

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**SIKIŞTIRILMIŞ ALGILAMADA SEZGİSEL
ALGORİTMALARIN KULLANIMINA DAYALI
YÖNTEMLERİN İNCELENMESİ VE GELİŞTİRİLMESİ**

**Hazırlayan
Murat Emre ERKOÇ**

**Danışman
Prof. Dr. Nurhan KARABOĞA**

Yüksek Lisans Tezi

**Aralık 2017
KAYSERİ**

**T.C.
ERCIYES ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK – ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI**

**SIKIŞTIRILMIŞ ALGILAMADA SEZGİSEL
ALGORİTMALARIN KULLANIMINA DAYALI
YÖNTEMLERİN İNCELENMESİ VE GELİŞTİRİLMESİ**

(Yüksek Lisans Tezi)

**Hazırlayan
Murat Emre ERKOÇ**

**Danışman
Prof. Dr. Nurhan KARABOĞA**

**Bu çalışma; Erciyes Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon
Birimi tarafından FYL-2017-7068 kodlu proje ile desteklenmiştir.**

**Aralık 2017
KAYSERİ**

BİLİMSEL ETİĞE UYGUNLUK

Bu çalışmadaki tüm bilgilerin, akademik ve etik kurallara uygun bir şekilde elde edildiğini beyan ederim. Aynı zamanda bu kural ve davranışların gerektirdiği gibi, bu çalışmanın özünde olmayan tüm materyal ve sonuçları tam olarak aktardığımı ve referans gösterdiğimi belirtirim.



Murat Emre ERKOÇ

YÖNERGEYE UYGUNLUK

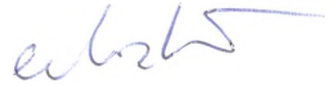
“Sıkıştırılmış Algılamada Sezgisel Algoritmaların Kullanımına Dayalı Yöntemlerin İncelenmesi Ve Geliştirilmesi” adlı Yüksek Lisans tezi, Erciyes Üniversitesi Lisansüstü Tez Önerisi ve Tez Yazma Yönergesi’ne uygun olarak hazırlanmıştır.

Tezi Hazırlayan



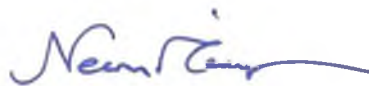
Murat Emre ERKOÇ

Danışman



Prof. Dr. Nurhan KARABOĞA

Elektrik-Elektronik Mühendisliği ABD Başkanı



Prof. Dr. Necmi TAŞPINAR

Prof. Dr. Nurhan KARABOĞA danışmanlığında **Murat Emre ERKOÇ** tarafından hazırlanan “**Sıkıştırılmış Algılamada Sezgisel Algoritmaların Kullanımına Dayalı Yöntemlerin İncelenmesi ve Geliştirilmesi**” adlı bu çalışma, jürimiz tarafından Erciyes Üniversitesi Fen Bilimleri Enstitüsü Elektrik-Elektronik Mühendisliği Anabilim Dalında **Yüksek Lisans** tezi olarak kabul edilmiştir.

28 /12 / 2017

JÜRİ:

Danışman : Prof. Dr. Nurhan KARABOĞA

Üye : Prof. Dr. İbrahim DEVELİ

Üye : Yrd. Doç. Dr. Serdar KOÇKANAT

ONAY:

Bu tezin kabulü Enstitü Yönetim Kurulunun 02/01/2018 tarih ve 2018/01-15... sayılı kararı ile onaylanmıştır.


02 / 01 / 2018

Prof. Dr. Mehmet AKKURT

Enstitü Müdürü

TEŞEKKÜR

Öncelikle bilgi ve tecrübelerini benimle paylaşan, göstermiş olduğu yakın ilgi ve çalışmaya teşvik edici tutumu ile daha kolay yol almamı sağlayan danışman hocam sayın Prof. Dr. Nurhan KARABOĞA'ya teşekkürlerimi sunarım.

Sadece bu tez çalışmasında değil, hayat boyu maddi ve manevi her türlü desteği sağlayıp bugünlere gelmemde büyük emekleri olan aileme çok teşekkür ederim.

Ayrıca sevgisi ve güler yüzüyle bana güç verip desteğini benden esirgemeyen sevgili eşim Tuba'ya teşekkür ederim.

Bu tez çalışmasına maddi destek veren Erciyes Üniversitesi Bilimsel Araştırma Projeleri Birimi'ne (Proje No: FYL-2017-7068) teşekkür ederim.

Murat Emre ERKOÇ

Kayseri, Aralık 2017

SIKIŞTIRILMIŞ ALGILAMADA SEZGİSEL ALGORİTMALARIN KULLANIMINA DAYALI YÖNTEMLERİN İNCELENMESİ VE GELİŞTİRİLMESİ

Murat Emre ERKOÇ

**Erciyes Üniversitesi, Fen Bilimleri Enstitüsü
Yüksek Lisans Tezi, Aralık 2017
Tez Danışmanı: Prof. Dr. Nurhan KARABOĞA**

ÖZET

Bu tez çalışması sıkıştırılmış algılamada seyrek sinyali geri oluşturmak için kullanılan algoritmaları incelemektedir. Sıkıştırılmış algılamada seyrek sinyali geri oluştururken genel olarak l_1 minimizasyonu ve ağgözlü algoritmalar kullanılmaktadır. Bu yöntemlere ek olarak, son zamanlarda sezgisel algoritmalar da seyrek sinyalin geri oluşturulmasında kullanılmaya başlanmıştır. Sezgisel algoritmalar, diğer yöntemlerle hibrit bir yapı oluşturularak veya kendi başına seyrek sinyali geri oluşturma problemini optimize ederek sıkıştırılmış algılama alanında kullanılmaktadır. Bu tez çalışmasında seyrek sinyali geri elde etmek için l_1 minimizasyon algoritmalarından Birincil İkili İç Nokta yöntemi ve ağgözlü algoritma sınıfının önde gelen algoritmalarından Dik Eşleştirme Takibi ve Sıkıştırılmış Algılamalı Eşleştirme Takibi algoritmaları incelenmiş ve bunların MATLAB ortamında benzetimi gerçekleştirilerek performans karşılaştırması yapılmıştır. Daha sonra, sezgisel algoritmaların bu alanda kullanılması için bir yöntem önerilmiştir. Sezgisel algoritmalar için önerilen bu yöntem, sıkıştırılmış algılamada kullanılan ağgözlü algoritmaların seyrek sinyali geri elde yöntemine benzer bir stratejiye sahiptir. Ayrıca bu yöntem Genetik, Diferansiyel Gelişim, Parçacık Sürüsü Optimizasyonu ve Yapay Arı Koloni algoritmaları için uygulanmıştır. Önerilen yöntemin performansı l_1 minimizasyon yöntemi ve ağgözlü algoritmalarla karşılaştırılmıştır. Elde edilen test sonuçlarına göre önerilen yöntem, seyrek sinyali geri elde etmede bazı durumlarda daha iyi sonuç vermiştir. Bazı durumlarda da en az klasik yöntemler kadar iyi sonuç vermiştir. Elde edilen sonuçlara göre sezgisel algoritmalar

iin nerilen yntem sıkıřtırılmıř algılamada seyrek sinyali geri elde etme algoritması olarak kullanılabilir.

Anahtar kelimeler: Sıkıřtırılmıř algılama, seyrek iřaret geri oluřturma, sezgisel algoritmalar, agzl algoritmalar.



ANALYZING AND DEVELOPMENT OF HEURISTIC ALGORITHMS IN COMPRESSED SENSING

Murat Emre ERKOÇ

Erciyes University, Graduate School of Natural and Applied Sciences

Master Thesis, December 2017

Thesis Supervisor: Prof. Dr. Nurhan KARABOĞA

ABSTRACT

In this study, compressed sensing sparse signal reconstruction methods are analyzed. In general, there are two classes of algorithms in recovery of sparse signal: greedy approaches and l_1 -minimization methods. In addition to these techniques, lately, heuristic algorithms have been started to studying as sparse signal recovery algorithms. Heuristic algorithms can be used in this area as hybrid mechanism with other techniques or can be used to optimize sparse recovery problem by itself. Firstly in this study, widely used algorithms of l_1 minimization and greedy approaches are compared. The compared algorithms are Primal Dual Interior Point method, Orthogonal Matching Pursuit and Compressed Sensing Matching Pursuit algorithms. They are implemented on MATLAB and their performance metrics are compared. And then, a new method that can solve compressed sensing sparse recovery problem is proposed for heuristic algorithms. This proposed method is similar to greedy algorithms used to solve sparse recovery problem. Also this proposed method is implemented on MATLAB for Genetic Algorithm, Differential Evolution, Particle Swarm Optimization and Artificial Bee Colony algorithms. The performance metrics of these algorithms are compared with the mentioned algorithms of l_1 minimization and greedy approaches. According to these performance metrics results, for some test cases, proposed method for heuristic algorithms have better performance than other compared three algorithms. Also for some other test cases, proposed method has at least good as l_1 minimization and greedy algorithms for sparse signal reconstruction. As a result from these studies, the proposed method for heuristic algorithms can be used as compressed sensing recovery algorithms.

Keywords: Compressed sensing, sparse signal reconstruction, heuristic algorithms, greedy algorithms.



İÇİNDEKİLER

SIKIŞTIRILMIŞ ALGILAMADA SEZGİSEL ALGORİTMALARIN KULLANIMINA DAYALI YÖNTEMLERİN İNCELENMESİ VE GELİŞTİRİLMESİ

BİLİMSEL ETİĞE UYGUNLUK SAYFASI	i
YÖNERGEYE UYGUNLUK SAYFASI.....	ii
KABUL VE ONAY SAYFASI	iii
TEŞEKKÜR.....	iv
ÖZET.....	v
ABSTRACT.....	vii
İÇİNDEKİLER	ix
KISALTMA VE SİMGELER.....	xii
ŞEKİLLER LİSTESİ	xv
TABLolar LİSTESİ.....	xix

GİRİŞ

Tezin Amacı ve Önemi.....	1
---------------------------	---

1. BÖLÜM

GENEL BİLGİLER VE LİTERATÜR ARAŞTIRMASI

1.1. Sıkıştırılmış Algılama	4
1.1.1. Seyrek ve Sıkıştırılabilir İşaretler	5
1.1.2. Kısıtlı İzometri Özelliği.....	6
1.1.3. Tutarsızlık	6
1.1.4. Literatür Araştırması	7

2. BÖLÜM

SEYREK İŞARET YENİDEN OLUŞTURMA ALGORİTMALARI

2.1. L1 minimizasyonu	12
2.1.1. Birincil İkili İç Nokta	13
2.2. Ağgözlü Algoritmalar	14
2.2.1. OMP (Dikey Eşleştirme Takibi) Algoritması	14
2.2.2 CoSaMP (Sıkıştırılmış Algılamalı Eşleştirme Takibi) Algoritması	15

3. BÖLÜM

SEZGİSEL ALGORİTMALAR

3.1. Genetik Algoritma	17
3.2. Diferansiyel Gelişim Algoritması	19
3.3. Parçacık Sürüsü Optimizasyon Algoritması	21
3.4. Yapay Arı Koloni Algoritması	23
3.5. Önerilen Metot	25

4. BÖLÜM

KLASİK ALGORİTMALARIN BENZETİM SONUÇLARI

4.1. Test 1	29
4.2. Test 2	32
4.3. Test 3	35

5. BÖLÜM

POPÜLASYON TABANLI ALGORİTMALARIN BENZETİM SONUÇLARI

5.1. Uygulanan Benzetim Çalışması Hakkında Bilgi	39
5.2. Test 1	43
5.2.1. Örnek 1 $N=64, M=24, k=8$	44
5.2.2. Örnek 2 $N=64, M=24, k=12$	46

5.2.3. Örnek 3 $N=64$, $M=48$, $k=8$	47
5.2.4. Örnek 4 $N=64$, $M=48$, $k=12$	48
5.3. Test 2	50
5.3.1 Popülasyon tabanlı algoritmalarla sinyalin ölçüm sayısına karşılık hatasının değişiminin incelenmesi	50
5.3.2 Klasik algoritmalarla sinyalin ölçüm sayısına karşılık hatasının değişiminin incelenmesi	53
5.3.3 Popülasyon tabanlı algoritmalarla seyreklik seviyesine karşılık geri oluşturma hatasının incelenmesi	55
5.3.4 Klasik algoritmalarla seyreklik seviyesine karşılık geri oluşturma hatasının incelenmesi	57
5.4. Gürültülü Ölçümden Sinyalin Geri Elde Edilmesi	60
5.5. Diğer Çalışmalarla Karşılaştırma	66
6. BÖLÜM	
SONUÇLAR TARTIŞMA VE ÖNERİLER.....	67
KAYNAKÇA	69
ÖZGEÇMİŞ.....	74

KISALTMA VE SİMGELER

<u>Kısaltma</u>	<u>Anlamı</u>
CS	(Compressed Sensing)→Sıkıştırılmış Algılama.
BP	(Basis Pursuit)→Taban Arayışı.
MP	(Matching Pursuit)→Uyumlu Arayış.
OMP	(Orthogonal Matching Pursuit)→Dik Uyumlu Arayış .
CoSaMP	(Compressive Sampling Matching Pursuit)→Sıkıştırılmalı Örneklemeli Uyumlu Arayış.
PDIP	(Primal Dual Interior Point)→ Birincil İkili İç Nokta yöntemi.
GA	(Genetic Algorithm)→Genetik Algoritma.
DE	(Differential Evolution)→Diferansiyel Gelişim.
PSO	(Particle Swarm Optimization)→ Parçacık Sürüşü Optimizasyon.
ABC	(Artificial Bee Colony) →Yapay Arı Koloni.
RIP	(Restricted Isometry Property)→ Kısıtlı İzometri Özelliği.
NESTA	(Nesterov's algorithm)→ Nesterov'un algoritması.
SOCP	(Second-Order Cone Programming)→ ikinci dereceden koni programlama.
FISTA	(Fast Iterative Shrinkage-Thresholding Algorithm)→Hızlı İteratif Küçülme-Eşik Algoritma.
FPC	(Fixed-Point Continuation)→ Sabit Nokta Devamlılığı.
GPSR	(Gradient Projection Sparse Reconstruction)→Eğimli İzdüşüm Seyrek Geri Elde Etme.

TWlst	(Two-step Iterative shrinkage-thresholding)→ İki basamaklı Yinelemeli küçülme-eşikleme.
Sparsa	(Sparse reconstruction by separable approximation) →Ayrılabilir yaklaşımlı seyrek geri elde etme.
ROMP	(Regularized orthogonal matching pursuit)→Düzenlenmiş dik uyumlu arayış.
stOMP	(stagewise Orthogonal Matching Pursuit)→Basamaklı Dik Uyumlu Arayış.
HSAT	(Hıybrıd Simulated Annealing Thresholding)→Hıbrıd Benzetimli Tavlama Eşikleme.
IT	(Iterative Thresholding)→ Özyinelemeli Eşikleme.
SA	(Simulated Annealing)→Isıl İşlem.
MSE	(Mean Squared Error)→Ortalama Karesel Hata.
SNR	(Signal to Noise Ratio)→Sinyalin Gürültüye Oranı.
BO	Başarı Oranı.

Simge**Anlamı**

N	Sinyal uzunluğu
M	Ölçüm sayısı.
k	Seyreklik seviyesi.
Φ	Ölçüm matrisi.
x	Orijinal sinyal.
y	Gözlem vektörü.

ψ	Ortonormal taban.
δ_k	İzometri katsayısı.
μ	Tutarlık (Coherence).
r	Rezidü vektör.
Λ	İndis Kümesi.
$\Phi^\dagger$	Ölçüm matrisinin sözde tersi.
NP	Popülasyon sayısı.
CR	Çaprazlama oranı.
F	Ölçekleme oranı.
G_{max}	Jenerasyon sayısı.
$\vec{p}_i$	Parçacığın uygunluk değeri.
$\vec{x}_i$	Parçacığın pozisyonu.
$\vec{v}_i$	Parçacığın hızı.
$x_{i,j}$	ABC algoritması için başlangıç değeri.
P_i	ABC algoritması için çözümlerin olasılık değeri.
fit_i	ABC algoritması için çözümlerin uygunluk değeri.

ŞEKİLLER LİSTESİ

Şekil 3.1	GA akış şeması.	18
Şekil 3.2	DE Algoritması akış şeması.	21
Şekil 3.3	PSO Algoritması akış şeması.	22
Şekil 3.4	ABC Algoritması akış şeması.....	25
Şekil 3.5	Önerilen Metot akış şeması.	27
Şekil 4.1	$N=256, M=128, k=32$ için sırasıyla orijinal sinyal, sinyalin Fourier dönüşümü ve orijinal sinyalin örneklenmiş şekli	30
Şekil 4.2	$N=256, M=128, k=32$ için sırasıyla PDIP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal	30
Şekil 4.3	$N=256, M=128, k=32$ için sırasıyla OMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.....	31
Şekil 4.4	$N=256, M=128, k=32$ için sırasıyla CoSaMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal	31
Şekil 4.5	$N=256, M=128, k=16$ için sırasıyla orijinal sinyal, sinyalin Fourier dönüşümü ve orijinal sinyalin örneklenmiş şekli	33
Şekil 4.6	$N=256, M=128, k=16$ için sırasıyla PDIP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal	33
Şekil 4.7	$N=256, M=128, k=16$ için sırasıyla OMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.....	34

Şekil 4.8	$N=256, M=128, k=16$ için sırasıyla CoSaMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal	34
Şekil 4.9	$N=256, M=64, k=16$ için sırasıyla orijinal sinyal, sinyalin Fourier dönüşümü ve orijinal sinyalin örneklenmiş şekli	36
Şekil 4.10	$N=256, M=64, k=16$ için sırasıyla PDIP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal	36
Şekil 4.11	$N=256, M=64, k=16$ için sırasıyla OMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.....	37
Şekil 4.12	$N=256, M=64, k=16$ için sırasıyla CoSaMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal	37
Şekil 5.1	Popülasyon sayısının 10 olması durumu için algoritmaların yinelemeye bağlı olarak uygunluk değerinin değişimi	41
Şekil 5.2	Popülasyon sayısının 20 olması durumu için algoritmaların yinelemeye bağlı olarak uygunluk değerinin değişimi.....	41
Şekil 5.3	Popülasyon sayısının 30 olması durumu için algoritmaların yinelemeye bağlı olarak uygunluk değerinin değişimi.....	42
Şekil 5.4	Popülasyon sayısının 40 olması durumu için algoritmaların yinelemeye bağlı olarak uygunluk değerinin değişimi	42
Şekil 5.5	Popülasyon sayısının 50 olması durumu için algoritmaların yinelemeye bağlı olarak uygunluk değerinin değişimi.....	43
Şekil 5.6	GA için geri oluşturulan sinyalin ölçüm sayısına karşılık MSE hatasının değişimi.....	51
Şekil 5.7	DE algoritması için geri oluşturulan sinyalin ölçüm sayısına karşılık MSE hatasının değişimi.....	51

Şekil 5.8	PSO algoritması için geri oluşturulan sinyalin ölçüm sayısına karşılık MSE hatasının değişimi.....	52
Şekil 5.9	ABC algoritması için geri oluşturulan sinyalin ölçüm sayısına karşılık MSE hatasının değişimi.....	52
Şekil 5.10	OMP algoritması ile geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatası.....	53
Şekil 5.11	CoSaMP algoritması ile geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatası	53
Şekil 5.12	PDIP algoritması ile geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatası	54
Şekil 5.13	Klasik ve sezgisel algoritmalar için geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatalarındaki değişim	54
Şekil 5.14	Geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatasının üç boyutlu çubuk grafiğinde gösterilmesi	55
Şekil 5.15	GA kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası.....	56
Şekil 5.16	DE algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası	56
Şekil 5.17	PSO algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası.....	56
Şekil 5.18	ABC algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası	57
Şekil 5.19	OMP algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası.....	57
Şekil 5.20	CoSaMP algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası	58
Şekil 5.21	PDIP algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası.....	58

Şekil 5.22	Klasik ve sezgisel algoritmalar için seyreklik oranı ve MSE'ye bağlı olarak algoritma sonuçlarının değişimi	59
Şekil 5.23	Seyreklik oranı ve MSE'ye bağlı olarak algoritmaların sonuçlarının 3 boyutlu çubuk grafiğinde gösterilmesi	59
Şekil 5.24	GA geri ile geri elde edilmiş sinyal ve orijinal sinyal	63
Şekil 5.25	DE algoritması ile geri elde edilmiş sinyal ve orijinal sinyal.....	63
Şekil 5.26	PSO algoritması ile geri elde edilmiş sinyal ve orijinal sinyal	64
Şekil 5.27	ABC algoritması ile geri elde edilmiş sinyal ve orijinal sinyal	64
Şekil 5.28	CoSaMP algoritması ile geri elde edilmiş sinyal ve orijinal sinyal.	64
Şekil 5.29	OMP algoritması ile geri elde edilmiş sinyal ve orijinal sinyal	65
Şekil 5.30	PDIP algoritması ile geri elde edilmiş sinyal ve orijinal sinyal	65

TABLOLAR LİSTESİ

Tablo 4.1	Test 1 için algoritmaların performans sonuçları.....	32
Tablo 4.2	Test 2 için algoritmaların performans sonuçları.....	35
Tablo 4.3	Test 3 için algoritmaların performans sonuçları.....	37
Tablo 5.1	Kullanılan sezgisel algoritmaların kontrol parametreleri.....	40
Tablo 5.2a	$N=64$, $M=24$, $k=8$ olması durumunda başarılı koşmalar için elde edilen sonuçlar.....	44
Tablo 5.2b	$N=64$, $M=24$ ve $k=8$ olması durumunda başarılı ve başarısız tüm koşmalar için elde edilen sonuçlar..	45
Tablo 5.3a	$N=64$, $M=24$, $k=12$ olması durumunda başarılı koşmalar için elde edilen sonuçlar.....	46
Tablo 5.3b	$N=64$, $M=24$ ve $k=8$ olması durumunda başarılı ve başarısız tüm koşmalar için elde edilen sonuçlar..	46
Tablo 5.4a	$N=64$, $M=48$, $k=8$ olması durumunda başarılı koşmalar için elde edilen sonuçlar.....	47
Tablo 5.4b	$N=64$, $M=48$ ve $k=8$ olması durumunda başarılı ve başarısız tüm koşmalar için elde edilen sonuçlar..	47
Tablo 5.5a	$N=64$, $M=48$, $k=12$ olması durumunda başarılı koşmalar için elde edilen sonuçlar.....	48
Tablo 5.5b	$N=64$, $M=48$ ve $k=12$ olması durumunda başarılı ve başarısız tüm koşmalar için elde edilen sonuçlar..	48
Tablo 5.6	Algoritmaların farklı durumlar için sinyali geri elde etmelerine ait yüzde başarımlar oranları.....	49
Tablo 5.7	Gürültü içeren ölçümlerde seyrek sinyalin geri oluşturulması aşaması için elde edilen sonuçlar	62
Tablo 5.8	Önerilen metodun Chakraborty'nin çalışmasıyla karşılaştırılması	66

GİRİŞ

Tezin Amacı ve Önemi

Örnekleme teoremi sürekli zamanlı işaretler ile ayrık zamanlı işaretler arasındaki temel köprüdür. Bu teorem, Shannon örnekleme teoremi veya Nyquist örnekleme teoremi olarak da adlandırılır. Teorem, temel olarak sınırlı bantlı bir $f(t)$ analog işaretinin yeniden ve bozulmadan elde edilebilmesi için örnekleme işleminin nasıl olması gerektiğini belirtmektedir. Nyquist-Shannon örnekleme kriterine göre bir işareti yeniden elde edebilmek için örnekleme oranının işaretin bant genişliğinin en az iki katı olması gerekmektedir [1]. Pratik uygulamaların çoğunda bu gerekliliği sağlamak oldukça zordur veya sistemin maliyetini ve karmaşıklığını artırmaktadır [2].

