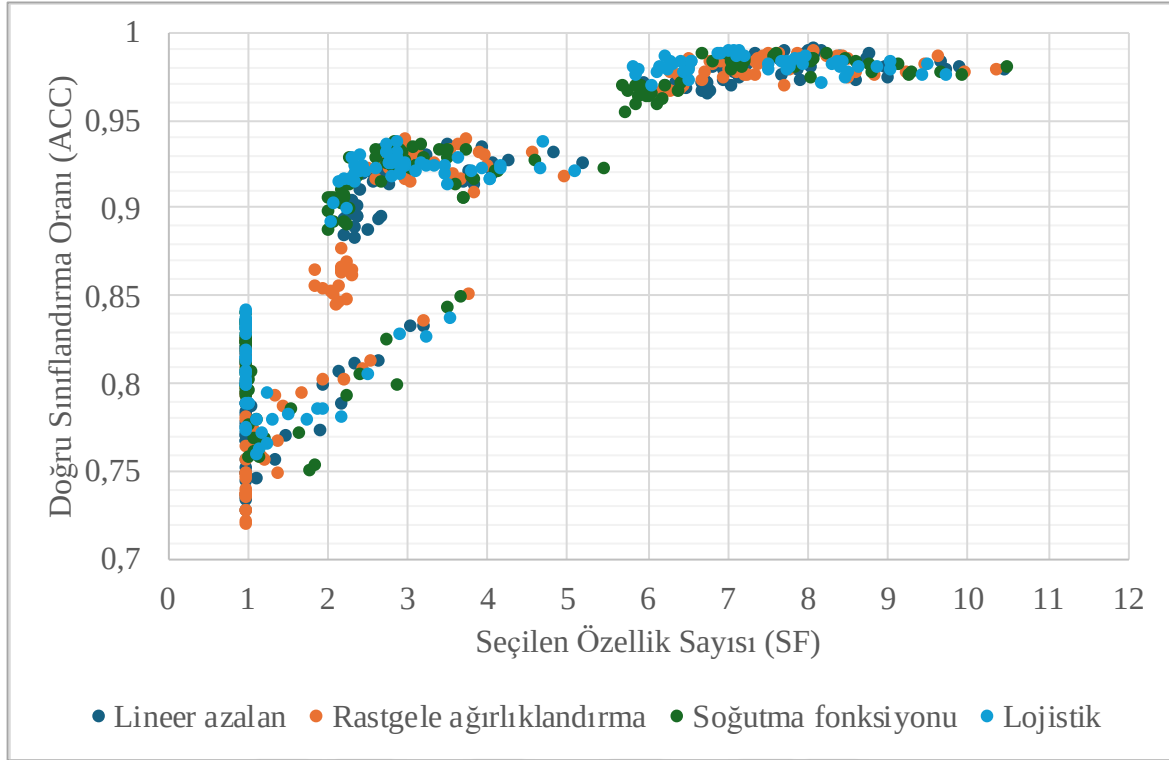


Şekil 6.10. Sınırlı fonksiyon türü açısından ACC ve seçilen özellik sayısı

Farklı amaç fonksiyonu ağırlıkları için Çizelge 6.8’de ki sonuçlar incelendiğinde, $(\beta; \gamma) = (0,1; 0,9)$ durumu için eylemsizlik ağırlığı türü açısından ACC ortalamaları arasında istatistiksel olarak anlamlı bir farklılık bulunurken ($p < 0,05$) seçilen özelliklerin ortalamaları arasında istatistiksel olarak anlamlı bir farklılık bulunmamaktadır ($p > 0,05$). Bundan dolayı, en yüksek ACC değerini veren Lojistik türü $(\beta; \gamma) = (0,1; 0,9)$ durumu için tercih edilebilir. $(\beta; \gamma) = (0,9; 0,1)$ ve $(\beta; \gamma) = (0,5; 0,5)$ durumlarında, ACC ve seçilen özelliklerin ortalamaları arasında istatistiksel olarak anlamlı bir farklılık bulunmamaktadır ($p > 0,05$). Eylemsizlik ağırlığı türünün model sonuçları için herhangi bir anlamlı etkisinin bulunmamasından dolayı bu tez çalışmasında ele alınan herhangi bir eylemsizlik ağırlığı türü MBPSO algoritmasında kullanılabilir. Eylemsizlik ağırlığı türü açısından ACC ve seçilen özellik sayısı sonuçları Şekil 6.11’de görselleştirilmiştir.

Çizelge 6.8. Eylemsizlik ağırlığı türü açısından ortalama ACC, ortalama seçilen özellik sayısı ve test istatistiği

Amaç Fonksiyonu Ağırlıkları	Eylemsizlik Ağırlığı	Ortalama (SS.)	Ortalama (SS.)	<i>p</i> -değeri	
		ACC	SF	ACC	SF
$(\beta; \gamma) = (0,9; 0,1)$	Lineer azalan	0,9769 (0,0072)	7,6233 (1,0351)	0,2027	0,6888
	Rastgele ağırlıklandırma	0,9776 (0,0062)	7,5583 (1,0347)		
	Soğutma fonksiyonu	0,9745 (0,0095)	7,3854 (1,2674)		
	Lojistik	0,9792 (0,0046)	7,4729 (1,0835)		
$(\beta; \gamma) = (0,5; 0,5)$	Lineer azalan	0,9149 (0,0164)	3,0007 (0,6984)	0,7525	0,6827
	Rastgele ağırlıklandırma	0,9024 (0,0336)	2,8833 (0,7311)		
	Soğutma fonksiyonu	0,9165 (0,0138)	2,9556 (0,7575)		
	Lojistik	0,9200 (0,0083)	3,0188 (0,7366)		
$(\beta; \gamma) = (0,1; 0,9)$	Lineer azalan	0,7806 (0,0288)	1,2715 (0,5727)	<0,0001	0,3891
	Rastgele ağırlıklandırma	0,7774 (0,0354)	1,2757 (0,6104)		
	Soğutma fonksiyonu	0,8019 (0,0277)	1,3292 (0,6724)		
	Lojistik	0,8126 (0,0405)	1,4757 (0,8121)		

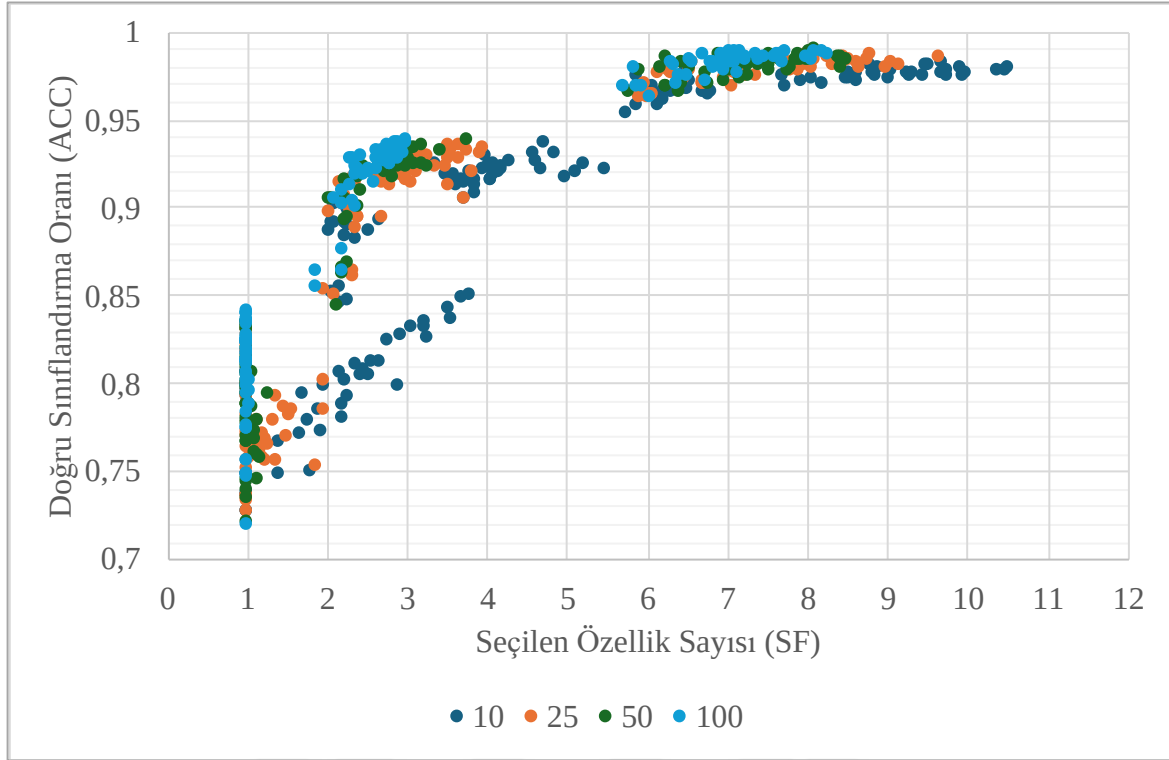


Şekil 6.11. Eylemsizlik ağırlığı türü açısından ACC ve seçilen özellik sayısı

Farklı amaç fonksiyonu ağırlıkları için Çizelge 6.9’da ki sonuçlar incelendiğinde, tüm durumlar için parçacık sayısı açısından ACC ve seçilen özelliklerin ortalamaları arasında istatistiksel olarak anlamlı bir farklılık bulunmaktadır ($p < 0,05$). Farklı amaç fonksiyonu ağırlıkları için parçacık sayısının artması ile seçilen ortalama özellik sayısında bir düşüş meydana gelmektedir. Fakat $(\beta; \gamma) = (0,1; 0,9)$ durumu hariç diğer durumlarda ortalama ACC arasında çok büyük farklılıklar bulunmamaktadır. $(\beta; \gamma) = (0,1; 0,9)$ durumu için parçacık sayısı 10 ve 50 iken birbirine çok yakın ACC elde edilmiş olup, parçacık sayısı 100 iken ACC’de küçük bir artış meydana gelmiştir. Parçacık sayısı 25 olduğunda ise en düşük ACC elde edilmiştir. Bu sonuçlara göre, yüksek parçacık sayısı ile çalışmak problemin amacına göre daha iyi sonuçlar elde edilmesini sağlayacaktır. Parçacık sayısı açısından ACC ve seçilen özellik sayısı sonuçları Şekil 6.12’de görselleştirilmiştir.