İlk olarak 2006 yılında Donoho, Candès, Romberg ve Tao tarafından ortaya çıkarılan Sıkıştırılmış algılama (Compressed Sensing-CS), seyrek sinyallerin gerekli olduğu düşünülen Nyquist/Shannon örnekleme teorilerinde belirtilen orandan daha az sayıda örnekleme oranı ile tekrar elde edilebileceğini ileri sürmektedir [3-5]. Bu teori, yeni görüntü sistemleri, kameralar, sensör ağları, manyetik rezonans görüntüleme, radar hedef tespiti, örnekten sayısala dönüştürücü tasarımı gibi bir çok uygulama alanında kullanılmaktadır.

CS temel olarak üç ana kısımdan oluşur; işareti seyrek olarak temsil etmek (sparse representation), ölçümleme (measurement) ve seyrek işaret geri oluşturma (reconstruction) [6]. İşaretin seyrek olarak ifade edilmesi, işaretin uygun bir domende izdüşümlerinin gösterilmesi prensibine dayanır. Çoğu işaret kendi domeninde ya da dönüşüm (Transform) domeninde seyrek yapılıdır. Fourier dönüşümü, dalgacık (wavelet) dönüşümü, kosinüs dönüşümü (Cosine Transform) işareti seyrek olarak ifade etmek için kullanılan dönüşümlerdir. CS'nin ölçümleme işleminde ise seyrek işaretin

uzunluğunun daha az sayısı kadar ölçüm yapılarak gözlem vektörü oluşturulur. Bu durumun matematik ifadesi Eşitlik 1 ile verilmektedir.

$$y = \Phi x \quad (1)$$

Yukarıdaki denklemde $x \in R^N$ seyrek işaret, $\Phi \in R^{M \times N}$ ($M < N$) ölçüm matrisi ve $y \in R^M$ gözlem vektörünü ifade etmektedir. Ölçüm işleminde, Eşitlik (1)'deki seyrek işaret ve ölçüm matrisi çarpılarak gözlem vektörü oluşturulur. Son kısım olan seyrek işaret geri oluşturma aşaması ise gözlem vektöründen yani ölçümlerden, x işaretinin geri oluşturulmasıdır. x işaretini geri oluşturmak için Eşitlik (1)'deki denklem sistemini çözmek gerekir. Yukarıdaki denklem sisteminin çözümünde üç durum söz konusudur:

- $M=N$ için Φ^{-1} yani Φ matrisinin tersinin olması durumudur. Bu durumda x 'in çözümü vardır ve tektir. x 'in çözümü Eşitlik (2)'deki gibidir.

$$x = \Phi^{-1} y \quad (2)$$

- $M > N$ olduğu durumda, eğer ölçümde gürültü yoksa birinci durumla aynıdır. Gürültü olması durumunda ise, denklemin çözümü yoktur ancak, en küçük kareler yaklaşımıyla çözüme yakınsanabilir.
- $M < N$ için Eşitlik 3'de Φ 'nin sıfır uzayında $y = \Phi(x + z)$ olan sonsuz tane z vektörü vardır. Bu durumda çözüm kümesi sonsuzdur.

$$x' = \Phi^T (\Phi \Phi^T)^{-1} y \quad (3)$$

CS teorisi, $M < N$ olduğu için son durumla alakalıdır [7]. Seyrek işareti geri oluşturma işleminde y vektörü, Φ matrisi ve işaretin seyreklik bilgisinden yararlanılır. CS'de bu sistemin çözümü için çeşitli yöntemler sunulmuştur. Bu yöntemler genel olarak 11 en küçükleme ve açgözlü algoritmalar olarak ikiye ayrılır [8]. Taban Arayışı (Basis Pursuit, BP), Uyumlu Arayış (Matching Pursuit, MP), Dik Uyumlu Arayış (Orthogonal Matching Pursuit, OMP) ve Sıkıştırılmalı Örneklemeli Uyumlu Arayış (Compressive Sampling Matching Pursuit, CoSaMP), Birincil İkili İç Nokta yöntemi (Primal Dual Interior Point-PDIP) yaygın olarak kullanılan geri oluşturma algoritmalarından bazılarıdır [9]. Ayrıca son zamanlarda sezgisel algoritmalar da seyrek geri oluşturma yöntemi olarak kullanılmaktadır. Özellikle bahsedilen klasik yöntemlerin seyrek işareti geri oluşturmada bazı sınırlayıcı durumları vardır. Bu durumlar işaret uzunluğu, ölçüm sayısı ve seyreklik oranına göre değişir. Klasik yöntemlerin işareti geri oluşturmada yetersiz kaldığı durumlarda sezgisel algoritmalar da kullanılabilir.

Seyrek işareti geri oluşturmak için tez çalışmasının ilk bölümünde klasik algoritmalar kullanılmıştır. Tez çalışmasının ilk bölümünde seyrek sinyali geri elde etmek için PDIP, OMP ve CoSaMP algoritmaları incelenmiştir. Bu üç farklı klasik optimizasyon algoritmasının MATLAB ortamında benzetimi gerçekleştirilerek performans karşılaştırması yapılmıştır. Tez çalışmasının devamında seyrek işaretin geri oluşturulmasında popülasyon tabanlı sezgisel algoritmaların kullanımına dayalı bir yöntem önerilmiştir. Önerilen yöntem literatürde yaygın kullanım alanına sahip dört farklı optimizasyon algoritması üzerinde test edilmiştir. Bu algoritmalar; Genetik Algoritma (Genetic Algorithm-GA), Diferansiyel Gelişim (Differential Evolution-DE), Parçacık Sürüsü Optimizasyon (Particle Swarm Optimization-PSO) ve Yapay Arı Koloni (Artificial Bee Colony-ABC) algoritmalarıdır. Popülasyon tabanlı algoritmaların, işareti geri oluşturma performansları kendi aralarında ve klasik yöntemlerle karşılaştırılmıştır. Karşılaştırma işlemi gürültüsüz işaret üzerinde ve değişik oranlarda gürültü eklenmiş işaret temel alınarak bunların istatistiksel analizleri üzerinden gerçekleştirilmiştir.

Tez çalışmasının organizasyonu şu şekilde düzenlenmiştir: Birinci bölümde CS teorisinin matematiksel temeli ve literatürde var olan seyrek işaret geri oluşturma algoritmaları hakkında bilgi verilmektedir. İkinci bölümde, tez çalışmasında kullanılan klasik yöntemlerin yani OMP, CoSaMP ve PDIP algoritmalarının temel adımları anlatılmıştır. Üçüncü bölümde kullanılan sezgisel algoritmalar yani GA, DE, PSO ve ABC algoritmalarının adımları verilerek CS teorisinde seyrek sinyalin geri elde edilmesinde bu algoritmaların kullanımı için önerilen metot açıklanmaktadır. Dördüncü bölümde; test problemleri, algoritmaların performans sonuçları ile bu sonuçlarla ilgili detaylı yorumlar ve son bölümde ise tartışma, sonuç ve öneriler verilmektedir.

1. BÖLÜM

GENEL BİLGİLER VE LİTERATÜR ARAŞTIRMASI

1.1 Sıkıştırılmış Algılama

Bu bölümde sıkıştırılmış algılamanın matematik temeli anlatılmaktadır. Matematikteki norm kavramı CS’de sık kullanılan bir kavramdır. Haberleşme sistemiyle ilgili klasik problemlerde genel olarak l_2 normu olarak da bilinen Öklid normu kullanılmasına rağmen CS algoritmalarının analizinde değişik normlar da kullanılmaktadır. Bir $x = [x_1, x_2, \dots, x_n] \in R^N$ vektörünün p . normu Eşitlik (1.1)’deki gibidir [10].

$$\|x\|_p = (\sum_{i=1}^n |x_i|^p)^{\frac{1}{p}} \quad (1.1)$$

Eşitlik (1.1)’de, x vektörünün p . normu hesaplanırken öncelikle x vektörünün her bir elemanının p . kuvveti alınarak bunlar toplanır ve toplamın p . kökü alınır. Ayrıca $0 < p < 1$ aralığı için Eşitlik (1.1)’in sonucu üçgen eşitliği teoremini sağlamadığı için matematiksel olarak bir norm değildir, ancak CS içinde sıklıkla kullanılmaktadır [8]. Eşitlik (1.2)’de tanımlanan ifade x ’in sıfırcı normu olarak adlandırılır.

$$\|x\|_0 = \text{supp}(x), \text{supp}(x) = \{i: x_i \neq 0\} \quad (1.2)$$

x ’in sıfırcı normu, x vektörünün içindeki sıfıra eşit olmayan elemanlarının toplam sayısıdır. CS teorisine göre bazı işaretleri ve görüntü işaretlerini klasik örnekleme yöntemlerinininkinden daha az bir örnekleme miktarıyla elde edilebilir. Ancak bunun mümkün olması için işlenecek işaretin ve ölçüm matrisinin bazı özellikler ve şartları sağlaması gerekmektedir [11].

1.1.1. Seyrek ve Sıkıştırılabilir İşaretler

Ses, görüntü, sismik işaretler gibi birçok işaret uygun bir tabanda izdüşümü alınarak sıkıştırılmış bir formda saklayabilir. Uygun bir tabanda temsil edilen işaretlerin çoğu örneği sıfır ya da ihmal edilecek kadar küçük olabilmektedir. x işareti $x \in R^{N \times 1}$ şeklinde bir vektör ise bu durumda işaret, $N \times N$ ortonormal taban olan $\psi = \{\psi_1 | \psi_2 | \dots | \psi_N\}$, ω_i sütun vektörleriyle Eşitlik (1.3)'deki gibi şekilde ifade edilebilir [7].

$$x = \sum_{i=1}^N s_i \psi_i \quad \text{ya da} \quad x = \psi \cdot s \quad (1.3)$$

$$s_i = \psi_i^T x \quad (1.4)$$

S , $N \times 1$ sütun vektörü olan Eşitlik (1.4)'de verilen s_i 'nin ağırlık katsayılarıdır. Burada x , zaman ya da uzay domeninde ve s , ψ domeninde bir işaretin eşdeğer gösterimleridir. $k \ll N$ için Eşitlik (1.3)'de sadece k sayıdaki katsayı sıfırdan farklı ve $(N-k)$ sayıdaki s_i katsayıları sıfır ise x işareti k -seyrek (k -sparsity) denir. Eşitlik (1.3)'de k sayıdaki katsayıları büyük ancak çoğu katsayıları küçükse x işareti sıkıştırılabilir bir işarettir. Bu tanıma göre seyrek ve sıkıştırılabilir bir işaretin bilgi içeriğinin çoğu veya tamamı N örnek yerine k örneğinde yer alır ve katsayıların çoğu fazla bir kayıp olmaksızın ihmal edilebilir. CS temel olarak seyrek veya sıkıştırılabilir işaretlerin bu özelliğini kullanır.

İşareti seyrek olarak ifade edildikten sonra $M < N$ olan doğrusal ölçümde, ölçümler/gözlem vektörü Eşitlik (1.5)'deki gibi yeniden ifade edilebilir.

$$y = \phi \cdot x = \phi \cdot \psi \cdot s = \Theta \cdot s \quad (1.5)$$

$\Theta = \phi \cdot \psi$ matrisinin boyutu $M \times N$ dir ve bu ölçüm işlemi adaptif değildir. Yani algılama matrisinin işarete bağlı olarak değişmemesi gerekir. Ayrıca ölçüm esnasında $x \in R^{N \times 1}$ den $y \in R^{M \times 1}$ 'e geçerken önemli bilgiler kaybolmamalıdır. Bu yüzden kararlı bir ϕ ölçüm matrisi tasarlanmalıdır. Dolayısıyla ϕ ölçüm matrisinin ve ψ matrisinin aşağıda belirtilecek özellikleri sağlaması gerekir [11].

1.1.2. Kısıtlı İzometri Özelliği

X , k -seyrek bir işaret olsun. Bütün x işaretleri için ölçüm matrisi kısıtlı izometri özelliğini (Restricted Isometry Property-RIP) sağlamalıdır. δ_k , herhangi bir k değeri için

ϕ matrisinin en küçük izometri katsayısı şeklinde tanımlansın. Eşitlik (1.6) bütün k -seyrek x işaretleri için sağlandığında eğer δ_k , 1'e çok yakın değilse ϕ ölçüm matrisi kısıtlı izometri özelliğini sağlamaktadır [11].

$$(1 - \delta_k) \|x\|_2^2 \leq \|\phi x\|_2^2 \leq (1 + \delta_k) \|x\|_2^2 \quad (1.6)$$

ϕ matrisi kısıtlı izometri özelliğini sağladığı zaman, k -seyrek işaretleri ϕ 'nin sıfır uzayında yer almaz ve işareti yeniden elde edebilmek mümkün olabilir. Eğer kısıtlı izometri özelliğini sağlamazsa Öklid uzaklığı kaybolacağından dolayı işareti yeniden elde etme olasılığı ortadan kalkar [12, 13].

1.1.3. Tutarsızlık

Ölçüm matrisi için diğer bir koşul tutarsızlıktır. Tutarlık (Coherence) iki farklı matrisin herhangi iki elemanı arasındaki korelasyonun bir ölçüsüdür. Bu iki matris iki farklı tabanı/gösterimi ifade edebilir [11].

- ϕ : $\phi_1, \phi_2, \phi_3 \dots \phi_m$ satır vektörleriyle beraber, $M \times N$ boyutlu ölçüm matrisi olsun.
- ψ : $\psi_1, \psi_2 \dots \psi_N$ sütun vektörleriyle beraber, $N \times N$ boyutlu taban dönüşüm matrisi olsun.

Bu iki matrisin tutarlılığı Eşitlik (1.7)'deki gibidir.

$$\mu(\phi, \psi) = \sqrt{n} \cdot \max_{1 \leq k, j \leq n} \phi_k \psi_j \quad (1.7)$$

Burada $\mu(\phi, \psi)$, tutarlık değeri, 1 ile $\sqrt{n}$ arasındadır. Bu değer 1 olduğunda ϕ matrisi ile ψ matrisi arasındaki korelasyon minimum, $\sqrt{n}$ olduğunda ise maksimumdur. CS teorisinde seyrek işareti yeniden oluşturmak için ϕ ölçüm matrisi ve ψ temsil matrisinin düşük tutarlığa sahip olması gerekir. Düşük tutarlı ölçüm/temsil tabanı ikilisine örnek olarak sinüzoidal ve dürtü sinyali verilebilir. Bu ikili bütün boyutlarda tutarsızdır. Ölçüm matrisi, rastgele gauss matrisi seçilirse yüksek bir olasılıkla kısıtlı izometri ve tutarsızlık özellikleri sağlanır [7].

Ölçümleri oluşturduktan sonra gözlem vektöründen seyrek işareti yeniden oluşturmak için seyrek işareti yeniden oluşturma algoritmaları kullanılır. Bu algoritmalar/yöntemler ölçüm sayısı M , ölçüm matrisi ϕ ve eğer işaret farklı bir tabanda seyrek yapıldıysa

taban matrisi ψ 'yi kullanarak N uzunluklu x işaretini ya da eşdeğer olarak x 'in seyrek katsayı vektörü s 'yi elde eder. Seyrek işareti yeniden oluşturmak için kullanılan metotların temelini aşağıda anlatıldığı gibi özetlenebilir [7, 14].

Minimum l_0 yeniden oluşturması: İlk olarak x seyrek işaretindeki sıfır olmayan eleman sayıları hesaplanır. x yeniden oluşturulmuş işaret, Eşitlik (1.8)'deki optimizasyon problemiyle çözülür.

$$x = \operatorname{argmin} \|x\|_0 \text{ öyle ki } y = \Phi x \quad (1.8)$$

l_0 normun çözümü kararsız ve NP-zordur (NP-hard).

Minimum l_1 yeniden oluşturması: Bu durumda l_1 normuna bağlı optimizasyon Eşitlik (1.9)'deki gibi çözülür.

$$x = \operatorname{argmin} \|x\|_1 \text{ öyle ki } y = \Phi x \quad (1.9)$$

l_1 normu, k -seyrek bir işareti yüksek bir olasılıkla $M \geq ck \log(N/k)$ sayıda bağımsız ve özdeş olarak dağıtılmış (independent and identically distributed-iid) gauss ölçümüyle oluşturabilir [3, 4]. L_1 minimizasyonu konveks bir problemdir ve doğrusal programlama yöntemiyle çözülebilir.

Minimum l_2 yeniden oluşturması: Eşitlik (1.10)'da verilen l_2 optimizasyon probleminin kapalı formda çözümü Eşitlik (1.11)'daki gibidir.

$$x = \operatorname{argmin} \|x\|_2 \text{ öyle ki } y = \Phi x \quad (1.10)$$

$$x = \Phi^T (\Phi \Phi^T)^{-1} y \quad (1.11)$$

L_1 minimizasyon yöntemi neredeyse hiçbir zaman k -seyrek bir çözüm bulmamakta, seyrek olmayan çözümler bulmaktadır. Yapılan araştırmalara göre eğer ölçüm matrisi kısıtlı izometri özelliğini sağlıyorsa l_0 norm optimizasyonu yerine l_1 norm optimizasyonu kullanılabilir. [5, 12]. Literatürde var olan seyrek yeniden oluşturma algoritmalarının çoğu l_1 norm optimizasyon yöntemini kullanır.

1.1.4. Literatür araştırması

CS'de seyrek işaretin geri oluşturulması için orijinal problem, genel olarak NP-zordur. Bu NP-zor problem için literatürde çeşitli yöntemler önerilmiştir. Bu yöntemler çok

daha fazla sınıfa ayrılabilmesine rağmen temelde iki sınıfta incelenmektedirler: L_1 optimizasyonu ve açgözlü yöntemler. Literatürde bahsi çok geçen ve pratik uygulamalarda en yaygın olarak kullanılan yöntemler genel olarak bu iki kategoriden birine dâhil olur. L_1 optimizasyon problemi ve bu problemi çözen yöntemler şu şekildedir. Eşitlik (1.9)'da gösterilen l_1 optimizasyon probleminin çözümü için Doğrusal Programlama (LP-Linear Programming) kullanılmaktadır [8]. LP metoduyla çözüm için çeşitli algoritmalar bulunmaktadır. Bunlardan biri bu tez çalışmasında da kullanılan PDIP metodudur [15]. Bu metodun detaylı açıklaması bir sonraki bölümde verilmiştir. Diğer bir problem aşağıdaki Eşitlik (1.12)'de verilen sınırlandırılmış l_1 minimizasyonudur [8].

$$\|\Phi x' - y\|_2 \leq \epsilon \text{ koşulu ile } \min \|x'\|_1 \quad (1.12)$$

CS'de kullanılan bu metodun çözümü için Nesterov'un algoritması (Nesterov's algorithm-NESTA) [16] algoritması önerilmiştir. Bu sınıfa ait problemler ayrıca PDIP [15] veya Log Barrier [17] metoduyla çözülebilen ikinci dereceden koni programlama (second-order cone programming-SOCP) sınıfına girmektedir. Ancak bunlara daha iyi çözüm sunan alternatif bir yol olarak NESTA önerilir.

Araştırmacılar, l_1 optimizasyonunun yerine l_1 - l_2 optimizasyonu üzerine de etkili algoritmalar önermişlerdir [8].

$$\min_x \left(\frac{1}{2} \cdot \|\Phi x - y\|_2^2 + \lambda \|x\|_1 \right) \quad (1.13)$$

Eşitlik (1.13)'de verilen bu problem, Hızlı İteratif Küçülme-Eşik Algoritma (Fast Iterative Shrinkage-Thresholding Algorithm-FISTA) [18], sabit nokta devamlılığı (Fixed-Point Continuation FPC) [19], Bregman [20], (Gradient Projection Sparse Reconstruction-GPSR [21], (two-step iterative shrinkage-thresholding) TWIst [22], (Sparse reconstruction by separable approximation) Sparsa [23] algoritmalarıyla çözülmektedir.

Literatürde yaygın olarak kullanılan diğer bir yöntem açgözlü algoritmalarlardır. Açgözlü yöntemler destek kümesini birer birer oluşturarak CS seyrek sinyalin geri elde edilmesi probleminin çözümüne yakınsamaya çalışır. Açgözlü algoritmalar, optimal çözüme yakınsamayıp genel olarak bölgesel minimumu bulmasına rağmen bazı durumlarda l_1 optimizasyon yöntemlerinden daha iyi performans sergiler. CS'de kullanılan açgözlü

yaklaşım sergileyen algoritmalar şunlardır. En basit açgözlü algoritma MP'dir [24]. OMP, MP'nin bir iyileştirilmiş versiyonudur [25]. OMP ile MP arasındaki temel fark, OMP aynı indeksi iki defa seçmez çünkü artık vektör, $r[k]$, daha önce seçilen ölçüm matrisinin sütun vektörüne diktir. Seyrek işaret geri oluşturmadaki diğer açgözlü algoritmalar OMP tabanı üzerinden bazı değişiklikler yapılarak sunulmuştur. Eğimli arayış (Gradient pursuit-GP) algoritmasında OMP'nin arayış yönü değiştirilerek hesaplama maliyeti azaltılmaya çalışılmıştır [26]. MP ya da OMP her iterasyonda ölçüm vektöründen sadece bir sütun vektörü seçer. Bu da k -seyrek bir işaret için k iterasyon sürer. Birden fazla sütun vektörü seçerek bu algoritmayı hızlandırmak için (stagewise orthogonal matching pursuit-stOMP) sunulmuş ve OMP'deki adım güncellemesi farklı bir şekilde geliştirilmiştir [27]. Düzenlenmiş dik uyumlu arayış (Regularized orthogonal matching pursuit) ROMP algoritması her iterasyonda birden fazla sütun seçen bir alternatif açgözlü algoritmadır [28], [29]. Diğer bir açgözlü algoritma olan CoSaMP, Needel ve Tromp tarafından CS için önerilmiştir [30]. Bu algoritma, stOMP ve ROMP gibi birden fazla indeks seçmektedir. Ayrıca bu algoritmada yeni olan budama (pruning) adımı vardır. Bu adım da öncelikle seçilen $2k$ tane adımı en küçük kareler yaklaşımıyla k 'ya düşürür. Bu tezde kullanıldığı için, CoSaMP algoritmasının bir sonraki bölümde detaylı adımı verilmiştir.

Yukarıdaki algoritmalar çeşitli araştırmalarla etkisi kanıtlanan algoritmalarlardır. Ancak bahsedilen algoritmaların çoğu özellikle gürültü olduğu durumlarda sınırlı bir yakınsamaya sahiptir. Doğadan esinlenilerek ortaya çıkarılmış sezgisel algoritmalar küresel araştırma stratejileri ve üstün performanslarıyla optimizasyon problemlerinin çözümünde oldukça popülerdir. Bu algoritmaların konveks olmayan CS problemi için kullanılması oldukça isabetlidir [31]. Bu yüzden literatürde son 3-4 yıldır sezgisel algoritmalar da CS'de seyrek işaret geri oluşturma yöntemi olarak kullanılmaktadır. Sezgisel algoritmalar ya kendi başına ya da diğer bahsedilen yöntemlerle beraber CS'de kullanılmaktadır. Xinpeng et al. tarafından yapılan çalışma [32], bu alanda sezgisel algoritmaların kullanıldığı ilk çalışmalardan biridir. Bu çalışmada M-SISR diye adlandırılan bir yöntem, seyrek geri oluşturma probleminin çözümü için önerilmiş ve test edilmiştir. Çalışmada sezgisel algoritma olarak PSO kullanılmış, PSO'nun başlangıç çözümünün oluşturulmasında q -eşikleme (q -thresholding) yöntemi kullanılmıştır. Ayrıca M-SISR yönteminde güncelleme stratejisi olarak PSO ile beraber,

CoSaMP’da kullanılan budama yöntemi kullanılmıştır. Shah et al. tarafından yapılan çalışmada, GA ile beraber PCD algoritması kullanılmış olup hibrit bir çözüm mekanizması oluşturulmuştur ve detaylı olmamakla beraber önerilen yöntem bazı yöntemlerle kıyaslanmıştır [33]. Fengmin ve Shanhe tarafından yapılan başka bir çalışmada hibrit benzetimli tavlama eşikleme (Hibrid Simulated Annealing Thresholding-HSAT) denilen bir yöntem önerilmiştir. Bu yöntemde özyinelemeli eşikleme (Iterative Thresholding-IT) algoritmasının hesaplamada basit oluşu avantajı ile beraber benzetimli tavlama algoritmasının küresel araştırma yeteneği bir araya getirilerek hibrit bir yapı oluşturulmuştur. Önerilen yöntemin performansı normal işaret ve görüntü işareti üzerinde test edilmiştir [34]. Bir konferans çalışmasında, GA ve BFO birleştirilerek bir yöntem sunulmuştur [35]. Bir diğer makalede Xinpeng et al. ile benzer olarak açgözlü algoritmalarından yararlanılmıştır. Farklı olarak PSO yerine benzetilmiş tavlama (Simulated Annealing-SA) kullanmıştır. Bu yöntem, HSAMMV, yapılan test sonuçlarına göre güçlü bir global araştırma yeteneğine ve iyi bir geri elde etme performansına sahiptir [36]. Lei et al. tarafından yapılan bir çalışmada eşyerellik (homotopy) metodu ve evrimsel programlamanın avantajları bir araya getirilerek bir yöntem sunulmuştur [37]. Bu hibrit mekanizmaların dışında CS ile ilgili sezgisel algoritmaların değişik formatlarda sunulduğu çalışmalarda bulunmaktadır [38–41].

Ayrıca sezgisel algoritmalar çoklu amaç fonksiyonu şeklinde ve iki evreli olarak da CS’de kullanılmaktadır. Çoklu amaç fonksiyonu olan, Li et al. tarafından yapılan çalışmalardan birinde önerilen yöntemde, STeMO, amaç fonksiyonlarından biri ölçüm hatası diğeri seyreklik indirgeme terimidir [42]. Bu çalışmadan önceki çalışmalarda bu iki amaç fonksiyonu tek bir terimde kullanılmıştır. Bu iki amaç fonksiyonun çözümünde evrimsel algoritma kullanılmıştır. Bunun haricinde, yine birden fazla amaç fonksiyonunun optimize edilmesiyle ilgili çalışmalar da vardır [43].

Sezgisel algoritmaların seyrek işaret geri oluşturmada kullanılmasıdaki bir diğer metot iki evreli yaklaşımdır. Liu et al. bu metodu kullanmışlardır [31]. Bahsedilen iki evreli yeniden oluşturma metodunda birinci evre olan geri oluşturma evresi, görüntü bloklarının bütün yönlerde atomik boyutlarının tahminini yapan genetik algoritma tasarlanmıştır. İkinci geri oluşturma evresinde ise, birinci evrenin çıktısı olan alt sözlükteki görüntü bloklarının alt sözlüğündeki atomik kombinasyonlarından daha iyi olanını araştırmak için klonal seçim algoritması kullanılmıştır. Liu et al. bu yöntemle

seyrek geri oluřturma yntemlerindeki belirsizlięi ve kararsızlıęı azalttıklarını belirtmiřlerdir. Yapılan test sonularına gre nerdikleri bu yntem klasik agzlu ve eřikleme algoritmalarından daha iyi performans gstermektedir. Sezgisel algoritmaların kullanıldıęı iki evreli yaklařıma dięer bir rnek alıřma, Eyll 2017’de Zhou et al. tarafından yayınlanmıřtır [44]. Bu alıřmada Birinci evrede; ayrıřım tabanlı ok ama fonksiyonlu evrimsel algoritma, l_1 normunun optimizasyonuna grbz bir zmretmek iin uygulanmıřtır. İkinci evrede; ileri tabanlı bir seme yntemi uygulanmıřtır.



2. BÖLÜM

SEYREK İŞARET YENİDEN OLUŞTURMA ALGORİTMALARI

CS ilk ortaya çıktığından beri literatüre birçok seyrek işareti yeniden oluşturma algoritmaları sunulmuştur. Bu algoritmaların hepsi farklı bakış açılarıyla değişik şekilde sınıflandırılabilmelerine rağmen genel olarak l_1 minimizasyonu ve açgözlü (greedy) algoritmalar şeklinde iki sınıfta değerlendirilmektedir. Tez çalışmasında öncelikle seyrek işareti geri oluşturmada kullanılan l_1 minimizasyon yöntemi ve açgözlü algoritmalar incelenmiş ve bu algoritmaların MATLAB ortamında benzetimleri yapılarak kendi aralarında bir karşılaştırması verilmiştir. l_1 minimizasyon yönteminden PDIP yöntemi ve açgözlü algoritmalarından da OMP ve CoSaMP kullanılarak bu üç algoritmanın seyrek işaret geri oluşturma performans sonuçları kıyaslanmıştır. Tezin bu bölümünde karşılaştırması yapılan bu algoritmaların teorik altyapısı anlatılmaktadır.

2.1. l_1 minimizasyonu

$x \in R^N$ işareti ψ tabanında seyrek bir işaret ve Φ ölçüm matrisi ψ matrisiyle tutarsız olsun. Ayrıca $\theta = \phi \cdot \psi$, RIP özelliğini de sağlamış olsun. CS teorisine göre x işareti $M > O(k \log(N)) \ll N$ ölçüm sayısı ile yeniden elde edilebilir [7]. x işaretinin sadece k elemanı sıfır olmadığı için gözlem vektörü y , ϕ matrisinin sütun vektörlerinin k tanesinin doğrusal birleşimidir. Dolayısıyla x işaretinin yeniden oluşturulması için, y 'nin oluşmasında katkısı bulunan ϕ matrisinin sütun vektörlerinin bulunması gerekir. CS'de seyrek işareti geri oluşturma temeline Eşitlik (2.1) ile belirtilen l_0 minimizasyonu bulunmaktadır.

$$x = \operatorname{argmin} \|x\|_0 \text{ öyle ki } y = \phi x \quad (2.1)$$

Ancak önceki bölümde bahsedildiği gibi l_0 probleminin çözümü NP-zor'dur ve çözümü bulmak için çok fazla döngü gerekmektedir. Ancak yukarıda bahsedilen koşullar altında

yani RIP ve tutarsızlık sağlandığında l_1 minimizasyonu, l_0 minimizasyonunun yerine kullanılabilir. Eşitlik (2.2) bir konveks optimizasyon problemidir ve doğrusal programlama ile çözülebilir. Bu tezde doğrusal programlama yöntemi olarak PDIP metodu kullanılmıştır.

$$x = \operatorname{argmin} \|x\|_1 \text{ öyle ki } y = \phi x \quad (2.2)$$

2.1.1. Birincil İkili İç Nokta

Eşitlik (2.2) ile verilen doğrusal programlama problemi Eşitlik (2.3)'deki gibi ifade edilebilir:

$$\min_z 1_{2n}^T \text{ subject to } \Phi_0 z = y \text{ and } z \geq 0 \quad (2.3)$$

Eşitlikte T devrik (transpoze) alınması durumunu temsil etmektedir. $\Phi_0 = [\Phi, -\Phi]$ 'dir.

Tez çalışmasında Eşitlik (2.3)'deki doğrusal problemi çözmek için PDIP yöntemi kullanılmıştır. Eşitlik (2.2)'nin minimal noktası z^* 'deki Kuhn Tucker koşulları göz önüne alınırsa $v^* \in R^N$ ve $\lambda^* \in R^N$ şeklinde iki vektör,

$$1_{2N}^T + \Phi_0^T v^* - \lambda^* = 0, \quad (2.4)$$

$$\lambda_i^* z_i^* = 0, i = 1, 2, \dots, 2N, \quad (2.5)$$

$$\Phi_0 z^* = y, \quad (2.6)$$

$$z^* \geq 0, \quad \lambda^* \geq 0. \quad (2.7)$$

yazılabilir [8, 17]. PDIP algoritması yukarıdaki Kuhn Tucker koşullarını Newton iterasyon yöntemiyle çözmektedir. Bu metoda aynı zamanda iç nokta (interior point) metodu da denir çünkü $z[k], v[k]$ ve $\lambda[k]$ benzetilmiş vektörleri Eşitlik (2.7) ile tanımlanmış bölgenin içerisinde kalmaktadır. Kısaca PDIP algoritması yukarıdaki doğrusal olmayan denklemleri çözerek optimum z^* 'yi bulur. Çözüm metodu klasik Newton yönteminin uygulandığı olarak düşünülebilir. $z[k], v[k], \lambda[k]$ iç noktasında sistem doğrusallaştırılarak çözülür.

MATLAB araç kutusu olan “ l_1 -magic”de lineer program problemin çözümü için kullanılan PDIP algoritması bulunmaktadır. Bu tez çalışmasında l_1 minimizasyonu için bu araç kutusu (toolbox) kullanılmıştır [15, 45].

2.2 Açgözlü Algoritmalar

Açgözlü algoritmalar herhangi bir seyrek sinyali geri oluşturma problemini adım adım yinelemeli yani iteratif olarak açgözlü bir şekilde çözer. Her bir iterasyon için gözlem vektörü y ile en çok ilintili olan ϕ 'nin sütun vektörü seçilir. Daha sonra sütunların katkısı rezidü vektörden çıkarılır. Bu işlem doğru sütunlar bulununcaya kadar yani artık vektör sıfır oluncaya kadar devam eder. Aynı zamanda bu bulunan doğru sütun kümesi seyrek x işaretinin sıfır olmayan elemanlarının indisleridir. Bu indisler kullanılarak en küçük kareler yöntemiyle seyrek x işareti bulunur. En çok kullanılan açgözlü algoritmalar OMP, CoSaMP ve MP'dir. Bu tez çalışmasında OMP ve CoSaMP kullanıldığı için bu iki yöntem detaylı olarak alt bölümlerde anlatılmaktadır.

2.2.1. OMP (Dikey Eşleştirme Takibi) Algoritması

OMP algoritmasında öncelikle y ile en çok ilintili olan ϕ matrisinin sütun vektörü seçilir ve her iterasyonda bu seçilen sütunun katkısı ilk başta y vektörüne eşit olan rezidü vektörden çıkarılır. Rezidü vektör sıfır veya belli bir değerin altına düştüğü zaman doğru sütun kümesi bulunmuştur. OMP algoritması k iterasyon sürer ve her iterasyonda bir indis bulur. Son olarak da bu k tane indisler en küçük kareler yaklaşımıyla beraber kullanılarak seyrek x işareti geri oluşturulur. OMP algoritmasının adımları aşağıda verilmektedir [25, 46, 47].

Girdiler:

- Φ : $M \times N$ boyutlu ölçüm/örnekleme matrisi.
- k : x sinyalinin sıfırdan farklı eleman sayısı yani seyreklik derecesi.
- y : $M \times 1$ boyutlu gözlem vektörü.

Çıktılar:

- Λ_K : sıfırdan farklı elemanların indislerinin kümesi
- Φ_K : ölçüm matrisinin seçilmiş sütun matrisleri.
- a_k : yaklaşık olarak y vektörü.
- $r_k = y - a_k$: rezidü vektör

Başlangıç Değerleri:

- $r_0 = y$
- $\Lambda_K = \emptyset$
- $t = 1$, İterasyon sayısı

- $\Phi_0 = \emptyset$

İterasyonlar:

1. Aşağıdaki optimizasyon problemini çözen λ_t indislerini bul. λ_t, Φ matrisinin r_{t-1} ile en çok ilintili sütunudur.

$$\lambda_t = \operatorname{argmax}_{j=1..N} |\langle r_{t-1}, \varphi_j \rangle| \quad (2.8)$$

2. İndis kümesini ekle.

$$\Lambda_K = \Lambda_K \cup \{\lambda_t\} \quad (2.9)$$

3. Seçilen sütunları ekle.

$$\Phi_t = [\Phi_{t-1} \quad \varphi_{\lambda_t}] \quad (2.10)$$

4. Aşağıdaki denklemde en küçük kareler yöntemini kullanarak x_t 'yi bul

$$x_t = \operatorname{argmin}_{x'} \|y - \Phi_t x'\|_2 \quad (2.11)$$

5. Yeni gözlem vektörünü yaklaşık olarak hesapla.

$$a_t = \Phi_t x_t \quad (2.12)$$

6. Rezidü vektörü güncelle.

$$r_t = y - a_t \quad (2.13)$$

7. Eğer $t < K$ ise t 'yi artır ve yeni iterasyon için **adım 1**'e git.

2.2.2. CoSaMP (Sıkıştırılmış Algılamalı Eşleştirme Takibi) Algoritması

Bu algoritmada öncelikle Φ^T ölçüm matrisinin devriği ile gözlem vektörü y çarpılarak hata hesaplanır. Daha sonra seyreklik değerinin iki katı kadar en çok ilintili sütun vektörü seçilir ve bunlardan bir destek kümesi oluşturulur. Daha sonra en küçük kareler yaklaşımı kullanılarak destek kümesinin eleman sayısı $2k$ 'dan k 'ya düşürülür. Son olarak rezidü vektör hesaplanır. Rezidü vektör belirlenen hatanın altına düşünceye kadar bu döngü devam ettirilir. En son döngüdeki en küçük kareler yaklaşımının sonucu, geri oluşturulmuş seyrek x işaretidir. CoSaMP algoritmasının detaylı adımları aşağıda anlatılmaktadır [8, 30].

Girdiler:

- Φ : $M \times N$ boyutlu ölçüm/örnekleme matrisi.
- k : x sinyalinin sıfırdan farklı eleman sayısı yani seyreklik derecesi.

- y : $M \times 1$ boyutlu gözlem vektörü.

Çıktı: $x[t + 1]$: Yeniden oluşturulmuş k -seyrek x işareti.

Başlangıç değerleri:

- $x[0] = 0$
- $r[0] = y$, Rezidü vektörü.
- $\Lambda = \emptyset$, Destek kümesi.
- $t=0$, İterasyon sayısı.

Algoritma adımları:

- 1- Seyreklik seviyesinin iki katı kadar yani $2k$ kadar olacak şekilde en güçlü destek kümesini seç.

$$\Lambda = \Lambda \cup \text{supp}(H_{2k}|\Phi^T r[k]|) \quad (2.14)$$

- 2- En küçük kareler yaklaşımıyla z vektörünü hesapla.

$$z = \Phi_{\Lambda}^{\dagger} r, \quad (2.15)$$

$$\Phi_{\Lambda}^{\dagger} = (\Phi_{\Lambda}^T \Phi_{\Lambda})^{-1} \Phi_{\Lambda}^T \quad (2.16)$$

- 3- En büyük k tanesi hariç hepsini sıfıra eşitle.

$$x[t + 1] = H_k(z) \quad (2.17)$$

- 4- Artık vektörü aşağıdaki Eşitlik (2.18) ve Eşitlik (2.19)'u kullanarak güncelle.

$$\Lambda = \text{supp}(x[t + 1]) \quad (2.18)$$

$$r[t + 1] = y - \Phi x[t + 1] = y - \Phi_{\Lambda} x_{\Lambda}[t + 1] \quad (2.19)$$

- 5- İterasyon sayısını artır

$$t = t + 1 \quad (2.20)$$

- 6- Koşul sağlandığında döngüden çık.