Çizelge 6.9. Parçacık sayısı açısından ortalama ACC, ortalama seçilen özellik sayısı ve test istatistiği

Amaç Fonksiyonu Ağırlıkları	Parçacık Sayısı	Ortalama (SS.)	Ortalama (SS.)	<i>p</i> -değeri	
		ACC	SF	ACC	SF
$(\beta; \gamma) = (0,9; 0,1)$	10	0,9713 (0,0065)	8,1993 (1,4979)	<0,0001	<0,0001
	25	0,9768 (0,0064)	7,5382 (1,0105)		
	50	0,9791 (0,0060)	7,2812 (0,7178)		
	100	0,9810 (0,0064)	7,0212 (0,6256)		
$(\beta; \gamma) = (0,5; 0,5)$	10	0,9064 (0,0225)	3,5403 (1,0278)	<0,0001	<0,0001
	25	0,9120 (0,0209)	2,9681 (0,5792)		
	50	0,9159 (0,0203)	2,7389 (0,3826)		
	100	0,9196 (0,0196)	2,6111 (0,3055)		
$(\beta; \gamma) = (0,1; 0,9)$	10	0,7943 (0,0373)	2,0354 (0,9459)	<0,0001	<0,0001
	25	0,7779 (0,0336)	1,2125 (0,3591)		
	50	0,7917 (0,0354)	1,0653 (0,2669)		
	100	0,8087 (0,0332)	1,0389 (0,2548)		



Şekil 6.12. Parçacık sayısı açısından ACC ve seçilen özellik sayısı

Farklı veri setleri için BPSO ve MBPSO sonuçlarının karşılaştırılması amacıyla literatürde sıkça kullanılan “*BreastEW*”, “*Glass*”, “*Ionesphere*”, “*Sonar*”, “*Vehicle*”, “*Wine*” ve “*Zoo*” veri setlerinin gözlem sayısı, özellik sayısı, sınıf sayısı ve sınıfta yer alan frekans sayıları Çizelge 6.10’da verilmiştir [81].

Çizelge 6.10. Veri setlerinin yapılarına ilişkin bilgiler

Veri Adı	Gözlem Sayısı	Özellik Sayısı	Sınıf Sayısı	Sınıf Frekans Sayısı
BreastEW	569	30	2	{357, 212}
Glass	214	9	6	{70, 76, 17, 13, 9, 29}
Ionesphere	351	34	2	{126, 225}
Sonar	208	60	2	{111, 97}
Vehicle	846	18	4	{218, 212, 217, 199}
Wine	178	13	3	{59, 71, 48}
Zoo	101	16	7	{41, 20, 5, 13, 4, 8, 10}

Çizelge 6.11 ve Çizelge 6.12’de sırasıyla farklı performans ölçülerine göre lineer azalan ve lojistik eylemsizlik türleri için BPSO ve MBPSO algoritmalarının karşılaştırmalı sonuçları incelenmiştir. Ayrıca BPSO algoritması için transfer fonksiyonu S şekilli ve V şekilli olarak seçilmiş olup, bu fonksiyonlara benzemesinden dolayı MBPSO algoritması için SF1 ve SF2

sınırlı fonksiyonları tercih edilmiştir. Amaç fonksiyon ağırlıklandırılması $(\beta; \gamma) = (0,9; 0,1)$ ve parçacık sayısı 100 olarak alınmıştır. BPSO algoritmasında en iyi ACC değerlerinin elde edilmesi ve MBPSO algoritması için eşik değerleri açısından ACC ortalamaları arasında istatistiksel olarak anlamlı bir farklılık bulunmaması nedeniyle karşılaştırmada kullanılan eşik değeri 0,5 olarak alınmıştır. İterasyon sayısı 100 olacak şekilde algoritmaların 10 kez çalıştırılması ile elde edilen ortalama değerler Çizelge 6.11 ve Çizelge 6.12’de verilmiştir.

Çizelge 6.11. Farklı performans ölçülerine göre lineer azalan eylemsizlik türü için BPSO ve MBPSO sonuçlarının karşılaştırılması

Performans Ölçüsü	Veri Adı	Transfer / Sınırlı Fonksiyon							
		BPSO (S)		MBPSO (SF1)		BPSO (V)		MBPSO (SF2)	
		ACC	SF	ACC	SF	ACC	SF	ACC	SF
ACC	BreastEW	0,8515	17,20	0,9304	3,40	0,8246	14,00	0,9082	3,10
	Glass	0,3923	4,80	0,6585	3,20	0,4077	4,30	0,6523	3,10
	Ionesphere	0,9302	18,50	0,9840	7,10	0,9245	17,70	0,9745	7,00
	Sonar	0,7333	31,50	0,9238	12,00	0,7667	30,80	0,9000	11,70
	Vehicle	0,4713	11,00	0,6894	4,70	0,4724	10,00	0,6772	3,70
	Wine	0,5000	6,80	0,9556	4,00	0,5870	6,30	0,9444	3,90
	Zoo	0,8613	8,20	0,9774	5,50	0,8581	7,20	0,9935	5,40
F1	BreastEW	0,8196	16,90	0,9134	4,10	0,8244	16,40	0,9031	1,40
	Glass	0,1509	4,80	0,4046	2,50	0,1603	4,40	0,4065	2,30
	Ionesphere	0,9184	16,70	0,9855	7,40	0,9245	15,50	0,9728	7,30
	Sonar	0,7202	31,90	0,9276	14,20	0,7426	30,80	0,8967	12,10
	Vehicle	0,4299	10,60	0,6802	4,90	0,4438	10,10	0,6667	3,90
	Wine	0,4251	7,60	0,9595	4,00	0,5184	7,20	0,9443	3,10
	Zoo	0,6313	8,60	0,9658	5,20	0,6672	7,40	0,9921	5,10
MCC	BreastEW	0,6455	14,70	0,8402	5,60	0,7068	13,90	0,8218	4,90
	Glass	0,1219	4,50	0,5061	3,40	0,1964	4,00	0,5074	3,20
	Ionesphere	0,8302	17,80	0,9615	8,50	0,8926	17,40	0,9510	8,40
	Sonar	0,4964	31,50	0,8346	15,00	0,5341	30,00	0,7831	14,20
	Vehicle	0,3212	10,90	0,6084	5,00	0,3158	8,60	0,5850	4,40
	Wine	0,4740	7,10	0,9343	3,80	0,3692	6,60	0,9314	3,60
	Zoo	0,7577	7,40	0,9233	5,70	0,8319	7,10	0,9875	5,00

Çizelge 6.12. Farklı performans ölçülerine göre lojistik eylemsizlik türü için BPSO ve MBPSO sonuçlarının karşılaştırılması

Performans Ölçüsü	Veri Adı	Transfer / Sınırlı Fonksiyon							
		BPSO (S)		MBPSO (SF1)		BPSO (V)		MBPSO (SF2)	
		ACC	SF	ACC	SF	ACC	SF	ACC	SF
ACC	BreastEW	0,8433	17,20	0,9263	3,70	0,8649	15,20	0,9240	2,90
	Glass	0,3800	4,20	0,6477	2,80	0,4138	4,00	0,6354	2,70
	Ionesphere	0,9217	19,00	0,9792	7,30	0,9528	18,00	0,9774	5,70
	Sonar	0,7429	32,80	0,9111	12,70	0,7317	28,00	0,9206	7,70
	Vehicle	0,4685	9,50	0,7016	4,90	0,4909	9,40	0,6850	4,40
	Wine	0,5370	7,10	0,9426	3,80	0,6278	6,60	0,9407	3,70
	Zoo	0,8548	7,90	0,9871	4,80	0,8226	7,80	0,9839	4,20
F1	BreastEW	0,8103	16,70	0,9191	4,70	0,8491	13,80	0,9212	2,50
	Glass	0,1475	4,40	0,4087	2,80	0,1649	4,30	0,4076	2,50
	Ionesphere	0,9129	18,20	0,9824	7,80	0,9209	15,50	0,9812	6,30
	Sonar	0,7101	32,20	0,9226	14,10	0,7415	31,40	0,9029	7,00
	Vehicle	0,4365	10,00	0,6736	5,00	0,4583	7,70	0,6733	4,50
	Wine	0,4390	6,70	0,9402	4,00	0,5285	5,90	0,9543	3,50
	Zoo	0,5916	8,40	0,9718	5,70	0,7039	7,50	0,9802	5,10
MCC	BreastEW	0,6816	16,40	0,8240	4,20	0,6652	13,60	0,8645	3,70
	Glass	0,1158	4,90	0,5215	3,00	0,1616	4,40	0,4867	2,50
	Ionesphere	0,8713	18,30	0,9591	8,40	0,8417	17,80	0,9777	7,90
	Sonar	0,5040	31,50	0,8562	14,10	0,5672	31,10	0,8297	8,70
	Vehicle	0,3087	8,70	0,6216	5,40	0,3236	8,40	0,5992	4,30
	Wine	0,3501	8,50	0,9318	3,70	0,3967	6,20	0,9073	3,40
	Zoo	0,8055	7,70	0,9672	5,30	0,8073	7,50	0,9958	3,70