Koşul, maksimum iterasyon sayısı veya minimum hata değeridir.

Yukardaki durumlar için H_k , eşikleme operatörünü temsil etmektedir. Bu operatör en büyük k elemanı hariç bütün elemanları sıfıra eşitler.

3. BÖLÜM

SEZGİSEL ALGORİTMALAR

Popülasyon tabanlı optimizasyon algoritmaları çözülmesi zor olan optimizasyon problemlerine optimuma yakın çözümler sunar. Bütün popülasyon tabanlı algoritmaların ortak özelliği, problemin muhtemel çözümü olan popülasyonun amaç fonksiyonuna bağlı olan bazı operatörlerin çözüme uygulanmasıyla ortaya çıkmasıdır. Bu operatörler sayesinde popülasyon, çözüm uzayındaki iyi olan çözümlere doğru ilerler. Evrimsel algoritmalar ve sürü zekalı algoritmalar popülasyon tabanlı algoritmaların iki önemli sınıfıdır [48].

3.1. Genetik Algoritma

Genetik algoritma (Genetic Algorithm-GA) popülasyon tabanlı algoritma grubu içerisinde değerlendirilen sezgisel araştırma metodudur ve J. Holland tarafından 1970'lerin başlarında ortaya çıkarılmıştır [49]. Doğal seleksiyon ile canlılarda bulunan genetik gelişimin benzetişimini gerçekleştirmektedir. Evrimsel algoritmalarda olduğu gibi araştırma uzayındaki bazı çözümlerin oluşturduğu başlangıç koşullarını kullanır. Daha sonra doğal seleksiyon işlemi ile birlikte genetik operatörlerden olan çaprazlama ve mutasyon, gelecek jenerasyondaki yeni çözümleri üretmek amacıyla kullanılır. Gelecek jenerasyonun oluşturulmasında her bireye uygunluk değerlendirme işlemi uygulanır. Birbirini takip eden jenerasyonların değerlendirilmesi ve geliştirilmesini temel alan bu döngü optimum bir çözüm bulununcaya kadar devam eder. Bu tezde kullanılan GA'nın temel adımları aşağıda verilmektedir:

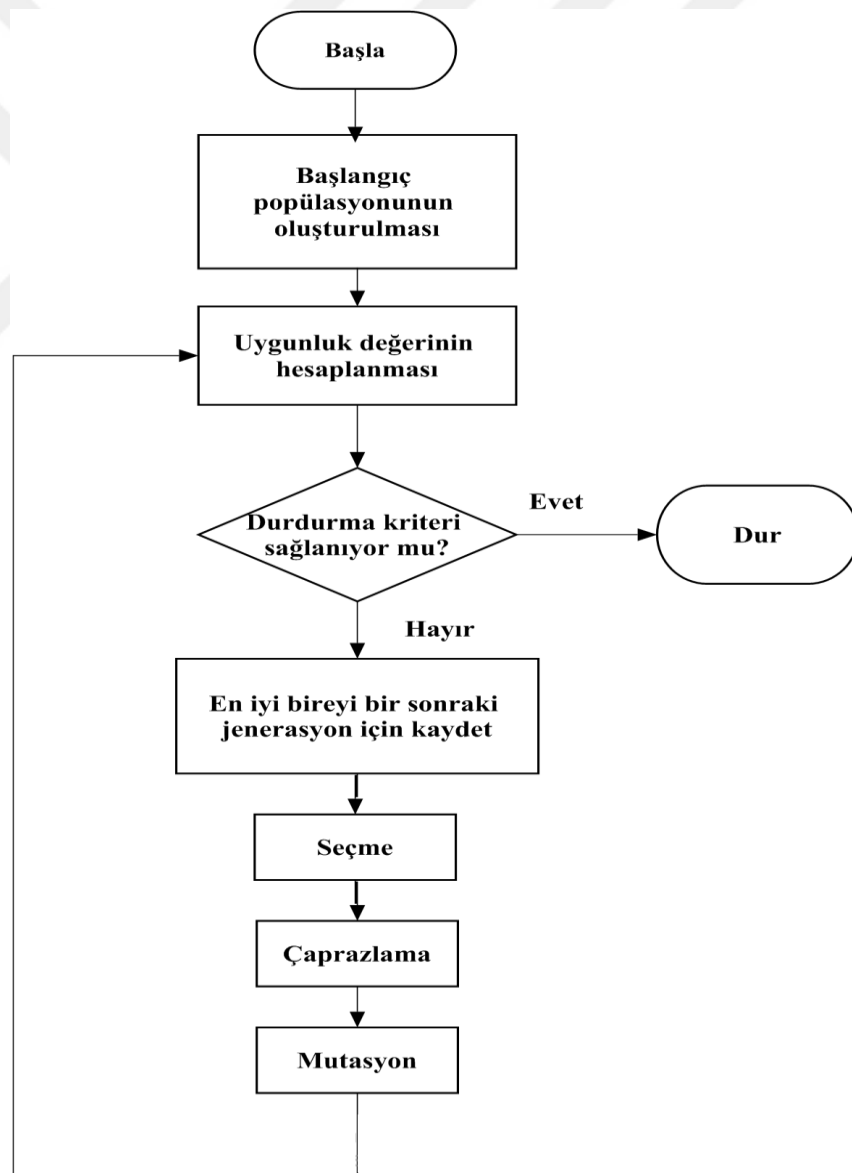
- 1) Başlangıç popülasyonu oluştur.
- 2) Popülasyonu uygunluk fonksiyonuna göre değerlendir.

Durdurma kriteri sağlanıyorsa araştırmayı durdur.

Sağlanmıyorsa durdurma kriteri sağlanıncaya kadar aşağıda belirtilen işlemlere devam et.

- 3) Elitizm uygula yani en iyi bireyi gelecek popülasyon için sakla.
- 4) Bir sonraki popülasyon için doğal seleksiyon işlemini uygula.
- 5) Çaprazlama işlemini uygula ve yeni bireyler üret.
- 6) Mutasyon işlemini uygula yani yeni bireylerde rastgele değişim oluştur.
- 7) Yeni popülasyonu değerlendir.

GA algoritmasının akış şeması Şekil 3.1’de verilmiştir.



Şekil 3.1. GA akış şeması.

GA'nın adımlarında uygulanan temel operasyonlar aşağıda kısaca anlatılmaktadır:

Seleksiyon: Bu işlem doğal seçme işlemi olarak da adlandırılır. Koşullara en uygun bireyin hayatta kalmasını temel almaktadır. Seçme işlemi doğada çevre tarafından, yapay sistemlerde de amaç fonksiyonu ve diğer uygunluk değerlendirme işlemleriyle yapılır.

Çaprazlama: Eşleştirme havuzunda bulunan genlerden birer çifti seçilir ve çaprazlama operatörü bu iki yapıdan yeni genler oluşturmak için kullanılır. Bu tezde kullanılan GA'da, uygunluk değerine göre en iyi iki birey seçilir. Bu seçilen en iyi iki birey, her defasında rastgele seçilen bir noktadan itibaren diğer bireyle birleştirilerek yeni bireyler oluştururlar.

Mutasyon: Oluşturulan yeni bireylerde mutasyon oranına göre rastgele değişim meydana gelebilir. Mutasyon yeni, görülmemiş ve araştırılmamış çözüm elemanlarının bulunmasını sağlar. Aynı zamanda mutasyon işlemiyle GA'nın bölgesel bir optimum çözüme takılması engellenir.

Elitizm: Bu tez çalışmasında kullanılan GA'da, bir önceki popülasyondan en iyi olan birey gelecek nesil için saklanır.

3.2. Diferansiyel Gelişim Algoritması

Diferansiyel Gelişim (Differential Evolution-DE) algoritması 1995 yılında Storn ve Kenneth tarafından geliştirilen popülasyon tabanlı sezgisel bir algoritmadır [50]. DE algoritması çaprazlama, mutasyon ve seleksiyon gibi genetik benzer algoritmalar kullanmaktadır. DE ile GA arasındaki temel fark; GA yeni çözümler üretirken çaprazlama işlemine bağlıdır, DE'de ise mutasyon işlemine bağlıdır. DE'nin temel çalışma metodu, popülasyondan rastgele seçilmiş olan çözüm çiftlerinin farkına dayanır [51]. Diferansiyel gelişim algoritması yeni bireyler üretmek için mutasyon ve rekombinasyon işlemlerini Eşitlik (3.2)'de görüldüğü gibi beraber kullanmaktadır. Bu denklemde görüldüğü gibi mutasyon ve rekombinasyon (yeniden birleştirme) işlemlerinde yeni oluşturulan vektörlerin en az bir parametre de olsa oluşturulduğu vektörlerden farklı olacak şekilde tasarlanmıştır. Seleksiyon kısmında, bir önceki adımda oluşan bireyler uygunluk fonksiyonu kullanılarak ebeveyn vektörüyle kıyaslanır. Eğer yeni bireyler daha iyiye, popülasyondaki ebeveyn vektörüyle yer

değiştirir. Eğer daha iyi değilse, ebeveyn vektör popülasyonda muhafaza edilir. Yeni birey oluşturma döngüsü, amaç fonksiyonu durdurma kriteri sağlanıncaya kadar devam eder. D optimize edilmek istenen bir fonksiyonun parametre sayısı, NP popülasyon sayısı, F ölçekleme oranı, CR çaprazlama oranı ve $Gmax$, jenerasyon sayısı olmak üzere temel DE algoritmasının adımları aşağıda verilmektedir [52].

- 1) Kontrol parametrelerinin $(D, Gmax, NP \geq 4, F \in (0,1 +), CR \in [0,1])$ değerlerinin ve sınırlarının (x_j^L, x_j^U) atanması.
- 2) Rastgele olarak $[x_j^L, x_j^U]$ aralığında başlangıç parametre değerlerini Eşitlik (3.1)'i kullanarak oluştur.

$$x_{j,i,G} = x_j^L + rand_j[0,1].(x_j^U - x_j^L) \quad (3.1)$$

$$i = (1,2, \dots NP), j = (1,2 \dots D), \quad G = 0, \quad rand_j[0,1] \in [0,1]$$

REPEAT

- 3) Mutasyon ve Rekombinasyon işlemlerini uygula:

$$r_1, r_2, r_3 \in (1, 2, \dots NP), \quad r_1 \neq r_2 \neq r_3 \neq i$$

$$j_{rand} \in (1,2 \dots D)$$

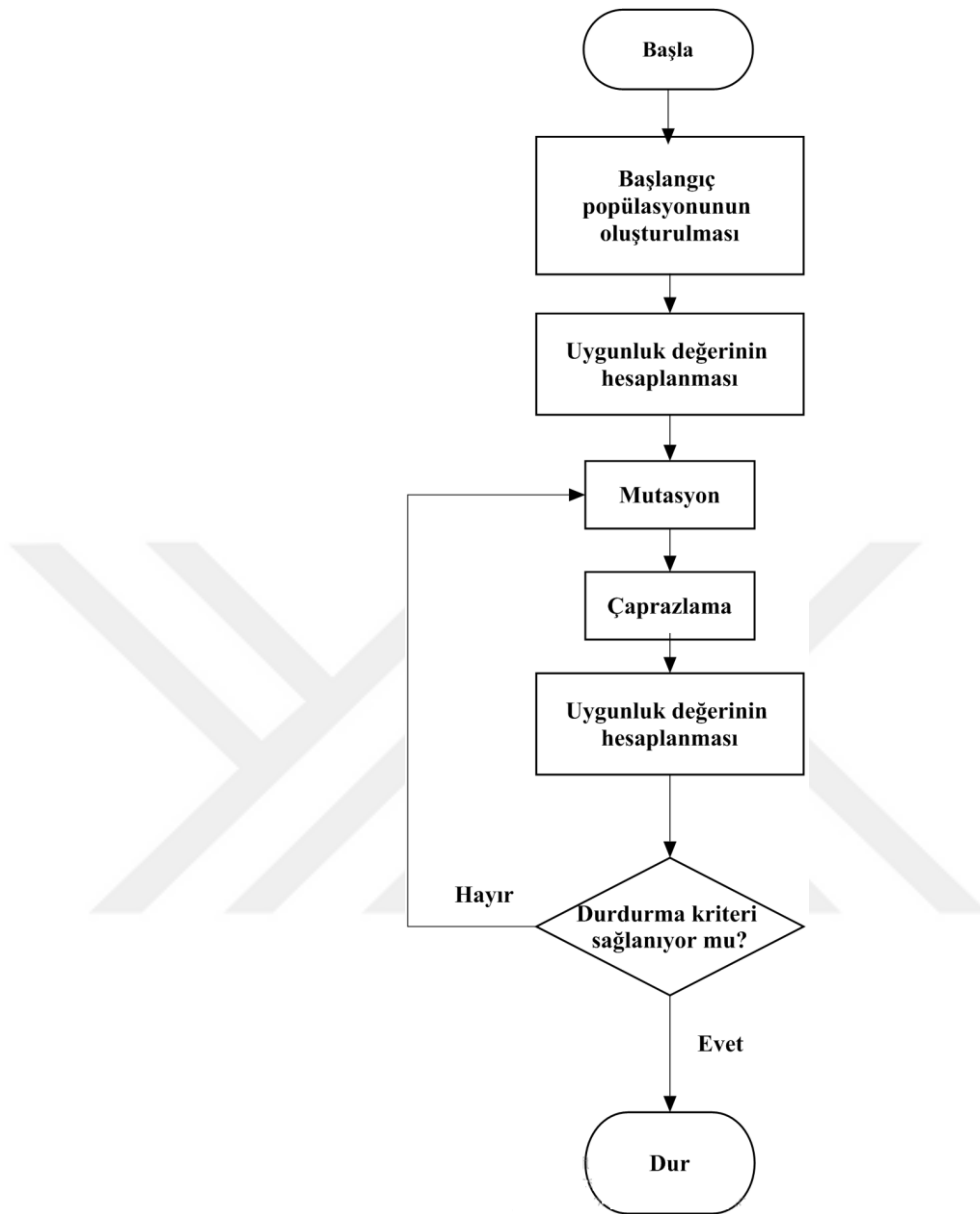
$$u_{j,i,G+1} = \begin{cases} x_{j,r_3,G} + F(x_{j,r_1,G} - x_{j,r_2,G}) & \text{eğer } rand_j[0,1] < CR \text{ veya } j = j_{rand} \\ x_{j,i,G} & \text{diğer durumlarda} \end{cases} \quad (3.2)$$

- 4) Seleksiyon işlemini uygula:

$$x_{i,G+1} = \begin{cases} u_{i,G+1} & \text{eğer } f(u_{i,G+1}) \leq f(x_{i,G}) \\ x_{i,G} & \text{diğer durumlarda} \end{cases} \quad (3.3)$$

UNTIL (durdurma kriteri sağlanıncaya kadar).

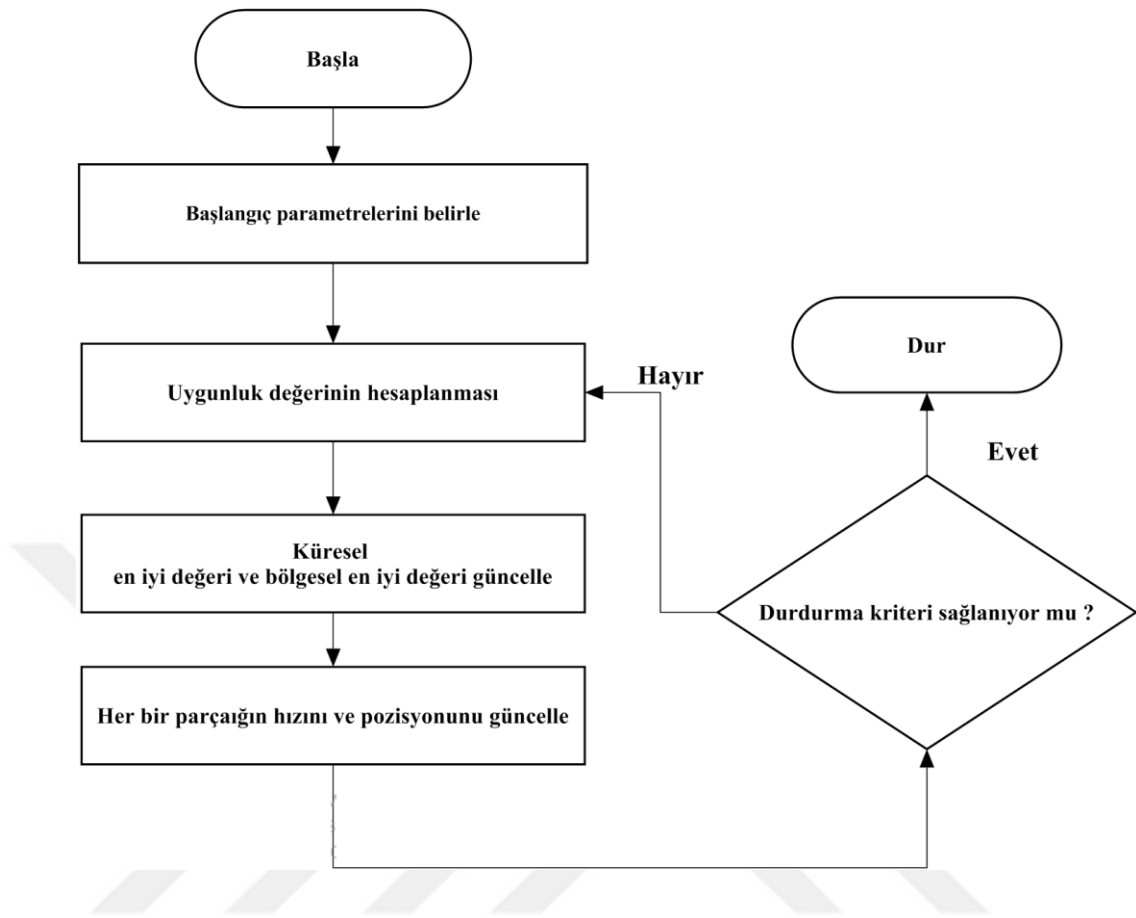
DE algoritmasının akış şeması Şekil 3.2'de verilmiştir.



Şekil 3.2. DE Algoritması akış şeması.

3.3. Parçacık Sürüsü Optimizasyon Algoritması

Kennedy ve Eberhart tarafından 1995 yılında ortaya çıkarılan parçacık sürüsü optimizasyon PSO, popülasyon tabanlı stokastik optimizasyon algoritmasıdır [53]. PSO algoritmasında parçacıklar, problemin araştırma uzayında hareket ederek problemi optimize etmeye çalışırlar. Bir parçacığın bulunduğu yer, optimizasyon probleminin çözümüne adaydır. Sürüdeki her parçacık, daha iyi yerler (çözümler) bulabilmek için hızlarının güncellemesini kuş sürülerinin davranışlarını model alarak değiştirir. PSO algoritmasının akış şeması Şekil 3.3’de verilmiştir.



Şekil 3.3 PSO Algoritması akış şeması.

PSO'nun adımları aşağıda verildiği gibidir [54].

- 1) Parçacıkların başlangıç popülasyon vektörleri araştırma uzayında rastgele konum ve hızlarla L boyutlu araştırma uzayı üzerinde oluştur.
- 2) **Repeat**
- 3) Her parçacık için uygunluk fonksiyonunu değerlendir.
- 4) Parçacığın uygunluğunu $pbest_i$ ile değerlendir. Eğer karşılaştırılan parçacık daha iyi ise $pbest_i$ 'yi karşılaştırılan değere eşitle ve L boyutlu uzayda, $\vec{p}_i$ karşılaştırılan değer pozisyonu $\vec{x}_i$ 'ye eşit olmaktadır.
- 5) Bu zamana kadarki en başarılı parçacığı tespit et ve indeksini g değerine ata.
- 6) Parçacığın hız ve pozisyonunu aşağıdaki Eşitlik (3.4)'e göre değiştir.

$$\begin{cases} \vec{v}_i \leftarrow \vec{v}_i + \vec{U}(0, \varphi_1) \oplus (\vec{p}_i - \vec{x}_i) + \vec{U}(0, \varphi_2) \oplus (\vec{p}_g - \vec{x}_i) \\ \vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i. \end{cases} \quad (3.4)$$

- 7) **Until:** Eğer durdurma kriteri sağlanıyorsa döngüden çık. (Durdurma kriteri, belirlenen minimum uygunluk değeri ya da maksimum iterasyon sayısıdır).

Not:

- $\vec{U}(0, \varphi_1)$: Her iterasyonda parçacık için $[0, \varphi_1]$ aralığında rastgele ve uniform dağıtılmış sayı vektörlerini temsil eder.
- $\oplus$: Bileşenler düzeyinde (component-wise) çarpımı.

3.4. Yapay Arı Koloni Algoritması

Derviş Karaboğa tarafından 2005’de literatüre sunulan Yapay Arı Koloni (Artificial Bee Colony-ABC) algoritması, nümerik problemlerin optimizasyonunda kullanılan metasezgisel bir algoritmadır [55]. Algoritma bal arılarının zeki yiyecek arama davranışlarını taklit eder. Gürbüz ve esnek bir algoritma olan ABC algoritması diğer sezgisel algoritmalarla göre daha az kontrol parametresi kullanmaktadır. ABC algoritmasının diğer sezgisel algoritmalarla test problemlerindeki performansının karşılaştırılması Karaboğa D. ve Akay B. tarafından ayrıntılı olarak gösterilmiştir [48]. Algoritmanın temel adımları aşağıda verildiği gibidir:

- 1) Başlangıç popülasyonu, $x_{i,j}$ ’yi üret.
- 2) Popülasyonu değerlendir.
- 3) Çevrim=1

4) Repeat

- 5) (3.5) eşitliğini kullanarak, $v_{i,j}$ ile yeni çözümler üret ve onları değerlendir.

$$v_{i,j} = x_{i,j} + \Theta_{i,j}(x_{i,j} - x_{m,j}) \quad (3.5)$$

- m : i ’nin komşuluğunda bir çözüm.
 - Θ : $[-1, 1]$ aralığında rastgele bir sayı.
- 6) x_i ve v_i arasında açgözlü bir seçim yap.

- 7) Eşitlik (3.6)'i kullanarak uygunluk fonksiyonları aracılığıyla x_i çözümleri için P_i olasılık değerlerini hesapla.

$$P_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \quad (3.6)$$

Çözümlerin uygunluk değerlerinin hesaplanması için Eşitlik (3.7) kullanılmaktadır

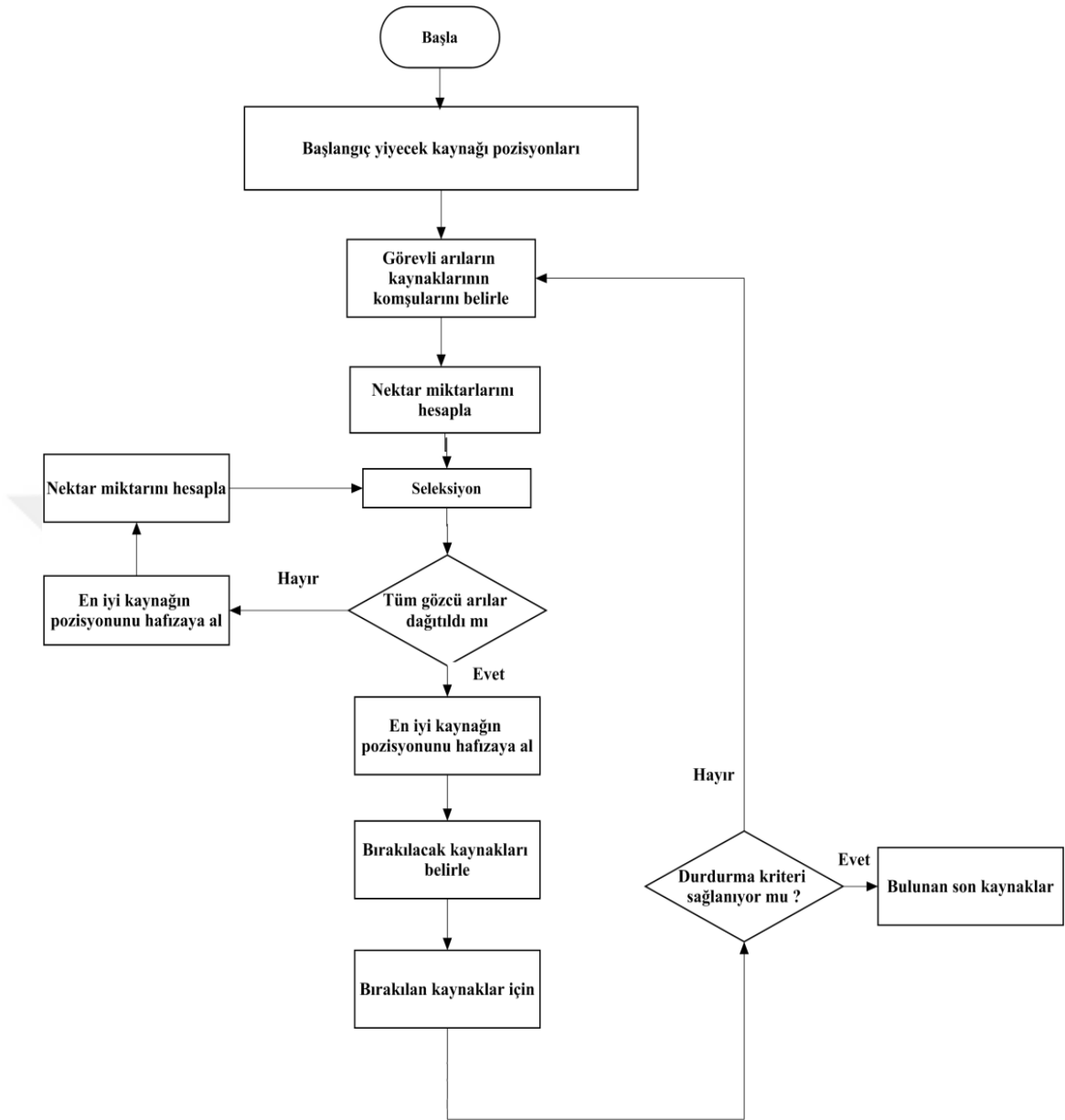
$$fit_i = \begin{cases} 1/(1 + fit_i) & \text{if } f_i \geq 0 \\ 1 + abs(f_i) & \text{if } f_i < 0 \end{cases} \quad (3.7)$$

- 8) x_i çözümlerinden gelen gözcü arılar için yeni çözümler olan v_i 'yi üret ve onları değerlendir.
- 9) x_i ve v_i arasında gözcü arılar için açgözlü seçim (greedy selection) sürecini uygula.
- 10) Terkedilmiş çözümü tespit et eğer varsa Eşitlik (3.8)'i kullanarak kaşif arıları için rastgele üretilen yeni x_i çözümleriyle değiştir.

$$x_{i,j} = min_j + rand(0,1) * (max_j - min_j) \quad (3.8)$$

- 11) Buraya kadar oluşturulan en iyi çözümleri kaydet.
- 12) Çevrim = çevrim + 1
- 13) **Until** (çevrim = maksimum çevrim sayısı)

ABC algoritmasının akış şeması Şekil 3.4'de verilmiştir.



Şekil 3.4. ABC algoritması akış şeması.

3.5. Önerilen Metot

Seyrek x işaretinin sıfır olmayan elemanlarını bulan bu algoritmalar diğer çoğu metotta olduğu gibi gözlem vektörü y , ölçüm matrisi Φ ve k seyreklilik derecesi bilgilerini kullanmaktadır. Sonuç olarak da x orijinal işaret olmak üzere yeniden oluşturulmuş x' sinyalini bulmaktadır. Tez çalışmasında kullanılan sezgisel algoritmaların temel adımları üst bölümlerde açıklanmıştır. Öncelikle hepsinde ortak olan başlangıç popülasyonu oluşturulurken k tane indis oluşturulur. Aynı zamanda bu indisler seyrek x

işaretinin sıfır olmayan elemanlarının indisleridir. Daha sonra bu popülasyon indisleri Eşitlik (3.9), (3.10), (3.11) kullanılarak değerlendirilir.

$$\text{Uygunluk fonksiyonu} = ||y - y'||_2 \quad (3.9)$$

$$y = \Phi x \quad (3.10)$$

$$y' = \Phi x' \quad (3.11)$$

Yukarıdaki denklemde y' 'nü hesaplamak için, aşağıdaki Eşitlik (3.12) ve (3.13) denklemleri kullanılır.

$$x' = \Phi_{\Lambda}^{\dagger} y \quad (3.12)$$

$$y' = \Phi \Phi_{\Lambda}^{\dagger} y \quad (3.13)$$

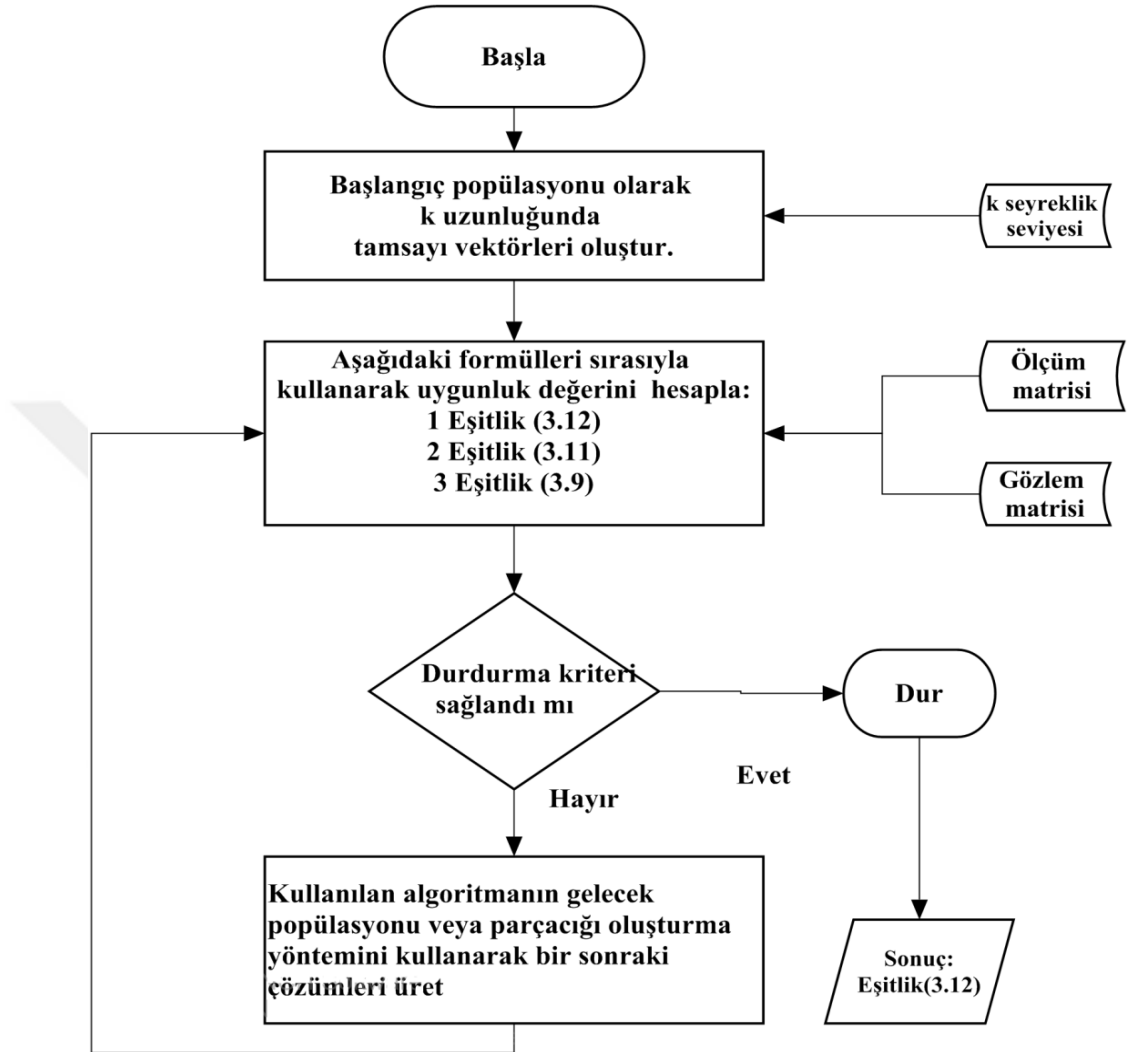
Sıfır olmayan elemanların indislerinin kümesi, $\Lambda \subset \{ 1, 2, \dots, N \}$ dir.

- $\Phi_{\Lambda}: M_{x|\Lambda}$, Λ kümesi tarafından indekslenen Φ matrisinin sütunlarının seçilmesiyle oluşan matris,
- $\Phi_{\Lambda}^{\dagger} = (\Phi_{\Lambda}^T \Phi_{\Lambda})^{-1} \Phi_{\Lambda}^T$, Φ_{Λ} Matrisinin Moore-Penrose pseudo inversidir.

Kullanılan algoritmalar, Eşitlik (3.12) ve Eşitlik (3.13)'i Eşitlik (3.11)'de yerine konulduğunda, Λ kümesiyle gösterilen sıfır olmayan indisleri optimize etmektedir. Diğer bir ifadeyle, x işaretinin sadece k tane sıfır olmayan elemanı olduğu için gözlem vektörü olan y , ölçüm matrisinin k tane sütun vektörünün doğrusal birleşimidir. x işaretini geri elde etmek için y vektörünün oluşumuna katkı sağlayan sütunları bulmak gereklidir. OMP tekniğinde olduğu gibi y vektörüyle en çok ilintili olan Φ matrisinin sütun vektörleri seçilir. Yöntemin akış şeması Şekil 3.5'de verilmektedir.

Önerilen yöntemde seçme işlemi popülasyon tabanlı algoritmalarla yapılmaktadır. Seçme işlemi yapılırken, y vektörüne katkısı olan ölçüm matrisinin sütun vektörü bulunduğu uygunluk değeri düşeceği için bu işlemi popülasyon tabanlı algoritmalarla yapmak isabetli olmaktadır. Daha sonra, geri oluşturulan x' işareti gözlem vektörü y ile bulunan sütun vektörlerinin Moore Penroose sözde tersi (psudeo inverse) alınarak Eşitlik (3.11)'deki gibi hesaplanır. Daha sonra Φ matrisi ve x' kullanılarak y' vektörü hesaplanır. Son olarak da uygunluk değeri Eşitlik (3.9)'daki gibi hesaplanarak, buna göre bir sonraki popülasyon oluşturulur. Doğru destek kümesi

bulduğunda, uygunluk değeri yaklaşık olarak sıfır olur ve doğru geri elde edilmiş x' işareti Eşitlik (3.12)'deki gibidir.



Şekil 3.5. Önerilen metot akış şeması.

Önerilen yöntemin genel adımları aşağıda verilmiştir.

- 1) Başlangıç popülasyonunun oluşturulması.
 - Popülasyon sayısı kadar k uzunluğunda tamsayı vektörleri oluşturulur.
- 2) Eşitlik (3.9)'daki uygunluk fonksiyonunu kullanarak popülasyonu değerlendir.
 - Bulunan indis numaralarına karşılık gelen ölçüm matrisinin sütun vektörlerinin sözde tersini alarak gözlem vektörüyle çarp ve x' vektörünü Eşitlik (3.12)'den hesapla.
 - x' vektörünü ölçüm matrisiyle çarparak y' vektörünü Eşitlik (3.11)'i kullanarak hesapla.

- Eşitlik (3.9)'u kullanarak uygunluk değerini hesapla.
- 3) Durdurma kriteri sağlanıyorsa dur.
- 4) Durdurma kriteri sağlanmıyorsa, kullanılan algoritmanın gelecek popülasyonu veya parçacığı oluşturma yöntemi kullanılarak bir sonraki çözümleri üret.
- 5) Üretilen çözümleri değerlendir ve **adım 3** e git.

Bu tez çalışmasında GA, DE, PSO, ABC algoritmaları CS'de seyrek işareti geri oluşturmak için önerilen yöntemle göre kullanılmıştır. Uygulama metodu bu dört algoritma için aynıdır.



4. BÖLÜM

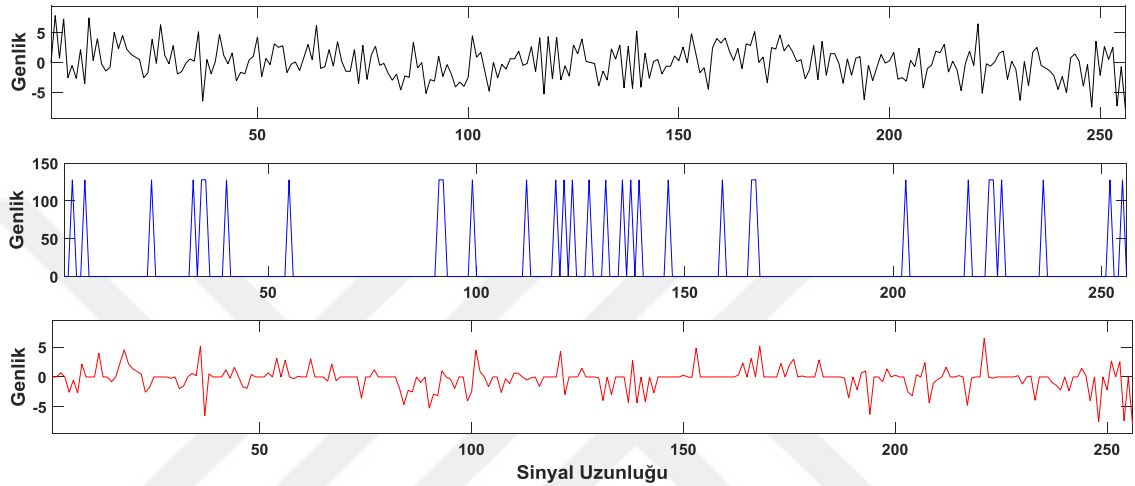
KLASİK ALGORİTMALARIN BENZETİM SONUÇLARI

Tez çalışmasının bu bölümünde seyrek işareti tekrar oluşturmada kullanılan açgözlü algoritmalar ile PDIP algoritmalarının MATLAB ortamında karşılaştırması yapılmış ve bunlara ait elde edilen sonuçlar karşılaştırmalı olarak verilmiştir. Sinüzoidal bir sinyalin Fourier dönüşümü alınarak seyrek bir işaret olarak ifade edilebilmesi mümkündür. Sinyali seyrek olarak ifade ettikten sonra RIP ve tutarsızlık özelliğini de sağlaması için ölçüm matrisi, “rastgele gauss” matrisi olarak üretilmiştir [4, 11, 56]. Daha sonra seyrek işaret ve ölçüm matrisi çarpılarak gözlem vektörü oluşturulmuş yani $M < N$ olması durumunda N uzunluklu işaret, M sayısı kadar örneklenmiştir. Gözlem vektörü y oluşturulduktan sonra “PDIP”, “OMP” ve “CoSaMP” algoritmaları kullanılarak x' sinyalinin geri oluşturulmuş hali bulunmuştur. Aynı işaret ve örnekleme sayısı ile kullanılan üç algoritma üç farklı test problemi üzerinde değerlendirilmiş ve performansları karşılaştırılmıştır. Birinci probleminin sinyal uzunluğu 256, sinyalin sinüzoidal sayısı 16, örnekleme sayısı 128’dir. İkinci problemin sinyal uzunluğu 256, sinyalin sinüzoidal sayısı 8, örnekleme sayısı 128’dir. Son problemin özellikleri ise; sinyal uzunluğu 256, sinyalin sinüzoidal sayısı 8, örnekleme sayısı 64’tür. Alt bölümlerde klasik algoritmaların kullanılması durumunda elde edilen performans sonuçları detaylı bir şekilde verilmektedir.

4.1. Test 1

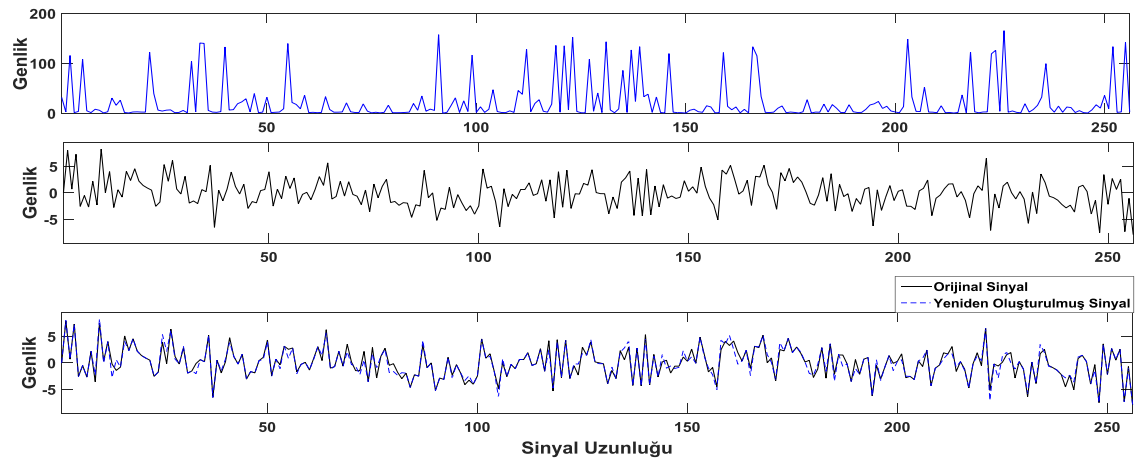
Gerçekleştirilen ilk uygulamada 256 örnek uzunluğa sahip orijinal işaret kullanılmıştır. 16 örneklik sinüzoidal bileşene sahip olan işaret 128’le örneklenmiştir. Şekil 4.1.’de orijinal işaret verilmektedir. Şekilden anlaşılacağı gibi işaret seyrek bir işaret değildir. Seyrek olarak ifade etmek için işaretin Fourier dönüşümü alınmıştır. Fourier dönüşümü sonucunda elde edilen işaret yine Şekil 4.1’de gösterilmektedir. Yine Şekil 4.1 üzerinde

orijinal sinyalden örneklenmiş sinyal gösterilmektedir. Elde edilen işaret, sadece 32 tane elemanı sıfır olmayan 256 uzunluklu seyrek bir işarettir. Bu seyrek işaret, sinyal uzunluğunun yarısı kadar olan $M=128$ ile örneklenip PDIP, OMP ve CoSaMP algoritmalarıyla geri elde edilmiştir. Yani kullanılan algoritmalara ait tüm değerlendirmeler $N=256$, $M=128$, $k=32$ için gerçekleştirilmiştir.