Farklı performans ölçülerine göre lineer azalan ve lojistik eylemsizlik türlerinin ayrı ayrı ele alındığı tüm veri setleri için MBPSO algoritmasında BPSO algoritmasına göre daha az seçilen özellik sayısı ile daha yüksek ACC elde edilmiştir. Yani önerilen MBPSO algoritmasının farklı veri setleri için BPSO algoritmasına göre hem ACC hem de seçilen özellik sayısı bakımından daha iyi olduğu sonucuna varılmıştır. Transfer ve sınırlı fonksiyonlar açısından incelendiğinde, V şekilli transfer fonksiyonunun ele alındığı BPSO algoritmasında tüm veri setleri için seçilen özellik sayısı S şekilli transfer fonksiyonunun ele alındığı BPSO algoritmaya göre daha azdır. Benzer şekilde SF2 sınırlı fonksiyonunun ele alındığı MBPSO algoritmasında tüm veri setleri için seçilen özellik sayısı SF1 sınırlı fonksiyonunun ele alındığı MBPSO algoritma göre daha azdır.



7. SONUÇ VE ÖNERİLER

Bilim ve teknolojinin gelişmesi büyük miktarda veri toplanmasına ve bunun neticesinde büyük boyutlu veri kümelerinin hızla büyümesine neden olmuştur. Buna bağlı olarak veri madenciliği uygulamaları da paralel bir şekilde artmaya başlamıştır. Fakat büyük boyutlu veri kümelerinde yer alan alakasız ve gereksiz birçok özellik, veri madenciliği çalışmalarının optimizasyon sürecinin kalitesini etkileyip öğrenme algoritmaların performansında ciddi sorunlara neden olabilir. Öğrenme algoritmaları yardımıyla büyük boyutlu verilerde var olan bilginin elde edilmesi için model kurulması işleminden önce bu verilerde yer alan alakasız ve gereksiz özellikler çıkartılarak boyut indirgeme işlemi yapılabilir. Bu işlem doğrultusunda, öğrenme algoritmasının performansı artırılarak hesaplama zorluğundan kaçınılabilir. Boyut indirgeme işleminde kullanılan yaklaşımlardan birisi olan özellik seçimi, bu tez çalışmasında ele alınmıştır. Özellik seçimi yaklaşımı, öğrenme sürecine büyük ölçüde etkisi olmayan özelliklerin yanı sıra güçlü bir şekilde ilişkili olan gereksiz özellikleri ortadan kaldırarak, model performansını artıracak önemli özelliklerin bir alt kümesini seçmektir.

Özellik seçimi ile ilgili yapılan çalışmalar göstermiştir ki, maksimum doğrulukla bir öğrenme modeli oluşturmak için bir dizi optimal özellik alt kümesini belirlemek zor bir problemdir. Bu tarz çözülmesi ve doğrulanması zor optimizasyon problemlerin üstesinden gelmek için öğrenme temellerine dayalı yöntemlere ek olarak meta-sezgisel yöntemler önerilmiştir ve optime yakın bir özellik kümesi bulunmaya çalışılmıştır. Bu tez çalışmasında ele alınan ve meta-sezgisel yöntemlerden biri olan BPSO algoritması optimal özellik alt kümesinin belirlenmesinde önemli bir rol oynamaktadır. BPSO algoritması, optimal çözümü bulmak yerine kabul edilebilir bir sürede tatmin edici yaklaşık çözümler bulabilen bir yaklaşımdır. Çünkü orijinal özellik sayısı n olmak üzere, en alakalı ve önemli özellikleri bulmak için tüm olası alt kümeler dahil toplam 2^n kadar özellik alt kümesi oluşturulması gerekmektedir. Uygun özelliklerin bulunmasını sağlayan bu şekildeki kapsamlı bir arama orta büyüklükteki veri kümeleri için bile hesaplama açısından mümkün değildir.

BPSO algoritmasında yer alan eylemsizlik ağırlığı, transfer fonksiyon türü, eşik değeri ve parçacık sayısı özellik seçimini etkileyebileceği düşünülen önemli parametrelerdir.

Literatürde bu parametreleri ayrı ayrı ele alan çalışmalar mevcut olmasına rağmen birlikte değerlendirildiği bir çalışma bilindiği kadarıyla bulunmamaktadır. Ayrıca yapılan çalışmaların büyük birçoğunda görülen bir diğer eksiklik, ele alınan parametre değerlerinin herhangi bir istatistiksel yöntem ile belirlenmesinden ziyade rastgele belirlenmesidir.

Bu tez çalışmasında, bahsedilen eksikliklerin giderilmesi amacıyla BPSO parametrelerinin özellik seçimindeki etkilerini ve birbirleri ile olan etkileşimlerini incelemek için 4^4 faktöriyel tasarım modeli oluşturulmuştur. Faktöriyel tasarımındaki ana amaç, ele alınan parametrelerin etkilerini ve bu parametreler arasındaki etkileşimlerin istatistiksel olarak anlamlı olup olmadığının test edilmesidir. Analiz sonucunda transfer fonksiyonu ve eşik değeri özellik seçimini etkileyen parametreler olduğu sonucuna ulaşılmıştır. Ayrıca, en çok etkiye sahip eşik değeri büyüdükçe seçilen özellik sayısı azalmaktayken buna bağlı olarak ACC'de düşmektedir. Fakat bu durum minimum sayıda özellik ile maksimum ACC'ye sahip olma amacımıza uymamaktadır. Ele alınan transfer fonksiyonlarının, modeli daha iyi açıklayan özellik alt kümelerini seçememesi nedeniyle BPSO algoritması yerel optimum noktasına takılmakta ve farklı özellik alt küme durumlarını inceleyememektedir.

Yerel optimum noktasına takılma probleminin çözümü için BPSO algoritmasının çalışma mekanizması güncellenerek MBPSO algoritması bu tez çalışmasında önerilmiştir. Önerilen algoritmada farklı özelliklerin seçilebilme olasılığı artırılarak yerel optimuma takılmanın önlenmesi ve model performansının iyileştirilmesi amaçlanmıştır. BPSO ve MBPSO algoritmaları amaç fonksiyonu değerleri açısından karşılaştırıldığında, BPSO algoritması ilk iterasyonlardan sonra herhangi bir iyileşme gösteremeyip daha iyi bir sonuca yakınsama gösterememiştir. Bunun aksine MBPSO algoritması daha ileri iterasyonlara geldiğinde bile daha iyi bir sonuca yakınsama gösterebilmektedir. Seçilen özellik sayıları açısından incelendiğinde, BPSO algoritmasında başlangıçta seçilen özellik sayısı ya değişmemiştir ya da artmıştır. MBPSO algoritmasında ise seçilen özellik sayısında azalma meydana gelmiştir. Dolayısıyla MBPSO algoritmasının BPSO algoritmasına göre daha iyi bir yakınsama özelliğine sahip olduğu sonucuna varılır. Bunun nedeni, farklı özelliklerin seçilmesi sonucunda birbirinden farklı seçilmiş özellik kombinasyonlarının meydana geldiği farklı sınıflandırma modellerinin oluşturulmasıdır.

BPSO ve MBPSO algoritmalarının karşılaştırılmasından çıkan diğer bir sonuç, MBPSO algoritmasında daha az özellik sayısı ile daha yüksek ACC değerleri elde edilmiştir. BPSO

algoritmasında eşik değeri arttıkça özellik sayısı azalırken buna bağlı olarak ACC'de düşmektedir. Buna karşın, MBPSO algoritmasında farklı eşik değerlerinde seçilen ortalama özellik sayısı birbirine yakın olup aynı şekilde ACC değerleri de birbirine çok yakındır. Yani eşik değeri BPSO algoritmasını etkilediği kadar MBPSO algoritmasını etkilememektedir. Ayrıca BPSO algoritmasının farklı kaotik haritalar ve transfer fonksiyonlarına ilişkin durumları ile MBPSO algoritması karşılaştırıldığında, önceki sonuçlara benzer şekilde MBPSO algoritması daha az özellik ile daha yüksek ACC sonuçlarına sahip olmuştur. Bu sonuçlar doğrultusunda, BPSO algoritmasının performansının geliştirilmesi için literatürde önerilen farklı kaotik haritalar ve transfer fonksiyonları olmasına rağmen MBPSO algoritması bu tez çalışmasında ele alınan yöntemlerden daha iyi sonuçlar vermiştir.