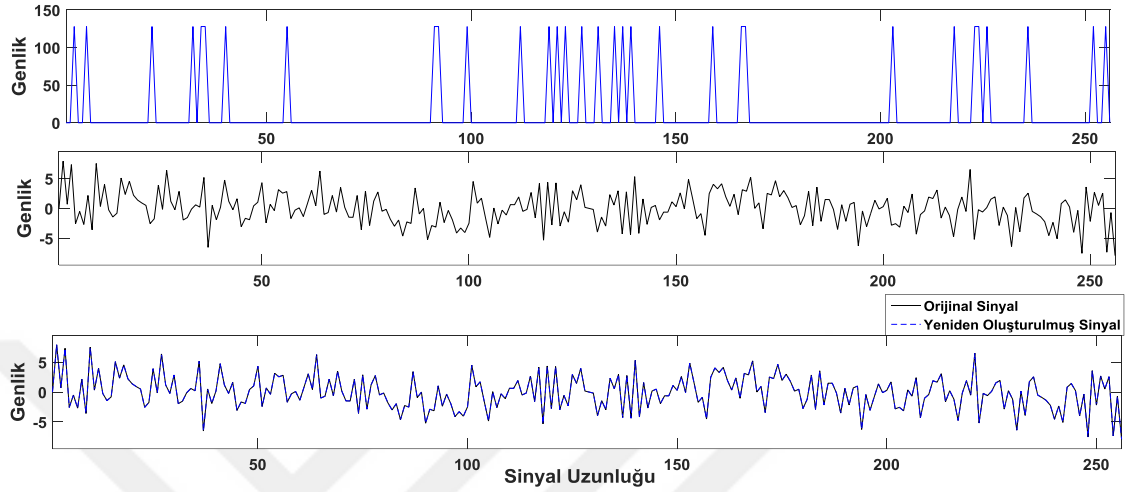
Şekil 4.1. $N=256$, $M=128$, $k=32$ için sırasıyla orijinal sinyal, sinyalin Fourier dönüşümü ve orijinal sinyalin örneklenmiş şekli.

Şekil 4.2’de PDIP yöntemi ile elde edilmiş katsayılar, elde edilen bu katsayıların ters Fourier dönüşümü ile yeniden oluşturulmuş orijinal sinyal ve orijinal sinyalle yeniden elde edilen sinyalin üst üste çizdirilmiş hali gösterilmektedir.



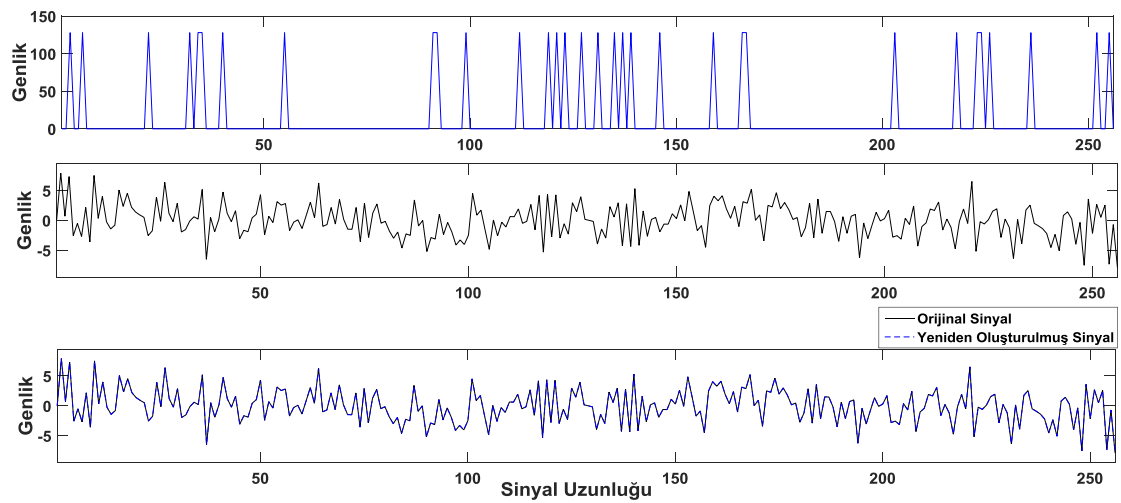
Şekil 4.2. $N=256$, $M=128$, $k=32$ için sırasıyla PDIP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.

Şekil 4.3’de OMP algoritması ile elde edilmiş katsayılar, elde edilen bu katsayıların ters Fourier dönüşümü ile sinyali yeniden oluşturulmuş şekli ve orijinal sinyalle yeniden elde edilen sinyalin üst üste çizdirilmiş hali gösterilmektedir.



Şekil 4.3. $N=256$, $M=128$, $k=32$ için sırasıyla OMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.

Şekil 4.4’de CoSaMP algoritması ile elde edilmiş katsayılar, elde edilen bu katsayıların ters Fourier dönüşümü ve orijinal sinyalle yeniden elde edilen sinyalin üst üste çizdirilmiş hali gösterilmektedir.



Şekil 4.4. $N=256$, $M=128$, $k=32$ için sırasıyla CoSaMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.

Tablo 4.1. Test 1 için algoritmaların performans sonuçları.

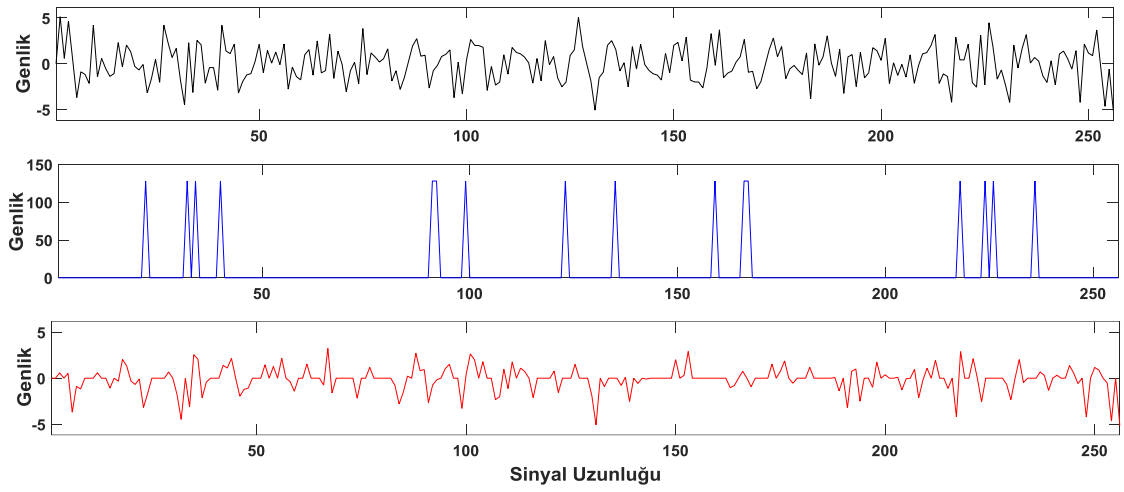
Performans parametreleri	PDIP	OMP	CoSaMP
Test koşulları	$N=256, M=128, k=32$		
MSE	0.6338	2.41×10^{-27}	2.89×10^{-27}
İterasyon sayısı	28	32	1
Koşma süresi (s)	0.1581	0.030263	0.017036

Her üç algoritma ile çizdirilen Şekil 4.2, 4.3 ve 4.4 incelenirse, Şekil 4.2’de orijinal ve geri oluşturulan sinyalin grafiği tam olarak örtüşmemektedir. Şekil 4.3 ve Şekil 4.4’de ise OMP ve CoSaMP’in geri elde ettiği sinyalin grafiği tam olarak örtüşmektedir. Bunlarda MSE hatası 10^{-27} seviyesinde olduğu için geri oluşturulan sinyalle orijinal sinyal arasında fark yoktur.

Tablo 4.1’de kullanılan test problemi için üç algoritmanın performans sonuçları verilmiştir. Tablodan anlaşılabileceği gibi OMP ve CoSaMP algoritmaları ile sinyal nerdeyse hatasız olarak geri elde edilmiştir. İki algoritmada da hata, yaklaşık 10^{-27} seviyesindedir. Fakat PDIP yönteminde elde edilen sinyalin MSE hatası 0.6338’dir. Diğer iki yönteme göre hata daha büyük olarak bulunmuştur. CoSaMP bir (1) iterasyonda sonuca ulaşırken, OMP tüm durumlarda seyreklik seviyesi kadar iterasyonda yani otuz iki (32) iterasyonda sonuca ulaşmıştır. Koşma süreleri değerlendirildiğinde de OMP ve CoSaMP algoritmalarının kullanıldığı durumlarda PDIP algoritmasına göre çok daha kısa sürede sonuca ulaşılmıştır. OMP ve CoSaMP algoritmaları karşılaştırıldığında ise CoSaMP’in OMP’nin iyileştirilmiş versiyonu olduğundan genelde OMP’den daha iyi sonuç vereceği beklenebilir. Dolayısıyla üç yöntem kendi içinde değerlendirildiğinde MSE, iterasyon sayısı ve koşma süresi açısından CoSaMP algoritması diğer iki algoritmaya göre daha iyi sonuçlar üretmiştir.

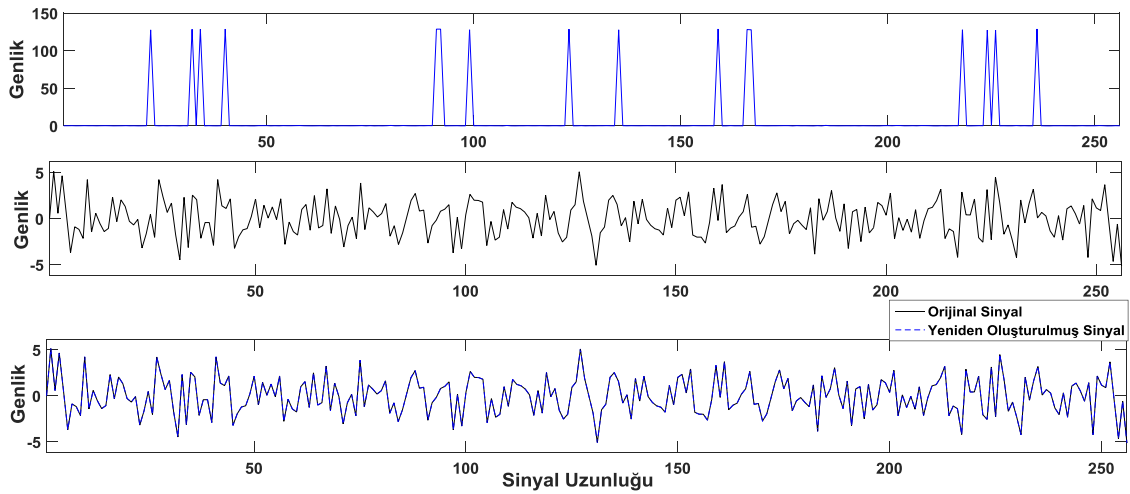
4.2. Test 2

Bu testte, ilk testte incelenen problem için orijinal sinyalin sinüzoidal sayısının yarıya düşürülmesi durumunda algoritmaların sinyali geri elde etmedeki performansları incelenmiştir. Yani $N=256$, $M=128$ ve $k=16$ alınması durumunda inceleme gerçekleştirilmiştir. İkinci test için kullanılan orijinal sinyal, orijinal sinyalin Fourier katsayıları ve orijinal sinyalin ölçülmüş örnek değerlerine ait grafikler Şekil 4.5’de gösterilmektedir.



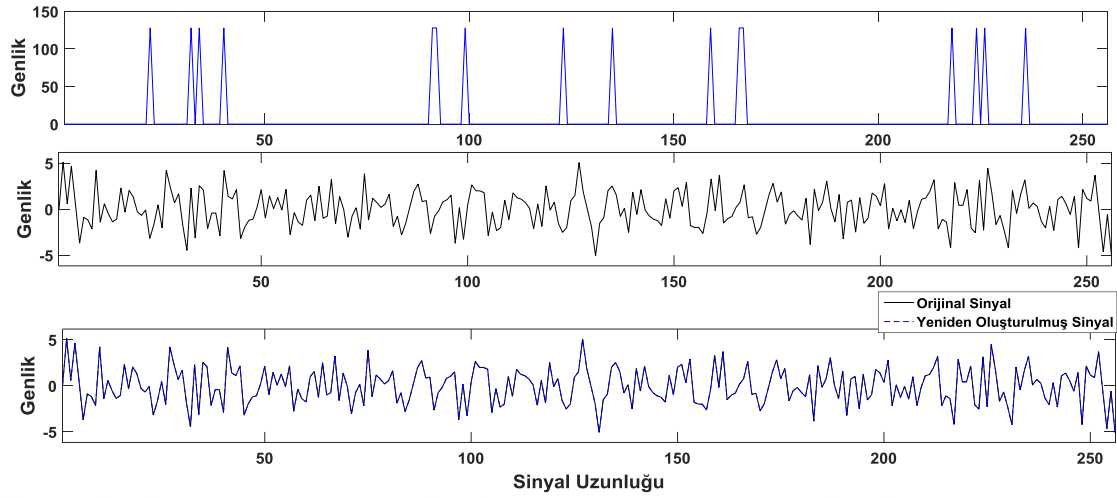
Şekil 4.5. $N=256$, $M=128$, $k=16$ için sırasıyla orijinal sinyal, sinyalin Fourier dönüşümü ve orijinal sinyalin örneklenmiş şekli.

Şekil 4.6'da PDIP yöntemi ile elde edilmiş sinyal, elde edilen bu katsayıların ters Fourier dönüşümü ile orijinal sinyal ve orijinal sinyalle yeniden elde edilen sinyalin üst üste çizilmiş hali gösterilmektedir.



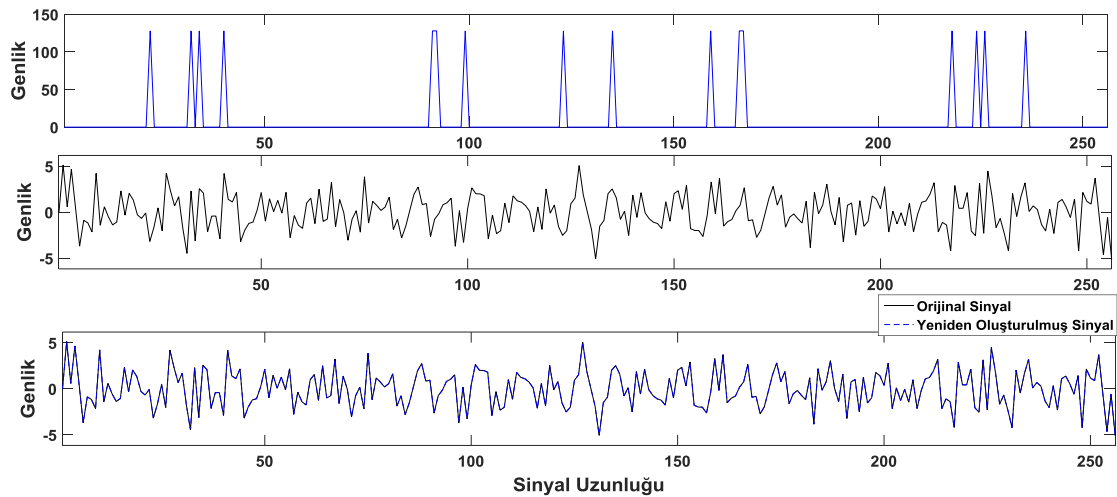
Şekil 4.6. $N=256$, $M=128$, $k=16$ için sırasıyla PDIP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.

Şekil 4.7'de OMP algoritması ile elde edilmiş katsayılar, elde edilen bu katsayıların ters Fourier dönüşümü yani yeniden oluşturulmuş orijinal sinyal ve orijinal sinyalle yeniden elde edilen sinyalin üst üste çizdirilmiş hali gösterilmektedir.



Şekil 4.7. $N=256$, $M=128$, $k=16$ için sırasıyla OMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.

Şekil 4.8’de CoSaMP algoritması ile elde edilmiş katsayılar, elde edilen bu katsayıların ters Fourier dönüşümü yani yeniden oluşturulmuş orijinal sinyal ve orijinal sinyalle yeniden elde edilen sinyalin üst üste çizilmiş hali gösterilmektedir.



Şekil 4.8. $N=256$, $M=128$, $k=16$ için sırasıyla CoSaMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.

Her üç algoritma ile çizdirilen Şekil 4.6, 4.7 ve 4.8 incelenirse; PDIP, OMP ve CoSaMP’ın geri elde ettiği sinyalin grafiği tam olarak örtüştüğü görülmektedir.

Tablo 4.2. Test 2 için algoritmaların performans sonuçları.

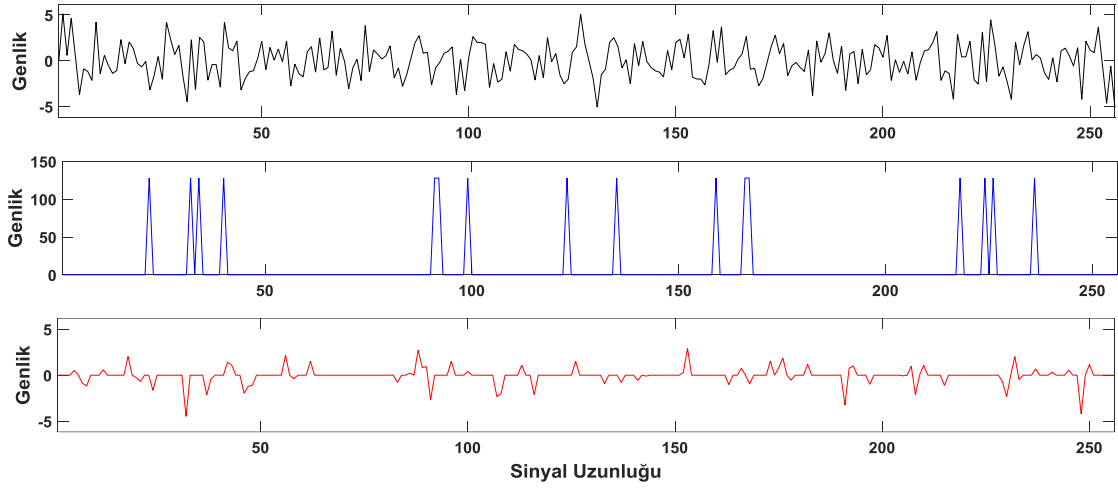
Performans parametreleri	PDIP	OMP	CoSaMP
Test koşulları	$N=256, M=128, k=16$		
MSE	9.79×10^{-05}	1.01×10^{-27}	1.05×10^{-27}
İterasyon sayısı	11	16	1
Koşma süresi (s)	0.0836	0.015424	0.014335

Tablo 4.2’de bu problem için elde edilen üç algoritmanın performans sonuçları verilmiştir. Tablodan görüldüğü gibi algoritmaların hepsi sinyali geri elde etmiştir. OMP ve CoSaMP algoritmalarının sinyali elde etmedeki performanslarının birbirine çok yakın neredeyse aynı olduğu tablodan görülmektedir. Hata bu durumda 10^{-27} ’ler mertebesinde. PDIP algoritmasında 10^{-5} ’ler seviyesinde az da olsa hata vardır. Yine bu test için OMP ve CoSaMP algoritmalarının koşma süreleri PDIP algoritmasından daha iyidir. Buradaki diğer bir konu OMP ve CoSaMP algoritmalarının iterasyonları farklı olmasına rağmen koşma sürelerinin birbirine eşit olmasıdır. OMP algoritması her seferde bir tane indisi hesaplarken CoSaMP seyreklik seviyesinin iki katı kadar hesaplar ve yarısını çıkarır. Bu testte OMP ve CoSaMP algoritmaları bu işlemleri yaklaşık olarak aynı sürede yapmıştır.

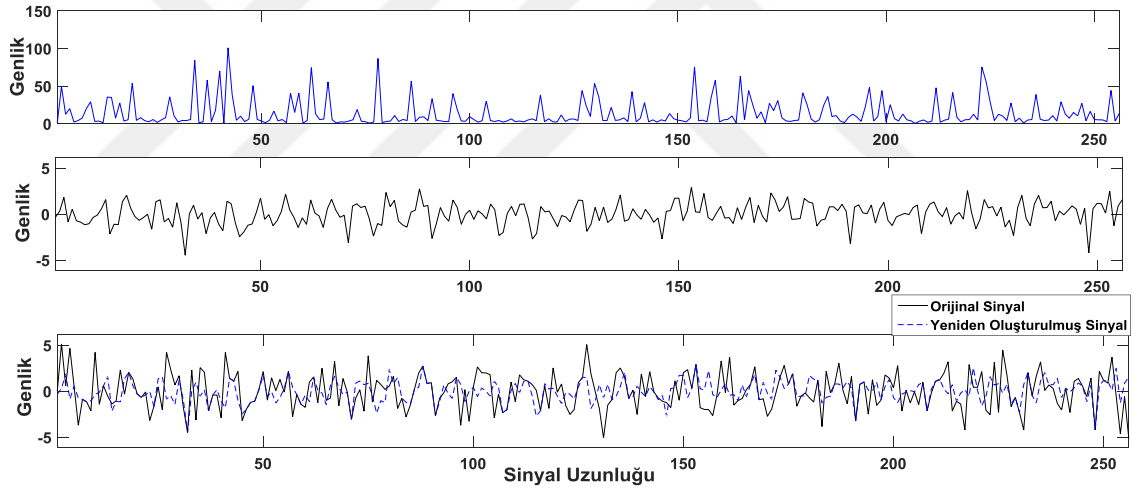
4.3. Test 3

Bu testte, ikinci testte incelenen problem için örnekleme sayısının yarıya düşürülmesi durumunda algoritmaların sinyali geri elde etmedeki performansları incelenmiştir. Yani $N=256, M=64$ ve $k=16$ alınarak değerlendirmeler yapılmıştır.

Üçüncü test için kullanılan orijinal sinyal, orijinal sinyalin Fourier katsayıları ve orijinal sinyalin ölçülmüş örnek değerleri Şekil 4.9’da gösterilmektedir. Şekil 4.10’da PDIP yöntemi, Şekil 4.11’de OMP algoritması Şekil 4.12’de CoSaMP algoritması kullanıldığı durumda elde edilen katsayılar ve ters Fourier dönüşümü alınarak yeniden oluşturulmuş orijinal sinyal gösterilmektedir.

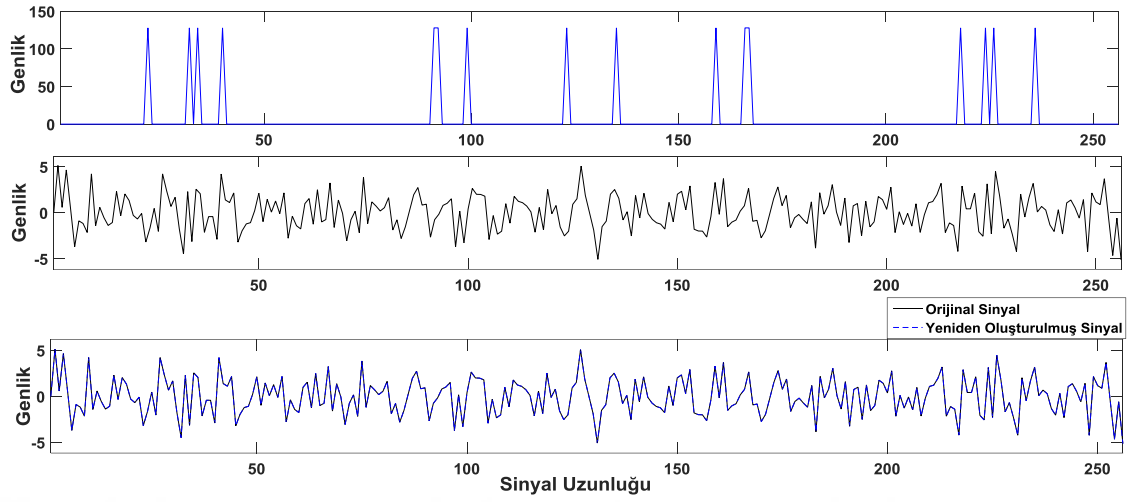


Şekil 4.9. $N=256$, $M=64$, $k=16$ için sırasıyla orijinal sinyal, sinyalin Fourier dönüşümü ve orijinal sinyalin örneklenmiş şekli.

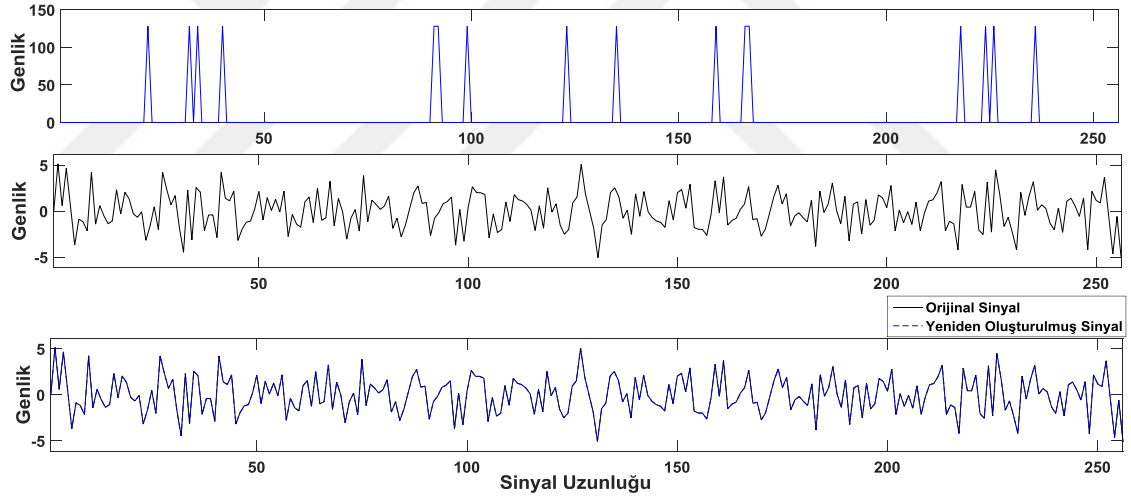


Şekil 4.10. $N=256$, $M=64$, $k=16$ için sırasıyla PDIP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.

Her üç algoritma ile çizdirilen Şekil 4.10, 4.11 ve 4.12 incelenirse, Şekil 4.10'da orijinal ve geri oluşturulan sinyalin grafiği tam olarak örtüşmemektedir. Şekil 4.11 ve Şekil 4.12'de ise OMP ve CoSaMP'ın geri elde ettiği sinyalin grafiği tam olarak örtüşmektedir. Bunlarda MSE hatası 10^{-27} seviyesinde olduğu için geri oluşturulan sinyalle orijinal sinyal arasında fark yoktur.



Şekil 4.11. $N=256$, $M=64$, $k=16$ için sırasıyla OMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.



Şekil 4.12. $N=256$, $M=64$, $k=16$ için sırasıyla CoSaMP algoritması ile elde edilen sinyal, sinyalin ters Fourier dönüşümü ve orijinal sinyal ile yeniden oluşturulmuş sinyal.

Tablo 4.3. Test 3 için algoritmaların performans sonuçları

Performans parametreleri	PDIP	OMP	CoSaMP
Test Koşulları	$N=256$, $M=64$, $k=16$		
MSE	4.0364	1.08×10^{-27}	1.27×10^{-27}
İterasyon Sayısı	17	16	2
Koşma süresi (s)	0.07695	0.014977	0.191402

Sonuçlar genel olarak değerlendirildiğinde, OMP ve CoSaMP algoritmasının seyrek sinyali geri elde etmede PDIP algoritmasından daha iyi olduğu görülmektedir. OMP ve CoSaMP'ın elde ettiği sinyalin MSE'si düşüktür. Ayrıca algoritmaların sinyali geri elde etme süreleri karşılaştırıldığında da açgözlü algoritmalar daha kısa sürede sinyali geri elde eder. Ancak OMP'nin sinyali geri elde etmeyi garanti etmemesi PDIP'ye göre bir dezavantajdır [57]. CoSaMP algoritması OMP'nin geliştirilmiş versiyonu olduğu için bu sorun CoSaMP algoritmasında yoktur.



5. BÖLÜM

POPÜLASYON TABANLI ALGORİTMALARIN

BENZETİM SONUÇLARI

5.1. Uygulanan Benzetim Çalışması Hakkında Bilgi

Tez çalışmasının bu bölümünde popülasyon tabanlı algoritmaların seyrek sinyalin geri oluşturulma aşamasındaki güçlü ve zayıf yönlerini görmek için klasik algoritmalarla aynı şartlar altında değerlendirilmesine yönelik bir inceleme gerçekleştirilmiştir. Bu bölümde gerçekleştirilen uygulamalarla, gürültüsüz ortamlardaki sezgisel ve klasik yöntemlerin performansı değerlendirilmiştir. Önerilen yöntem; GA, DE, PSO ve ABC algoritmaları ile seyrek sinyalin geri oluşturulmasında kullanılmış ve dört farklı örnek üzerinde gerçekleştirilen benzetimlerde elde edilen sonuçlar detaylı bir şekilde incelenerek değerlendirilmiştir. İncelenen test problemleri aynı zamanda önceki bölümlerde anlatılan üç farklı klasik algoritma ile hem kendi içlerinde hem de birbirleriyle karşılaştırılmışlardır. Böylelikle bu bölümde; GA, DE, PSO ve ABC algoritmaları ile PDIP, OMP ve CoSaMP algoritmaları ile aynı örnekler incelenmiş ve gerçekleştirilen işlemlerde elde edilen test sonuçları ile algoritmalar birbirleriyle kıyaslanmıştır.

Bu bölümde gerçekleştirilen incelemelerde dört temel aşama göz önüne alınmıştır. Birinci aşamada, önerilen yöntemin CS'de seyrek sinyalin geri oluşturulmasındaki başarımının, örnekler üzerinde ayrıntılı olarak incelenmesine dayanmaktadır. Böylelikle bu ilk kısımda algoritmaların farklı seyrek işaretleri geri oluşturmadaki performansları değerlendirilerek önerilen yöntemin farklı işaretlerde etkili olup olmadığı test edilmiştir. İkinci aşamada, bir sinyalin seyrekliğinin sabit tutulup ölçüm sayısının değiştirilmesi ve ölçüm sayısının sabit tutulup seyrekliğinin değiştirilmesi prensibine dayanmaktadır.

Böylelikle her iki farklı durum için algoritmaların seyrek işareti geri oluşturmadaki performansları değerlendirilmiştir. Üçüncü aşamada algoritmaların gürültü içeren bir ölçüme karşı gürbüzlüğü test edilmiştir. Son aşamada ise seyrek sinyal geri elde etmek için evrimsel algoritma kullanılan bir çalışma ile önerilen yöntem karşılaştırılmıştır. Test işlemlerine geçmeden önce kullanılan algoritmaların kontrol parametrelerinin, CS probleminde hangi değerlerde alınarak kullanılacağına dair ayrıntılı bir inceleme yapılmıştır.

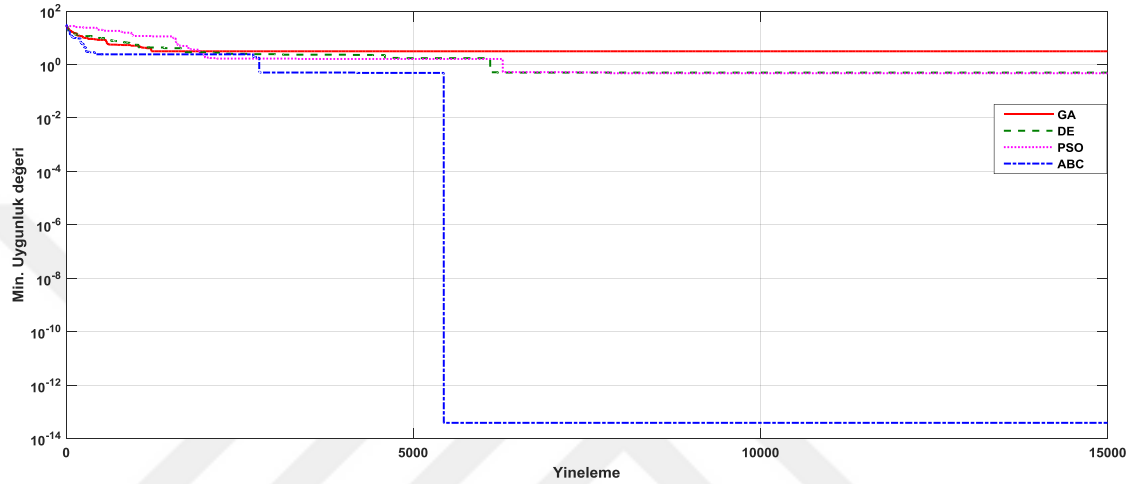
Test sonuçları için MATLAB yazılım paketi tercih edilmiştir. OMP ve CoSaMP algoritmaları MATLAB üzerinde uygulanmıştır. PDIP algoritması için “ l_1 magic” araç kutusundan yararlanılmıştır [45]. Önerilen metot için; GA, DE ve PSO algoritmaları algoritma için verilen temel adımlara göre kodlanarak kullanılmıştır. ABC algoritmasının kodu algoritma ile ilgili web sayfasında bulunmaktadır [58]. Önerilen metot ABC algoritmasına verilen koddan yararlanılarak uyarlanmıştır. Tablo 5.1’de bu çalışmada kullanılan sezgisel algoritmaların kontrol parametre değerleri verilmektedir.

Tablo 5.1. Kullanılan sezgisel algoritmaların kontrol parametreleri

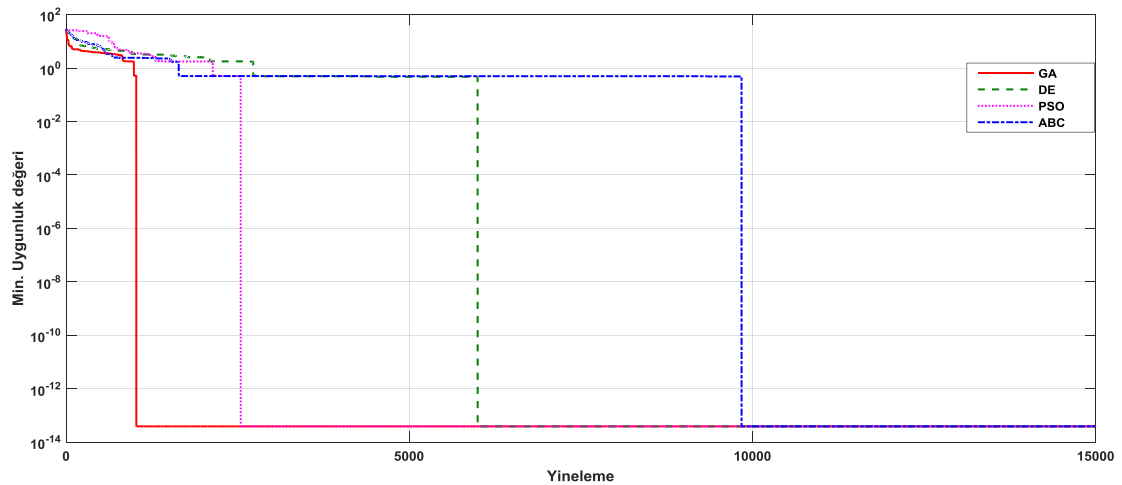
Parametre ismi	ABC	PSO	GA	DE
Popülasyon büyüklüğü	20	20	20	20
Maksimum yineleme	15000	15000	15000	15000
$C1$: Lokal öğrenme katsayısı	-	2.00	-	-
$C2$: Global öğrenme katsayısı	-	2.00	-	-
W : Atalet ağırlığı	-	1	-	-
$Wdamp$: Atalet ağırlığı sönümlleme oranı	-	0.999	-	-
Limit	200	-	-	-
Mutasyon oranı	-	-	0.01	-
Çaprazlama oranı	-	-	0.7	0.1
Ölçekleme faktörü (alt sınır)	-	-	-	0.2
Ölçekleme faktörü (üst sınır)	-	-	-	0.8

Tablo 5.1’den görüldüğü üzere bu algoritma parametrelerinin bir kısmı algoritmaya özeldir. Bazıları ise kullanılan dört algoritma için ortaktır. Ortak olmayan parametrelerin değerleri, CS’de sinyal geri oluşturma problemi için algoritmaların yaklaşık olarak en iyi sonuç vereceği şekilde incelenerek belirlenmiştir.

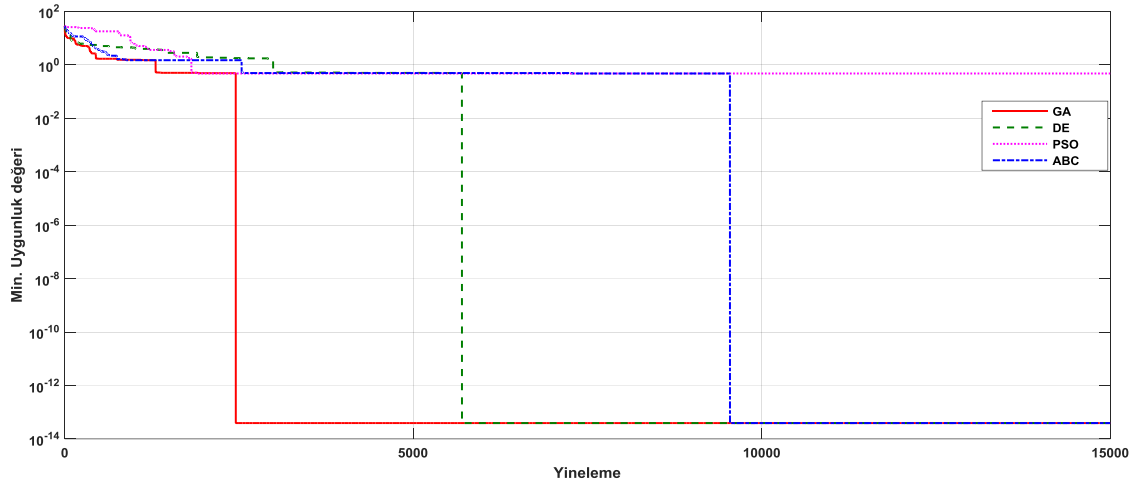
Ortak parametre değerlerinden olan popülasyon sayısının belirlenmesine yönelik olarak gerçekleştirilen çalışmalarda elde edilen şekiller, Şekil 6.1-6.5 arasındaki grafiklerde gösterilmektedir. Şekil 5.1.'de CS problemi için popülasyon sayısının 10, Şekil 5.2.'de 20, Şekil 5.3.'de 30, Şekil 5.4.'de 40 ve Şekil 5.5.'de 50 olması durumunda kullanılan algoritmaların yineleme bağı olarak uygunluk değerindeki değişim verilmektedir.



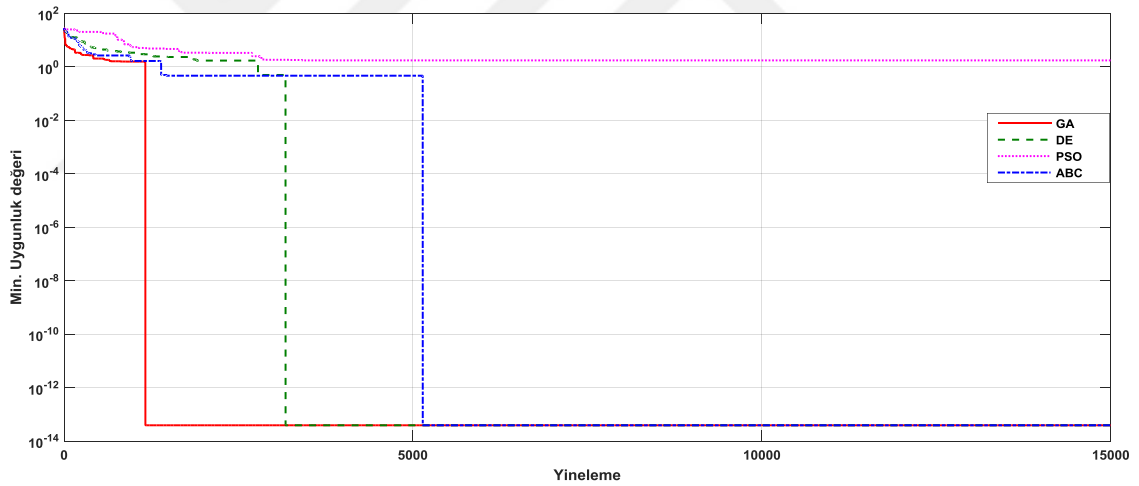
Şekil 5.1. Popülasyon sayısının 10 olması durumu için algoritmaların yinelemeye bağı olarak uygunluk değerinin değişimi.