MBPSO algoritması ile elde edilen sonuçlar istatistiksel açıdan incelendiğinde, amaç fonksiyonu ağırlıkları, sınırlı fonksiyon türü ve parçacık sayısı açısından ACC ortalamaları arasında istatistiksel olarak anlamlı bir farklılık bulunmaktadır. Ayrıca $(\beta, \gamma) = (0,1; 0,9)$ durumunda da eylemsizlik ağırlığı türü açısından ACC ortalamaları arasında istatistiksel olarak anlamlı bir farklılık bulunmaktadır. Amaç fonksiyonu ağırlıkları, sınırlı fonksiyon türü ve parçacık sayısı açısından seçilen özelliklerin ortalamaları arasında istatistiksel olarak anlamlı bir farklılık bulunmaktadır.

Önerilen MBPSO algoritmasının farklı veri setleri için vereceği sonuçları incelemek için literatürde sıkça kullanılan veri setleri ele alınmış ve BPSO algoritması ile sonuçları karşılaştırılmıştır. Farklı performans ölçüleri, eylemsizlik türleri, transfer ve sınırlı fonksiyon türleri açısından ayrı ayrı ele alınan tüm veri setleri için MBPSO algoritmasında BPSO algoritmasına göre daha az seçilen özellik sayısı ile daha yüksek ACC elde edilmiştir. Yani önerilen MBPSO algoritmasının farklı veri setleri için BPSO algoritmasına göre hem ACC hem de seçilen özellik sayısı bakımından daha iyi olduğu sonucuna varılmıştır.

Değerlendirilen tüm farklı durumlara ilişkin sonuçlar doğrultusunda bu tez çalışmasında önerilen MBPSO algoritmasının, literatürde yapılmış olan ve BPSO parametrelerini konu alan farklı BPSO çalışmalarından daha iyi sonuçlar verdiği oldukça açıktır. Bu çalışma, BPSO parametreleri üzerinde yapılan çalışmalar özellik seçiminin temel amacını tam olarak geliştiremediğini göstermiş olup, gerekli değişikliğin BPSO algoritmasının yapısında meydana gelmesi gerektiğini göstermiştir.



KAYNAKLAR

1. Tang, X., Dai, Y., ve Xiang, Y. (2019). Feature selection based on feature interactions with application to text categorization. *Expert Systems with Applications*, 120, 207-216.
2. Wang, H., Zhang, Y., Zhang, J., Li, T., ve Peng, L. (2019). A factor graph model for unsupervised feature selection. *Information Sciences*, 480, 144-159.
3. Liu, Y., Nie, F., Gao, Q., Gao, X., Han, J., ve Shao, L. (2019). Flexible unsupervised feature extraction for image classification. *Neural Networks*, 115, 65-71.
4. Labani, M., Moradi, P., Ahmadizar, F., ve Jalili, M. (2018). A novel multivariate filter method for feature selection in text classification problems. *Engineering Applications of Artificial Intelligence*, 70, 25-37.
5. Zhang, X., Wu, G., Dong, Z., ve Crawford, C. (2015). Embedded feature-selection support vector machine for driving pattern recognition. *Journal of the Franklin Institute*, 352(2), 669-685.
6. Santucci, V., Baiocchi, M., ve Milani, A. (2020). An algebraic framework for swarm and evolutionary algorithms in combinatorial optimization. *Swarm and Evolutionary Computation*, 55, 100673.
7. Li, J., Chen, G., Li, M., ve Chen, H. (2020). An enhanced-indicator based many-objective evolutionary algorithm with adaptive reference point. *Swarm and Evolutionary Computation*, 55.
8. Zhang, S., Lee, C. K. M., Chan, H. K., Choy, K. L., ve Wu, Z. (2015). Swarm intelligence applied in green logistics: A literature review. *Engineering Applications of Artificial Intelligence*, 37, 154-169.
9. Wang, C., Pan, H., ve Su, Y. (2020). A many-objective evolutionary algorithm with diversity-first based environmental selection. *Swarm and Evolutionary Computation*, 53, 100641.
10. Hu, Y., Zheng, J., Zou, J., Yang, S., Ou, J., ve Wang, R. (2020). A dynamic multi-objective evolutionary algorithm based on intensity of environmental change. *Information Sciences*, 523, 49-62.
11. Kennedy, J., ve Eberhart, R. (1995). Particle swarm optimization. *IEEE International Conference on Neural Networks - Conference Proceedings*.
12. Chuang, L. Y., Tsai, S. W., ve Yang, C. H. (2011). Improved binary particle swarm optimization using catfish effect for feature selection. *Expert Systems with Applications*, 38(10), 12699-12707.
13. Xue, B., Zhang, M., ve Browne, W. N. (2014). Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms. *Applied Soft Computing*, 18, 261-276.

14. Banka, H., ve Dara, S. (2015). A Hamming distance based binary particle swarm optimization (HDBPSO) algorithm for high dimensional feature selection, classification and validation. *Pattern Recognition Letters*, 52, 94-100.
15. Moradi, P., ve Gholampour, M. (2016). A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. *Applied Soft Computing*, 43, 117-130.
16. Mistry, K., Zhang, L., Neoh, S. C., Lim, C. P., ve Fielding, B. (2017). A Micro-GA Embedded PSO Feature Selection Approach to Intelligent Facial Emotion Recognition. *IEEE Transactions on Cybernetics*, 47(6).
17. Chuang, L. Y., Yang, C. H., ve Li, J. C. (2011). Chaotic maps based on binary particle swarm optimization for feature selection. *Applied Soft Computing*, 11(1), 239-248.
18. Örkücü, H. H., Özsoy, V. S., Aksoy, E., ve Dogan, M. I. (2015). Estimating the parameters of 3-p Weibull distribution using particle swarm optimization: A comprehensive experimental comparison. *Applied Mathematics and Computation*, 268.
19. Cui, Y., Shi, R., ve Dong, J. (2022). CLTSA: A Novel Tunicate Swarm Algorithm Based on Chaotic-Lévy Flight Strategy for Solving Optimization Problems. *Mathematics*, 10(18).
20. Gagnon, I., April, A., ve Abran, A. (2021). An investigation of the effects of chaotic maps on the performance of metaheuristics. *Engineering Reports*, 3(8).
21. Islam, M. J., Li, X., ve Mei, Y. (2017). A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO. *Applied Soft Computing Journal*, 59.
22. Lee, S., Soak, S., Oh, S., Pedrycz, W., ve Jeon, M. (2008). Modified binary particle swarm optimization. *Progress in Natural Science*, 18(9), 1161-1166.
23. Ting, T. O., Rao, M. V. C., ve Loo, C. K. (2006). A novel approach for unit commitment problem via an effective hybrid particle swarm optimization. *IEEE Transactions on Power Systems*, 21(1).
24. Wang, L., Wang, X., Fu, J., ve Zhen, L. (2008). A novel probability binary particle swarm optimization algorithm and its application. *Journal of Software*, 3(9).
25. Chen, Y., Liu, J., Zhu, J., ve Wang, Z. (2023). An improved binary particle swarm optimization combining V-shaped and U-shaped transfer function. *Evolutionary Intelligence*, 16, 1653-1666.
26. Guo, S. S., Wang, J. S., ve Guo, M. W. (2020). Z-Shaped Transfer Functions for Binary Particle Swarm Optimization Algorithm. *Computational Intelligence and Neuroscience*, 2020, 1-21.
27. Khosravi, H., Amiri, B., Yazdanjue, N., ve Babaiyan, V. (2022). An improved group teaching optimization algorithm based on local search and chaotic map for feature selection in high-dimensional data. *Expert Systems with Applications*, 204, 117493.

28. Mirjalili, S., Zhang, H., Mirjalili, S., Chalup, S., ve Noman, N. (2020). A Novel U-Shaped Transfer Function for Binary Particle Swarm Optimisation. *Advances in Intelligent Systems and Computing*, 1138, 241-259.
29. Crawford, B., Soto, R., Astorga, G., García, J., Castro, C., ve Paredes, F. (2017). Putting continuous metaheuristics to work in binary search spaces. *Complexity*, 2017, 1-19.
30. Prakash, J., ve Singh, P. K. (2019). Gravitational search algorithm and K-means for simultaneous feature selection and data clustering: a multi-objective approach. *Soft Computing*, 23(6).
31. Pereira, L. A. M., Papa, J. P., Coelho, A. L. V., Lima, C. A. M., Pereira, D. R., ve de Albuquerque, V. H. C. (2019). Automatic identification of epileptic EEG signals through binary magnetic optimization algorithms. *Neural Computing and Applications*, 31.
32. Crawford, B., Soto, R., Astudillo, G., Olguín, E., ve Misra, S. (2016). Solving set covering problem with fireworks explosion. *Computational Science and Its Applications - Lecture Notes in Computer Science*, 9786.
33. Crawford, B., Soto, R., Peña, C., Palma, W., Johnson, F., ve Paredes, F. (2015). Solving the set covering problem with a shuffled frog leaping algorithm. *Intelligent Information and Database Systems - Lecture Notes in Computer Science*, 9012.
34. Bellman, R. (1966). Dynamic Programming. *Science*, 153(3731), 34-37.
35. Tang, X., Dai, Y., ve Xiang, Y. (2019). Feature selection based on feature interactions with application to text categorization. *Expert Systems with Applications*, 120.
36. Rostami, M., Berahmand, K., Nasiri, E., ve Forouzande, S. (2021). Review of swarm intelligence-based feature selection methods. *Engineering Applications of Artificial Intelligence*, 100, 104210.
37. Sadeghian, Z., Akbari, E., Nematzadeh, H., ve Motameni, H. (2023). A review of feature selection methods based on meta-heuristic algorithms. *Journal of Experimental and Theoretical Artificial Intelligence*, 1-51.
38. Unler, A., Murat, A., ve Chinnam, R. B. (2011). mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification. *Information Sciences*, 181(20), 4625-4641.
39. Liu, Y., Nie, F., Gao, Q., Gao, X., Han, J., ve Shao, L. (2019). Flexible unsupervised feature extraction for image classification. *Neural Networks*, 115, 65-71.
40. Renuka Devi, D., ve Sasikala, S. (2019). Online Feature Selection (OFS) with Accelerated Bat Algorithm (ABA) and Ensemble Incremental Deep Multiple Layer Perceptron (EIDMLP) for big data streams. *Journal of Big Data*, 6(1).
41. Tadist, K., Najah, S., Nikolov, N. S., Mrabti, F., ve Zahi, A. (2019). Feature selection methods and genomic big data: a systematic review. *Journal of Big Data*, 6(1).

42. Raileanu, L. E., ve Stoffel, K. (2004). Theoretical comparison between the Gini Index and Information Gain criteria. *Annals of Mathematics and Artificial Intelligence*, 41(1).
43. He, X., Cai, D., ve Niyogi, P. (2005). Laplacian Score for feature selection. *International Conference on Neural Information Processing Systems*, 507-514.
44. Xu, Y., Jones, G., Li, J., Wang, B., ve Sun, C. (2007). Study on mutual information-based feature selection for text categorization. *Journal of Computational Information Systems*, 3(3).
45. Yamada, M., Jitkrittum, W., Sigal, L., Xing, E. P., ve Sugiyama, M. (2014). High-Dimensional feature selection by feature-Wise kernelized lasso. *Neural Computation*, 26(1).
46. Li, J., Chen, G., Li, M., ve Chen, H. (2020). An enhanced-indicator based many-objective evolutionary algorithm with adaptive reference point. *Swarm and Evolutionary Computation*, 55, 100669.
47. Santucci, V., Baiocchi, M., ve Milani, A. (2020). An algebraic framework for swarm and evolutionary algorithms in combinatorial optimization. *Swarm and Evolutionary Computation*, 55, 100673.
48. Zhang, S., Lee, C. K. M., Chan, H. K., Choy, K. L., ve Wu, Z. (2015). Swarm intelligence applied in green logistics: A literature review. *Engineering Applications of Artificial Intelligence*, 37, 154-169.
49. Wang, C., Pan, H., ve Su, Y. (2020). A many-objective evolutionary algorithm with diversity-first based environmental selection. *Swarm and Evolutionary Computation*, 53, 100641.
50. Hu, Y., Zheng, J., Zou, J., Yang, S., Ou, J., ve Wang, R. (2020). A dynamic multi-objective evolutionary algorithm based on intensity of environmental change. *Information Sciences*, 523, 49-62.
51. Bansal, J. C., Singh, P. K., ve Pal, N.R. (2019). *Evolutionary and Swarm Intelligence Algorithms*. İsviçre: Springer.
52. Parsopoulos, K. E., ve Vrahatis, M. N. (2010). *Particle Swarm Optimization and Intelligence: Advances and Applications*. Amerika Birleşik Devletleri: Information Science Publishing.
53. Thangaraj, R., Pant, M., Abraham, A., ve Snasel, V. (2012). Modified Particle Swarm Optimization with time varying velocity vector. *International Journal of Innovative Computing, Information and Control*, 8.
54. Kennedy, J. (2010). *Particle Swarm Optimization*. *Encyclopedia of Machine Learning*. Amerika Birleşik Devletleri: Springer.
55. Özsoy, V. S., Örkücü, H. H., ve Bal, H. (2018). Particle Swarm Optimization Applied to Parameter Estimation of the Four-Parameter Burr III Distribution. *Iranian Journal of Science and Technology, Transaction A: Science*, 42(2).

56. Shi, Y., ve Eberhart, R. (1998). A modified particle swarm optimizer. *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*, 69-73.
57. Chauhan, P., Deep, K., ve Pant, M. (2013). Novel inertia weight strategies for particle swarm optimization. *Memetic Computing*, 5(3).
58. Al-Hassan, W., Fayek, M. B., ve Shaheen, S. I. (2006). PSOSA: An optimized particle swarm technique for solving the urban planning problem. *2006 International Conference on Computer Engineering and Systems*, 401-405.
59. Kennedy, J., ve Eberhart, R. C. (1997). A discrete binary version of the Particle Swarm Optimization. *IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, 5, 4104-4108.
60. Xue, B., Zhang, M., ve Browne, W. N. (2013). Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE Transactions on Cybernetics*, 43(6).
61. Lu, H., Wang, X., Fei, Z., ve Qiu, M. (2014). The effects of using chaotic map on improving the performance of multiobjective evolutionary algorithms. *Mathematical Problems in Engineering*, 2014.
62. Kohli, M., ve Arora, S. (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. *Journal of Computational Design and Engineering*, 5(4).
63. Erramilli, A., Singh, R. P., ve Pruthi, P. (1995). An application of deterministic chaotic maps to model packet traffic. *Queueing Systems*, 20(1-2).
64. Caponetto, R., Fortuna, L., Fazzino, S., ve Xibilia, M. G. (2003). Chaotic sequences to improve the performance of evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 7(3).
65. Emary, E., ve Zawbaa, H. M. (2016). Impact of chaos functions on modern swarm optimizers. *PLoS ONE*, 11(7).
66. Chuang, L. Y., Yang, C. H., ve Li, J. C. (2011). Chaotic maps based on binary particle swarm optimization for feature selection. *Applied Soft Computing Journal*, 11(1).
67. Sayed, G. I., Hassanien, A. E., ve Azar, A. T. (2019). Feature selection via a novel chaotic crow search algorithm. *Neural Computing and Applications*, 31(1).
68. Nezamabadi-Pour, H., Rostami-Shahrbabaki, M., ve Farsangi, M. M. (2008). Binary Particle Swarm Optimization: challenges and New Solutions. *The Journal of Computer Society of Iran (CSI) On Computer Science and Engineering*, 6, 21-32.
69. Mirjalili, S., ve Lewis, A. (2013). S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm and Evolutionary Computation*, 9.
70. Fisher, R. A. (1935). *The design of experiments*. İngiltere: Oliver and Boyd.

71. Yates, F. (1937). *The design and analysis of factorial experiments*. İngiltere: The Imperial Bureau of Soil Science.
72. Şenoğlu, B., ve Acıtaş, Ş. (2020). *İstatistiksel Deney Tasarımı Sabit Etkili Modeller* (Dördüncü Baskı). Türkiye: NOBEL Yayınevi.
73. Alptekin, B., Acitas, S., Senoglu, B., ve Aladag, C. H. (2022). Statistical determination of significant particle swarm optimization parameters: the case of Weibull distribution. *Soft Computing*, 26(22).
74. Montgomery, D.C. (2017). *Design and analysis of experiments* (Dokuzuncu Baskı). Amerika Birleşik Devletleri: Wiley and Sons.
75. Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Amerika Birleşik Devletleri: Springer.
76. James, G., Witten, D., Hastie, T., ve Tibshirani, R. (2013). *An Introduction to Statistical Learning - with Applications in R*. Amerika Birleşik Devletleri: Springer.
77. Eiben, Á. E., Hinterding, R., ve Michalewicz, Z. (1999). Parameter control in evolutionary algorithms. *IEEE Transactions on Evolutionary Computation*, 3(2).
78. Bansal, J. C., ve Deep, K. (2012). A modified binary particle swarm optimization for Knapsack problems. *Applied Mathematics and Computation*, 218(22).
79. Jiang, H., Kwong, C. K., Chen, Z., ve Ysim, Y. C. (2012). Chaos particle swarm optimization and T-S fuzzy modeling approaches to constrained predictive control. *Expert Systems with Applications*, 39(1), 194-201.
80. Van den Bergh, F., ve Engelbrecht, A. P. (2001). Effects of swarm size on cooperative particle swarm optimisers. *Proceedings of the Genetic and Evolutionary Computation Conference*, 892-899.
81. İnternet: UC Irvine Machine Learning Repository. URL: <https://archive.ics.uci.edu/datasets>, Son Erişim Tarihi: 30.03.2024.



EKLER

EK-1. Farklı faktör seviyeleri için ACC sonuçları ve seçilen özelliklerin sayısı

Kaotik Harita	Transfer Fonksiyonu	Parçacık Sayısı	Eşik Değeri							
			0,5		0,6		0,7		0,8	
			ACC	SF	ACC	SF	ACC	SF	ACC	SF
Circle	S	10	0,9358	18,2555	0,9331	13,4666	0,9159	9,9888	0,9075	5,9888
		25	0,9349	17,8666	0,9276	13,5555	0,9174	9,6777	0,9184	6,2777
		50	0,9308	17,7666	0,9243	13,8666	0,9191	9,4777	0,9041	6,1666
		100	0,9356	18,4666	0,9302	14,0333	0,9148	9,5777	0,9334	5,9444
	U	10	0,9395	17,1222	0,9304	13,7777	0,9266	10,0888	0,9150	6,3888
		25	0,9438	16,6444	0,9285	13,5222	0,9265	10,1333	0,9153	6,9000
		50	0,9434	16,3666	0,9286	13,3000	0,9279	10,1555	0,9128	6,7555
		100	0,9306	16,6777	0,9322	13,8333	0,9328	10,0222	0,9156	6,8444
	V	10	0,9389	17,1000	0,9311	13,8888	0,9229	10,1666	0,9113	6,7222
		25	0,9318	16,6000	0,9329	13,9444	0,9252	10,3000	0,9165	6,7444
		50	0,9184	16,6888	0,9301	13,7333	0,9227	9,3666	0,9131	7,3000
		100	0,9470	17,6777	0,9299	13,6444	0,9253	10,2222	0,9211	6,8666
	Z	10	0,9340	17,4888	0,9228	12,9777	0,9183	10,0888	0,9150	6,7000
		25	0,9389	17,7333	0,9291	13,3888	0,9203	10,2666	0,9092	6,2888
		50	0,9420	17,9777	0,9245	13,7222	0,9161	9,5111	0,9062	6,5222
		100	0,9364	17,6333	0,9258	13,8444	0,9222	9,9888	0,9238	6,2222
Logistic	S	10	0,9351	18,1333	0,9304	13,4888	0,9170	9,7111	0,9341	6,2000
		25	0,9390	17,9444	0,9289	13,5777	0,9183	9,5888	0,9207	6,3777
		50	0,9354	18,5888	0,9278	14,1777	0,9170	10,1777	0,9058	6,1444
		100	0,9362	18,1111	0,9273	14,1777	0,9184	9,8333	0,9082	6,3555
	U	10	0,9391	17,2888	0,9283	13,1333	0,9282	10,0111	0,9170	6,6111
		25	0,9173	16,7444	0,9326	13,9777	0,9300	10,0444	0,9133	6,8555
		50	0,9455	16,9888	0,9267	13,5333	0,9251	10,3000	0,9145	6,5777
		100	0,9301	16,8222	0,9336	13,3555	0,9250	9,8888	0,9144	6,8888
	V	10	0,9306	17,1111	0,9338	13,5888	0,9181	9,8000	0,9146	7,4333
		25	0,9390	16,7666	0,9314	12,9888	0,9211	10,3333	0,9199	6,4777
		50	0,9334	16,7555	0,9332	13,2333	0,9225	10,0555	0,9174	7,0666
		100	0,9415	16,8555	0,9318	13,4111	0,9286	10,5888	0,9148	6,6555
	Z	10	0,9329	17,2888	0,9229	13,2333	0,9122	10,2888	0,9212	5,9222
		25	0,9370	17,6666	0,9233	13,4888	0,9177	10,1444	0,9338	6,3666
		50	0,9349	17,0555	0,9273	13,6000	0,9201	9,6777	0,9109	6,3222
		100	0,9380	17,8777	0,9222	13,6444	0,9203	10,0888	0,9053	6,4000
Piecewise	S	10	0,9351	18,2111	0,9329	13,6666	0,9170	9,7777	0,9336	6,4333
		25	0,9323	17,6777	0,9235	13,8666	0,9143	9,6222	0,9190	5,8222
		50	0,9339	18,2888	0,9313	14,0888	0,9153	9,2333	0,9220	6,0555
		100	0,9339	18,1777	0,9287	14,2666	0,9193	9,9666	0,9244	6,2333
	U	10	0,9166	17,2444	0,9355	13,4333	0,9254	10,2777	0,9143	7,0111
		25	0,9453	16,3555	0,9309	13,5222	0,9207	9,8555	0,9162	6,7666
		50	0,9294	17,2444	0,9308	13,5777	0,9260	10,1111	0,9157	6,7666
		100	0,9312	17,2000	0,9284	13,5555	0,9274	10,0000	0,9128	7,1222
	V	10	0,9391	18,0333	0,9349	13,3555	0,9240	10,7666	0,9220	6,6222
		25	0,9152	16,2000	0,9310	13,4444	0,9252	10,4111	0,9181	7,0333
		50	0,9338	16,5444	0,9329	13,9666	0,9248	10,1555	0,9186	7,3444
		100	0,9455	16,6777	0,9322	13,4444	0,9263	9,7777	0,9089	6,6000
	Z	10	0,9309	16,9000	0,9253	13,0111	0,9203	9,6555	0,9056	6,1333
		25	0,9367	17,3555	0,9257	13,2888	0,9196	9,3666	0,9013	6,3666
		50	0,9387	17,5222	0,9252	13,5444	0,9193	9,3777	0,9203	7,0111
		100	0,9400	18,1222	0,9252	13,8111	0,9199	10,0777	0,9091	6,2555
Singer	S	10	0,9289	18,1666	0,9267	13,8555	0,9159	10,1444	0,9087	6,3000
		25	0,9379	17,9111	0,9260	13,7888	0,9181	9,5222	0,9332	6,5000
		50	0,9340	18,3444	0,9309	14,0000	0,9150	10,1777	0,9048	6,0666
		100	0,9334	18,2888	0,9306	14,2333	0,9171	10,2333	0,9348	5,9000
	U	10	0,9212	16,1888	0,9287	13,6666	0,9267	10,2333	0,9094	6,7000
		25	0,9322	16,8888	0,9271	13,6666	0,9256	10,1777	0,9143	7,2333
		50	0,9287	16,8222	0,9324	13,2777	0,9226	9,8222	0,9205	6,7666
		100	0,9304	17,0111	0,9295	13,2111	0,9262	9,7777	0,9156	7,0111
	V	10	0,9282	17,2555	0,9335	13,0777	0,9222	9,8777	0,9135	6,7222
		25	0,9323	16,8222	0,9293	12,9555	0,9268	9,7000	0,9162	7,0888
		50	0,9335	16,9222	0,9277	13,4555	0,9248	10,0222	0,9138	6,5000
		100	0,9436	17,1777	0,9374	13,7111	0,9258	9,7888	0,9156	6,5333
	Z	10	0,9346	17,7111	0,9264	13,8555	0,9157	10,0222	0,9087	6,5000
		25	0,9349	18,1000	0,9249	13,7888	0,9146	9,7555	0,9172	6,0777
		50	0,9326	17,7888	0,9285	14,0000	0,9208	9,5777	0,9059	6,2444
		100	0,9378	17,8111	0,9244	13,7444	0,9200	10,2111	0,9079	6,1666

EK-2. $(\beta; \gamma) = (0,9; 0,1)$ durumu için MBPSO algoritma sonuçları

Eylemsizlik Ağırlığı	Sınırlı Fonksiyon	Parçacık Sayısı	Eşik Değeri							
			0,5		0,6		0,7		0,8	
			ACC	SF	ACC	SF	ACC	SF	ACC	SF
Lineer azalan	SF1	10	0,9720	9,0333	0,9711	8,6333	0,9701	7,9333	0,9739	7,7000
Lineer azalan	SF1	25	0,9786	8,0667	0,9783	7,0667	0,9767	7,9000	0,9755	7,1000
Lineer azalan	SF1	50	0,9799	7,2667	0,9821	7,6667	0,9827	7,2000	0,9824	6,9667
Lineer azalan	SF1	100	0,9808	7,0667	0,9818	7,1000	0,9821	7,0333	0,9811	7,0333
Lineer azalan	SF2	10	0,9648	6,8000	0,9645	6,7000	0,9623	6,7667	0,9660	6,5000
Lineer azalan	SF2	25	0,9695	6,7000	0,9673	7,0667	0,9686	5,9667	0,9686	6,7000
Lineer azalan	SF2	50	0,9692	6,7667	0,9720	7,1667	0,9701	6,9667	0,9670	6,2333
Lineer azalan	SF2	100	0,9777	6,8333	0,9764	6,9667	0,9708	6,7500	0,9698	6,3667
Lineer azalan	SF3	10	0,9764	10,4667	0,9774	9,7667	0,9786	9,9333	0,9814	9,7000
Lineer azalan	SF3	25	0,9786	9,0000	0,9833	8,7667	0,9852	8,8000	0,9852	8,0000
Lineer azalan	SF3	50	0,9840	8,4333	0,9830	8,4333	0,9890	8,1000	0,9874	8,0333
Lineer azalan	SF3	100	0,9874	8,2000	0,9852	7,3667	0,9868	7,7333	0,9865	7,1667
Rastgele ağırlıklandırma	SF1	10	0,9739	8,8667	0,9717	8,5333	0,9752	8,6333	0,9682	7,7333
Rastgele ağırlıklandırma	SF1	25	0,9770	7,8000	0,9742	7,3667	0,9792	6,8333	0,9720	6,9667
Rastgele ağırlıklandırma	SF1	50	0,9796	7,4000	0,9811	7,3667	0,9774	7,0333	0,9805	7,0333
Rastgele ağırlıklandırma	SF1	100	0,9827	7,4000	0,9843	7,1333	0,9830	6,5333	0,9814	6,9333
Rastgele ağırlıklandırma	SF2	10	0,9664	6,4333	0,9673	6,4667	0,9642	6,3000	0,9638	6,2000
Rastgele ağırlıklandırma	SF2	25	0,9733	7,2333	0,9704	6,7000	0,9758	6,3000	0,9717	6,4667
Rastgele ağırlıklandırma	SF2	50	0,9730	7,2667	0,9752	6,7333	0,9777	7,1333	0,9783	6,5333
Rastgele ağırlıklandırma	SF2	100	0,9752	7,1333	0,9808	6,8000	0,9739	6,4000	0,9736	6,5000
Rastgele ağırlıklandırma	SF3	10	0,9748	10,0000	0,9770	10,3667	0,9745	9,2667	0,9792	9,5000
Rastgele ağırlıklandırma	SF3	25	0,9840	9,6667	0,9833	8,5333	0,9843	8,4667	0,9849	8,2667
Rastgele ağırlıklandırma	SF3	50	0,9852	7,9000	0,9849	8,4000	0,9833	7,9000	0,9862	7,5333
Rastgele ağırlıklandırma	SF3	100	0,9874	8,1000	0,9855	7,6667	0,9849	7,6000	0,9846	7,4667
Soğutma fonksiyonu	SF1	10	0,9736	9,3000	0,9745	8,8333	0,9748	9,3333	0,9714	8,0667
Soğutma fonksiyonu	SF1	25	0,9821	8,0333	0,9780	7,8667	0,9780	7,2000	0,9774	7,0667
Soğutma fonksiyonu	SF1	50	0,9805	7,1667	0,9808	7,2333	0,9821	7,1000	0,9814	7,0667
Soğutma fonksiyonu	SF1	100	0,9833	7,2333	0,9840	7,1333	0,9852	6,7000	0,9811	6,8333
Soğutma fonksiyonu	SF2	10	0,9597	6,2000	0,9575	6,1333	0,9566	5,8667	0,9519	5,7333
Soğutma fonksiyonu	SF2	25	0,9629	6,0667	0,9619	6,0000	0,9626	6,0333	0,9610	5,9000
Soğutma fonksiyonu	SF2	50	0,9692	6,4333	0,9651	6,4000	0,9638	5,7667	0,9670	6,2333
Soğutma fonksiyonu	SF2	100	0,9679	5,8667	0,9673	5,9333	0,9673	5,7000	0,9610	6,0333
Soğutma fonksiyonu	SF3	10	0,9730	9,9667	0,9786	10,5000	0,9748	9,7000	0,9792	8,8000
Soğutma fonksiyonu	SF3	25	0,9796	9,1667	0,9814	8,6333	0,9824	8,1000	0,9805	8,0667
Soğutma fonksiyonu	SF3	50	0,9821	8,5000	0,9830	8,0333	0,9833	8,0000	0,9849	7,6000
Soğutma fonksiyonu	SF3	100	0,9833	8,0667	0,9862	8,2667	0,9865	7,6333	0,9862	7,0000
Lojistik	SF1	10	0,9730	9,4667	0,9783	8,9000	0,9726	8,5000	0,9686	8,2000
Lojistik	SF1	25	0,9783	8,6667	0,9796	7,9667	0,9799	7,8000	0,9811	7,7667
Lojistik	SF1	50	0,9774	7,7667	0,9786	7,8000	0,9770	7,5333	0,9852	6,9000
Lojistik	SF1	100	0,9818	7,7000	0,9836	7,2333	0,9865	6,9333	0,9843	7,1333
Lojistik	SF2	10	0,9739	5,8667	0,9682	6,0667	0,9701	6,5333	0,9755	6,4667
Lojistik	SF2	25	0,9748	6,1333	0,9764	6,2667	0,9770	6,5333	0,9818	6,5667
Lojistik	SF2	50	0,9811	6,4333	0,9840	6,2333	0,9780	6,1667	0,9770	5,9000
Lojistik	SF2	100	0,9799	6,3333	0,9780	5,8333	0,9814	6,3000	0,9818	6,5667
Lojistik	SF3	10	0,9730	9,7667	0,9792	9,5333	0,9761	9,0667	0,9755	8,5333
Lojistik	SF3	25	0,9805	9,0667	0,9811	8,4667	0,9814	8,4000	0,9799	8,3333
Lojistik	SF3	50	0,9789	8,4333	0,9830	7,9333	0,9824	7,8667	0,9792	7,5333
Lojistik	SF3	100	0,9849	8,0000	0,9877	7,1000	0,9874	7,1667	0,9868	7,0333

EK-3. $(\beta; \gamma) = (0,5; 0,5)$ durumu için MBPSO algoritma sonuçları

Eylemsizlik Ağırlığı	Sınırlı Fonksiyon	Parçacık Sayısı	Eşik Değeri							
			0,5		0,6		0,7		0,8	
			ACC	SF	ACC	SF	ACC	SF	ACC	SF
Lineer azalan	SF1	10	0,9107	3,8667	0,9170	4,1000	0,9182	3,8000	0,9135	3,7333
Lineer azalan	SF1	25	0,9242	3,2000	0,9289	3,5333	0,9157	3,0000	0,9116	2,8000
Lineer azalan	SF1	50	0,9223	2,8667	0,9195	2,7333	0,9239	2,7667	0,9264	2,8000
Lineer azalan	SF1	100	0,9236	2,8000	0,9305	2,7000	0,9192	2,5333	0,9176	2,4667
Lineer azalan	SF2	10	0,8858	2,5333	0,8805	2,3667	0,8912	2,6667	0,8821	2,2333
Lineer azalan	SF2	25	0,8934	2,4000	0,8934	2,7000	0,8934	2,4000	0,8865	2,3667
Lineer azalan	SF2	50	0,9082	2,4333	0,8934	2,2667	0,8912	2,2333	0,8994	2,4000
Lineer azalan	SF2	100	0,9129	2,6000	0,8997	2,3667	0,9009	2,2000	0,9025	2,3333
Lineer azalan	SF3	10	0,9242	5,2000	0,9289	4,8667	0,9239	4,1000	0,9245	4,3000
Lineer azalan	SF3	25	0,9327	3,9667	0,9340	3,5333	0,9280	3,2667	0,9308	3,1000
Lineer azalan	SF3	50	0,9242	3,1000	0,9362	3,0000	0,9305	2,8667	0,9270	2,8667
Lineer azalan	SF3	100	0,9349	3,0000	0,9362	3,0000	0,9302	2,8333	0,9327	2,8333
Rastgele ağırlıklandırma	SF1	10	0,9066	3,8667	0,9151	3,7000	0,9176	3,6000	0,9230	3,3667
Rastgele ağırlıklandırma	SF1	25	0,9129	3,0667	0,9142	3,0000	0,9148	2,6333	0,9198	2,8000
Rastgele ağırlıklandırma	SF1	50	0,9264	3,0333	0,9239	2,6667	0,9252	2,7667	0,9292	2,8000
Rastgele ağırlıklandırma	SF1	100	0,9204	2,5333	0,9302	2,8000	0,9252	2,7333	0,9173	2,3667
Rastgele ağırlıklandırma	SF2	10	0,8440	2,1667	0,8462	2,2667	0,8506	2,0667	0,8538	2,1667
Rastgele ağırlıklandırma	SF2	25	0,8491	2,1000	0,8597	2,3333	0,8629	2,3333	0,8528	1,9667
Rastgele ağırlıklandırma	SF2	50	0,8437	2,1333	0,8613	2,2000	0,8667	2,2667	0,8638	2,2000
Rastgele ağırlıklandırma	SF2	100	0,8541	1,8667	0,8632	2,2000	0,8629	1,8667	0,8752	2,2000
Rastgele ağırlıklandırma	SF3	10	0,9164	5,0000	0,9299	4,6000	0,9223	4,0333	0,9286	4,0000
Rastgele ağırlıklandırma	SF3	25	0,9289	3,9333	0,9311	3,5667	0,9346	3,6667	0,9277	3,1667
Rastgele ağırlıklandırma	SF3	50	0,9368	3,7667	0,9305	3,0333	0,9283	3,0000	0,9336	2,9333
Rastgele ağırlıklandırma	SF3	100	0,9336	2,9667	0,9365	3,0000	0,9343	2,8333	0,9324	2,8333
Soğutma fonksiyonu	SF1	10	0,9164	3,8333	0,9031	3,7333	0,9142	3,8667	0,9107	3,6333
Soğutma fonksiyonu	SF1	25	0,9217	3,1333	0,9031	3,7333	0,9211	3,0667	0,9132	2,7000
Soğutma fonksiyonu	SF1	50	0,9226	2,9000	0,9230	2,8667	0,9233	2,7667	0,9214	2,8333
Soğutma fonksiyonu	SF1	100	0,9352	2,8667	0,9274	2,8667	0,9305	2,6333	0,9267	2,6333
Soğutma fonksiyonu	SF2	10	0,8896	2,1000	0,8884	2,2667	0,8862	2,0333	0,8893	2,2333
Soğutma fonksiyonu	SF2	25	0,9006	2,2000	0,8969	2,0333	0,9057	2,2333	0,8981	2,3000
Soğutma fonksiyonu	SF2	50	0,9041	2,0667	0,9038	2,0333	0,9164	2,4000	0,9041	2,2333
Soğutma fonksiyonu	SF2	100	0,9264	2,3000	0,9079	2,2000	0,9110	2,3000	0,9041	2,1000
Soğutma fonksiyonu	SF3	10	0,9204	5,4667	0,9248	4,6333	0,9186	4,1333	0,9189	4,1667
Soğutma fonksiyonu	SF3	25	0,9318	3,7667	0,9261	3,5333	0,9314	3,5333	0,9264	3,2333
Soğutma fonksiyonu	SF3	50	0,9308	3,4333	0,9248	3,0000	0,9343	3,2000	0,9330	3,1000
Soğutma fonksiyonu	SF3	100	0,9302	2,8000	0,9314	2,9333	0,9292	2,9667	0,9333	2,8667
Lojistik	SF1	10	0,9201	4,7000	0,9204	4,2000	0,9204	3,9667	0,9179	3,5000
Lojistik	SF1	25	0,9110	3,5333	0,9214	3,5000	0,9186	3,1333	0,9223	3,0000
Lojistik	SF1	50	0,9233	3,0000	0,9164	2,8333	0,9230	2,9000	0,9233	2,8333
Lojistik	SF1	100	0,9239	2,8000	0,9340	2,7667	0,9302	2,9000	0,9204	2,6333
Lojistik	SF2	10	0,8903	2,0667	0,9003	2,1000	0,8978	2,2667	0,9142	2,3333
Lojistik	SF2	25	0,9142	2,3000	0,9129	2,1667	0,9129	2,3667	0,9179	2,3667
Lojistik	SF2	50	0,9189	2,5000	0,9223	2,4667	0,9192	2,4333	0,9138	2,2333
Lojistik	SF2	100	0,9270	2,3333	0,9220	2,3667	0,9274	2,4333	0,9201	2,4000
Lojistik	SF3	10	0,9186	5,1000	0,9349	4,7333	0,9214	4,2000	0,9151	4,0667
Lojistik	SF3	25	0,9195	3,8333	0,9258	3,6667	0,9223	3,3667	0,9179	2,9333
Lojistik	SF3	50	0,9226	3,2667	0,9233	3,2000	0,9217	3,0000	0,9245	2,8333
Lojistik	SF3	100	0,9239	2,8000	0,9264	2,9000	0,9355	2,9000	0,9292	2,7667

EK-4. $(\beta; \gamma) = (0,1; 0,9)$ durumu için MBPSO algoritma sonuçları

Eylemsizlik Ağırlığı	Sınırlı Fonksiyon	Parçacık Sayısı	Eşik Değeri							
			0,5		0,6		0,7		0,8	
			ACC	SF	ACC	SF	ACC	SF	ACC	SF
Lineer azalan	SF1	10	0,8104	2,6667	0,7711	1,9333	0,7871	2,2000	0,7969	1,9667
Lineer azalan	SF1	25	0,7651	1,1333	0,7660	1,0667	0,7689	1,0000	0,7811	1,0000
Lineer azalan	SF1	50	0,7868	1,0000	0,7761	1,0000	0,7874	1,0000	0,7994	1,0000
Lineer azalan	SF1	100	0,8050	1,0000	0,8088	1,0000	0,8179	1,0000	0,8242	1,0000
Lineer azalan	SF2	10	0,7267	1,0000	0,7475	1,0000	0,7478	1,0000	0,7349	1,0000
Lineer azalan	SF2	25	0,7506	1,0000	0,7324	1,0000	0,7330	1,0000	0,7695	1,0000
Lineer azalan	SF2	50	0,7686	1,0000	0,7651	1,0000	0,7434	1,0000	0,7651	1,0000
Lineer azalan	SF2	100	0,7726	1,0000	0,7547	1,0000	0,7818	1,0000	0,7742	1,0000
Lineer azalan	SF3	10	0,8305	3,2333	0,8302	3,0667	0,8097	2,3667	0,8050	2,1667
Lineer azalan	SF3	25	0,7682	1,5000	0,7547	1,3667	0,7654	1,0667	0,7708	1,1000
Lineer azalan	SF3	50	0,7443	1,1333	0,7852	1,0667	0,8179	1,0000	0,8145	1,0000
Lineer azalan	SF3	100	0,7928	1,0000	0,8230	1,0000	0,8160	1,0000	0,8211	1,0000
Rastgele ağırlıklandırma	SF1	10	0,8009	2,2333	0,7934	1,7000	0,7469	1,4000	0,7651	1,4000
Rastgele ağırlıklandırma	SF1	25	0,7557	1,2000	0,7629	1,0667	0,7758	1,0333	0,7909	1,0000
Rastgele ağırlıklandırma	SF1	50	0,7774	1,0000	0,7965	1,0000	0,8009	1,0000	0,8050	1,0000
Rastgele ağırlıklandırma	SF1	100	0,8233	1,0000	0,8116	1,0000	0,8176	1,0000	0,8226	1,0000
Rastgele ağırlıklandırma	SF2	10	0,7371	1,0000	0,7264	1,0000	0,7352	1,0000	0,7541	1,0000
Rastgele ağırlıklandırma	SF2	25	0,7258	1,0000	0,7355	1,0000	0,7626	1,0000	0,7349	1,0000
Rastgele ağırlıklandırma	SF2	50	0,7198	1,0000	0,7443	1,0000	0,7381	1,0000	0,7340	1,0000
Rastgele ağırlıklandırma	SF2	100	0,7182	1,0000	0,7478	1,0000	0,7453	1,0000	0,7469	1,0000
Rastgele ağırlıklandırma	SF3	10	0,8487	3,8000	0,8336	3,2333	0,8107	2,5667	0,8072	2,4667
Rastgele ağırlıklandırma	SF3	25	0,7997	1,9667	0,7858	1,4667	0,7541	1,2333	0,7909	1,3667
Rastgele ağırlıklandırma	SF3	50	0,7720	1,1000	0,7789	1,0000	0,7928	1,0000	0,8075	1,0000
Rastgele ağırlıklandırma	SF3	100	0,8110	1,0000	0,8151	1,0000	0,8217	1,0000	0,8346	1,0000
Soğutma fonksiyonu	SF1	10	0,7981	2,9000	0,8035	2,4333	0,7484	1,8000	0,7701	1,6667
Soğutma fonksiyonu	SF1	25	0,7667	1,2333	0,7610	1,1667	0,7566	1,0333	0,7739	1,0333
Soğutma fonksiyonu	SF1	50	0,7925	1,0000	0,7975	1,0000	0,7969	1,0000	0,8094	1,0000
Soğutma fonksiyonu	SF1	100	0,8239	1,0000	0,8107	1,0000	0,8214	1,0000	0,8041	1,0000
Soğutma fonksiyonu	SF2	10	0,7918	1,0000	0,7959	1,0000	0,7950	1,0000	0,8060	1,0000
Soğutma fonksiyonu	SF2	25	0,8208	1,0000	0,8167	1,0000	0,8220	1,0000	0,8299	1,0000
Soğutma fonksiyonu	SF2	50	0,8343	1,0000	0,8305	1,0000	0,8336	1,0000	0,8324	1,0000
Soğutma fonksiyonu	SF2	100	0,8390	1,0000	0,8387	1,0000	0,8327	1,0000	0,8248	1,0000
Soğutma fonksiyonu	SF3	10	0,8475	3,7000	0,8412	3,5333	0,8239	2,7667	0,7918	2,2667
Soğutma fonksiyonu	SF3	25	0,7513	1,8667	0,7836	1,5667	0,7638	1,2000	0,7774	1,1333
Soğutma fonksiyonu	SF3	50	0,7557	1,1667	0,7601	1,1000	0,7670	1,1000	0,8047	1,0667
Soğutma fonksiyonu	SF3	100	0,7943	1,0333	0,8006	1,0333	0,8223	1,0000	0,8261	1,0000
Lojistik	SF1	10	0,8248	3,2667	0,7796	2,2000	0,7840	1,9000	0,7770	1,7667
Lojistik	SF1	25	0,7780	1,3333	0,7638	1,2667	0,7607	1,1667	0,7717	1,0000
Lojistik	SF1	50	0,7777	1,1333	0,7868	1,0333	0,7865	1,0000	0,8129	1,0000
Lojistik	SF1	100	0,7978	1,0000	0,8003	1,0000	0,8129	1,0000	0,8110	1,0000
Lojistik	SF2	10	0,7984	1,0000	0,7978	1,0000	0,8053	1,0000	0,8057	1,0000
Lojistik	SF2	25	0,8170	1,0000	0,8151	1,0000	0,8321	1,0000	0,8264	1,0000
Lojistik	SF2	50	0,8296	1,0000	0,8302	1,0000	0,8289	1,0000	0,8336	1,0000
Lojistik	SF2	100	0,8336	1,0000	0,8352	1,0000	0,8384	1,0000	0,8399	1,0000
Lojistik	SF3	10	0,8355	3,5667	0,8267	2,9333	0,8038	2,5333	0,9151	4,0667
Lojistik	SF3	25	0,7836	1,9667	0,7802	1,5333	0,7701	1,2000	0,9179	2,9333
Lojistik	SF3	50	0,7931	1,2667	0,7575	1,1333	0,8031	1,0000	0,9245	2,8333
Lojistik	SF3	100	0,7736	1,0000	0,7862	1,0333	0,8126	1,0000	0,9292	2,7667



Gazili olmak ayrıcalıktır