Şekil 5.2. Popülasyon sayısının 20 olması durumu için algoritmaların yinelemeye bağı olarak uygunluk değerinin değişimi.

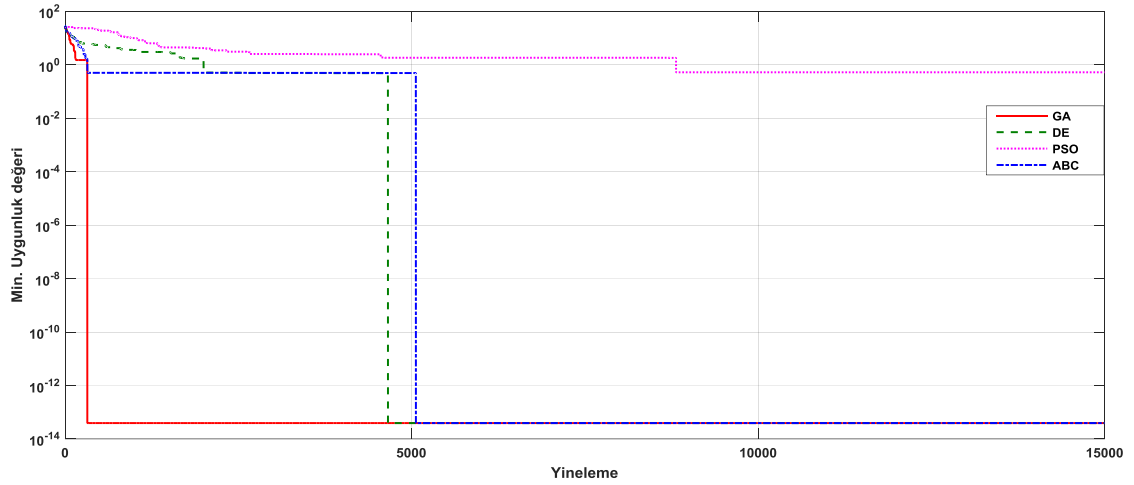


Şekil 5.3. Popülasyon sayısının 30 olması durumu için algoritmaların yinelemeye bağlı olarak uygunluk değerinin değişimi.



Şekil 5.4. Popülasyon sayısının 40 olması durumu için algoritmaların yinelemeye bağlı olarak uygunluk değerinin değişimi.

Kullanılan algoritmaların parametrelerinin belirlenmesinde kullanılan test şartları için sinyalin uzunluğu 256, ölçüm sayısı 64 ve seyreklik seviyesi de 16 olarak kullanılmıştır. Şekiller incelenecek olursa dört algoritmanın da ortak olarak iyi sonuç verdiği en küçük popülasyon sayısı 20 olarak belirlenmiştir. Bu tezde yapılan bütün testlerde popülasyon sayısı 20 olarak kullanılmıştır.



Şekil 5.5. Popülasyon sayısının 50 olması durumu için algoritmaların yinelemeye bağlı olarak uygunluk değerinin değişimi.

5.2. Test 1

Bu durum, önerilen metodun CS’de seyrek sinyalin geri oluşturulması aşamasındaki başarımının değişik örnekler üzerinde gösterilmesine dayanmaktadır. Bu örneklerin her birinde; N sinyalin uzunluğu, M ölçüm sayısı ve k seyreklik değerlerinden bir tanesi değiştirilerek yedi örnek üzerinde algoritmaların performansları denenmiştir. Her bir örnek popülasyon tabanlı algoritmalar ve klasik metotlarla 100’er defa çalıştırılmıştır. Algoritma her çalıştığında, sinyal ve ölçüm matrisinin değeri değişmektedir. Böylelikle algoritmanın gürbüzlüğü de test edilmesi de mümkün olmaktadır. Sonuç olarak her bir örnek için 100 kere denenmiş algoritmaya ait performans değerleri tablo olarak verilmiştir.

Toplam yedi örnekten dördünün detaylı istatistiksel değerleri Tablo 5.2 ve 5.5 arasındaki grafiklerde verilmiştir. Verilen tablolar, a ve b şeklinde ikiye ayrılmıştır. “a” şıklarında, 100 kere koşulmuş olan algoritmaların başarılı olan koşmalarına ait değerler verilmiştir. Başarı kriteri olarak yeniden oluşturulmuş sinyalle, orijinal sinyalin ortalama karesel hatası (Mean Squared Error-MSE)’nin 10^{-6} ’dan küçük olması temel alınmıştır. Başarım kriterleri olarak verilen bu tablolarda; MSE, sezgisel algoritmalar için amaç fonksiyonlarının değerleri, orijinal ve yeniden elde edilmiş sinyalin l_2 normu, MSE’nin standart sapması, algoritmaların geri elde etme süreleri, orijinal ve yeniden elde edilmiş sinyal arasındaki korelasyon değerleri kullanılmıştır. Örneğin, Örnek 1’de problem için PSO algoritması, 100 koşmanın 70’inde MSE’si 10^{-6} ’dan daha küçük sonuç üretmiştir.

Bu yüzden PSO için Tablo 5.2a'daki değerler 70 üzerindendir. “b” şıklarında verilen tablolar ise 100 koşmanın tamamını yani başarılı ve başarısız olan bütün koşmalara ait değerleri içermektedir. Başarım kriterleri olarak bu durumda; MSE, MSE'nin en küçük değeri, MSE'nin en büyük değeri, MSE'nin standart sapması, MSE'nin varyansı ve orijinal sinyalle geri oluşturulmuş sinyal arasındaki korelasyon değerleri verilmektedir. Örneğin PSO algoritması için Tablo 5.2b 100 koşmaya ait sonuçları içermektedir.

5.2.1. Örnek 1: $N=64$, $M=24$, $k=8$

Örnek 1'de her bir koşmada $N=64$ ve $k=8$ seyreklik seviyesine sahip bir x işareti, $M \times N$ boyutlu, random gauss ölçüm matrisi, üretilmektedir ve bunlar kullanılarak gözlem matrisi oluşturulmaktadır. Oluşturulan gözlem matrisi, ölçüm matrisi ve seyreklik seviyesi bilgisi kullanılarak algoritma yardımıyla x işareti yeniden oluşturulmaya çalışılmaktadır. Her bir koşmada N , M ve k değerleri aynı sinyale aittir fakat ölçüm matrisi farklıdır. Bu örnek için elde edilen performans değerleri Tablo 5.2a ve Tablo 5.2b'de verilmektedir.

Tablo 5.2a: $N=64$, $M=24$, $k=8$ olması durumunda başarılı koşmalar için elde edilen sonuçlar.

Alg.	B.O (%)	MSE(x, x') (ort.)	$\ y - y'\ _2$ (ort.)	$\ x - x'\ _2$ (ort.)	MSE(x, x') (Std. Sap.)	Koşma süresi (s) (ort.)	Korr. (x, x') (ort.)
GA	70	4.101×10^{-31}	1.887×10^{-14}	4.017×10^{-15}	8.676×10^{-31}	3.798	1.000
DE	98	2.523×10^{-31}	1.499×10^{-14}	3.334×10^{-15}	5.232×10^{-31}	3.4391	1.000
PSO	70	2.289×10^{-31}	1.481×10^{-14}	3.167×10^{-15}	2.540×10^{-28}	7.9100	1.000
ABC	100	1.954×10^{-31}	1.335×10^{-14}	2.936×10^{-15}	3.607×10^{-31}	1.8200	1.000
OMP	33	5.541×10^{-31}	-	4.564×10^{-15}	1.037×10^{-30}	0.0018	1.000
CoSaMP	60	1.088×10^{-30}	-	6.248×10^{-15}	2.737×10^{-30}	0.0088	1.000
L_1 min.	62	1.323×10^{-08}	-	5.828×10^{-05}	4.877×10^{-08}	0.0046	1.000

Algoritmaların başarılı olup olmadığı yani işaretin yeniden doğru olarak elde edilip edilmediği Tablo 5.2a'da ikinci sütunda verilen başarım oranından anlaşılabilir. Yine tabloda verilen MSE değerinin çok küçük olmasından da bu durum anlaşılabilir. Yani algoritmanın MSE değeri çok çok küçükse, girişteki sinyal sifıra çok yakın bir hatayla yani nerdeyse hatasız olarak elde edilmiştir. Tablo 5.2a'da ABC için önerilen metot 100 koşmanın hepsinde sinyali hatasız olarak geri elde etmiştir. Önerilen metot DE için 100

koşmanın 98'inde sinyali hatasız olarak geri elde etmiştir. Bu testteki şartlar altında genel olarak popülasyon tabanlı algoritmalar için olan önerilen yöntemin sinyali geri elde etmedeki başarısı, yüzde olarak OMP, CoSaMP ve L1 minimizasyon yöntemlerinden daha iyidir. Ancak algoritmaların sinyalin geri elde edilme süresi değerlendirildiğinde, klasik yöntemlerin popülasyon tabanlı yöntemlere göre daha iyi sonuç verdiği Tablo 5.2a'dan görülebilir.

Tablo 5.2b. $N=64$, $M=24$ ve $k=8$ olması durumunda başarılı ve başarısız tüm koşmalar için elde edilen sonuçlar.

Alg.	B.O (%)	MSE (x, x') (ort.)	MSE (x, x') (maks.)	MSE (x, x') (min.)	MSE (x, x') (Std. Sap.)	MSE (x, x') (Var.)	Korr. (x, x') (ort.)
GA	70	4.500×10^{-2}	7.018×10^{-1}	1.450×10^{-32}	1.007×10^{-1}	1.010×10^{-2}	0.8550
DE	98	4.484×10^{-4}	3.440×10^{-2}	1.296×10^{-32}	3.600×10^{-3}	1.283×10^{-5}	0.9951
PSO	70	1.660×10^{-2}	7.309×10^{-1}	1.088×10^{-32}	8.210×10^{-2}	6.800×10^{-3}	0.9678
ABC	100	1.954×10^{-31}	2.507×10^{-30}	1.231×10^{-32}	3.607×10^{-31}	1.301×10^{-61}	1.0000
OMP	33	4.840×10^{-2}	3.044×10^{-1}	2.571×10^{-32}	1.640×10^{-2}	6.930×10^{-2}	0.8045
CoSaMP	60	3.120×10^{-2}	6.909×10^{-1}	1.133×10^{-32}	3.070×10^{-2}	1.040×10^{-2}	0.9108
L1 min.	62	6.200×10^{-3}	9.700×10^{-2}	8.627×10^{-12}	1.522×10^{-2}	2.318×10^{-04}	0.9696

Yukarıdaki Tablo 5.2a ve 5.2b'de görüldüğü gibi, MSE belirlenen başarı kriteri olan 10^{-6} 'nın altına düştüğü zaman 10^{-30} 'lar seviyesine doğru yaklaşmaktadır. Yani sinyal tam olarak geri elde dileyebilmiştir. Bu yüzden Tablo 5.2a'daki MSE'lerin ortalaması PDIP algoritması hariç 10^{-30} seviyesindedir. Ayrıca Tablo 5.2a'da uygunluk değeri ve orijinal sinyal ile geri elde edilen sinyalin ikinci normu da 10^{-15} seviyesinde birbirine yakındır. Başarı oranı 100 olan algoritmanın MSE değeri Tablo 5.2a ve Tablo 5.2b'de aynıdır. Tablo 5.2b'de GA'nın başarı oranı 70 olmasına rağmen, başarı oranı 33 olan OMP algoritması ile MSE oranları birbirine yakındır. Bunun nedeni, GA'nın maksimum MSE değeri 0.7018 ve OMP algoritmasının 0.304 olmasıyla açıklanabilir. Ayrıca bu durum, GA'nın standart sapmasının OMP'ye göre daha yüksek olmasıyla ve GA'nın ortalama korelasyon değerinin daha düşük olmasıyla da açıklanabilir. Yani bu tablolara göre GA, başarılı olmayan koşmalarda orijinal sinyalden oldukça uzak değerler bulmaktadır.

5.2.2. Örnek 2: $N=64$, $M=24$, $k=12$

Bu örnekte seyreklik seviyesi 8’den 12’ye çıkarılmıştır. Bu şartlarda bir öncekine göre işareti geri elde etmek daha da zorlaşmıştır. Çünkü algoritmalar bu durumda 8 tane sıfır olmayan indis bulmak yerine 12 tane bulmak zorundadırlar. Tablo 5.3a ve Tablo 5.3b’de bu örnek için başarılı koşmalar ile başarılı ve başarısız tüm koşmalar için algoritmaların elde ettiği sonuçlar verilmiştir.

Tablo 5.3a. $N=64$, $M=24$, $k=12$ olması durumunda başarılı koşmalar için elde edilen sonuçlar.

Alg.	B.O (%)	MSE(x, x') (ort.)	$\ y - y'\ _2$ (ort.)	$\ x - x'\ _2$ (ort.)	MSE (x, x') (Std. Sap.)	Koşma süresi (s) (ort.)	Korr. (x, x') (ort.)
GA	7	2.654×10^{-31}	1.763×10^{-14}	3.913×10^{-15}	1.559×10^{-31}	8.3990	1
DE	73	6.249×10^{-31}	2.429×10^{-14}	5.529×10^{-15}	8.036×10^{-31}	23.2194	1
PSO	12	4.008×10^{-31}	2.075×10^{-14}	4.581×10^{-15}	4.338×10^{-31}	25.477	1
ABC	66	3.118×10^{-31}	1.855×10^{-14}	4.152×10^{-15}	3.215×10^{-31}	24.612	1
OMP	3	1.198×10^{-30}	-	7.361×10^{-15}	1.440×10^{-30}	0.0025	1
CoSaMP	0	-	-	-	-	-	-
PDIP	2	4.527×10^{-10}	-	2.348×10^{-04}	5.781×10^{-10}	0.0051	1

Tablo 5.3b. $N=64$, $M=24$ ve $k=12$ olması durumunda başarılı ve başarısız tüm koşmalar için elde edilen sonuçlar.

Alg.	B.O. (%)	MSE (x, x') (ort.)	MSE (x, x') (maks.)	MSE (x, x') (min.)	MSE (x, x') (Std. Sap.)	MSE (x, x') (Var)	Korr. (x, x') (ort.)
GA	7	0.2225	1.4665	4.219×10^{-32}	0.2137	4.57×10^{-2}	0.5643
DE	73	0.0373	0.8524	7.667×10^{-32}	0.1109	1.23×10^{-2}	0.9188
PSO	12	0.1240	0.8030	7.839×10^{-32}	0.1906	3.63×10^{-2}	0.7480
ABC	66	0.0105	0.2199	4.290×10^{-32}	0.0393	1.50×10^{-3}	0.9774
OMP	3	0.1422	0.6718	5.329×10^{-32}	0.1335	1.78×10^{-2}	0.6500
CoSaMP	0	0.0976	0.2819	1.740×10^{-2}	0.0505	2.60×10^{-3}	0.6789
PDIP	2	0.0478	0.2389	4.400×10^{-11}	0.0527	2.80×10^{-3}	0.8483

Tablo 5.3a’da görüldüğü gibi DE için önerilen metot en iyi başarı oranına sahiptir. Genel olarak değerlendirildiğinde DE ve ABC için önerilen metot ile diğerleri arasında başarı oranı açısından oldukça fark vardır. Bu örnekten, ölçüm sayısı ve seyreklik seviyesi birbirine yakın olduğunda önerilen metodun klasik yöntemlerden daha iyi sonuç verdiği görülmüştür. Tablo 5.3b’de GA’nın korelasyon değeri oldukça düşüktür.

Bunun nedeni de bir önceki örnekte açıklandığı gibidir. Bu örnekte diğer bir husus da DE algoritmasının başarı oranı ABC algoritmasından iyi olmasına rağmen MSE ortalaması üç kat daha büyüktür. Bu durumu Tablo 5.3b’de korelasyon, standart sapma ve MSE’nin maksimum değerleri de desteklemektedirler.

5.2.3. Örnek 3: $N=64$, $M=48$, $k=8$

Bu örnekteki testte ölçüm sayısı artırılmıştır. Bu nedenle işareti geri elde etmek buradaki bahsedilen algoritmalar için diğer örneklerden daha kolay olmaktadır. Çünkü işareti geri elde etmek için algoritmanın sahip olduğu bilgi de bu durumda daha çoktur. Örnek 3 için Tablo 5.4a’da başarılı koşmalar için Tablo 5.4b’de başarılı ve başarısız tüm koşmalar için algoritmalara bağlı olarak elde edilen performans değerleri verilmektedir.

Tablo 5.4a. $N=64$, $M=48$, $k=8$ olması durumunda başarılı koşmalar için elde edilen sonuçlar.

Alg.	B.O (%)	$MSE(x, x')$ (ort.)	$\ y - y'\ _2$ (ort.)	$\ x - x'\ _2$ (ort.)	$MSE(x, x')$ (Std. Sap.)	Koşma süresi (s) (ort.)	Korr. (x, x') (ort.)
GA	100	2.620×10^{-31}	2.253×10^{-14}	3.228×10^{-15}	6.701×10^{-31}	3.2875	1
DE	100	2.542×10^{-31}	2.156×10^{-14}	3.221×10^{-15}	4.886×10^{-31}	2.2360	1
PSO	87	2.407×10^{-31}	2.070×10^{-14}	2.955×10^{-15}	9.444×10^{-31}	7.0560	1
ABC	100	3.600×10^{-31}	2.240×10^{-14}	3.220×10^{-15}	1.650×10^{-30}	0.7660	1
OMP	97	2.510×10^{-31}		3.244×10^{-15}	5.103×10^{-31}	0.0018	1
CoSaMP	100	2.280×10^{-31}		3.418×10^{-15}	2.767×10^{-31}	0.0032	1
PDIP	100	8.299×10^{-12}		1.899×10^{-05}	1.128×10^{-11}	0.0128	1

Tablo 5.4b. $N=64$, $M=48$ ve $k=8$ olması durumunda başarılı ve başarısız tüm koşmalar için elde edilen sonuçlar.

Alg.	B.O (%)	$MSE(x, x')$ (ort.)	$MSE(x, x')$ (maks.)	$MSE(x, x')$ (min.)	$MSE(x, x')$ (Std. Sap.)	$MSE(x, x')$ (Var)	Korr. (x, x') (ort.)
GA	100	2.620×10^{-31}	6.211×10^{-30}	8.354×10^{-33}	6.701×10^{-31}	4.491×10^{-61}	1.0000
DE	100	2.542×10^{-31}	2.733×10^{-30}	2.530×10^{-33}	4.886×10^{-31}	2.388×10^{-61}	1.0000
PSO	87	1.140×10^{-04}	6.100×10^{-3}	4.911×10^{-33}	6.484×10^{-04}	4.205×10^{-07}	0.9995
ABC	100	3.601×10^{-31}	1.630×10^{-29}	4.061×10^{-33}	1.651×10^{-30}	2.722×10^{-60}	1.0000
OMP	97	4.959×10^{-05}	3.900×10^{-3}	8.450×10^{-33}	3.966×10^{-04}	1.573×10^{-07}	0.9998
CoSaMP	100	2.281×10^{-31}	1.772×10^{-30}	5.832×10^{-33}	2.768×10^{-31}	7.662×10^{-62}	1.0000
PDIP	100	8.299×10^{-12}	7.934×10^{-11}	1.228×10^{-13}	1.128×10^{-11}	1.273×10^{-22}	1.0000

Tablolardan anlaşılaçağı üzere, PSO ve OMP haricindeki algoritmalar % 100 başarı göstermiştir. Genel olarak bu koşullarda yani ölçüm sayısının yüksek, seyreklik seviyesinin düşük olduğu durumlarda klasik yöntemleri kullanmak daha doğru olacaktır. Çünkü klasik algoritmalar sinyalin geri elde edilme süresi açısından sezgisel algoritmalarla göre daha kısa sürede sonuca ulaştıklarından daha avantajlıdır.

5.2.4. Örnek 4: $N=64$, $M=48$, $k=12$

Bu örnekte, Örnek 3’de verilen seyreklik seviyesi daha da artırılmış böylelikle sinyalin geri elde edilmesi biraz daha zorlaştırılmıştır. Tablo 5.5a’da bu örnekteki başarılı koşmalar için Tablo 5.5b’de başarılı ve başarısız tüm koşmalar için algoritmaların sonuçlarına ait performans değerleri verilmektedir. Bir önceki örnekte olduğu gibi bu örnekte de OMP ve PSO haricindeki algoritmalar sinyali yeniden elde etmede %100 başarıya ulaşmıştır.

Tablo 5.5a. $N=64$, $M=48$, $k=12$ olması durumunda başarılı koşmalar için elde edilen sonuçlar.

Alg.	B.O (%)	$MSE(x, x')$ (ortalama)	$\ y - y'\ _2$ (ortalama)	$\ x - x'\ _2$ (ortalama)	$MSE(x, x')$ (Std. Sap.)	Koşma süresi (s) (ortalama)	Korr. (x, x')
GA	100	5.367×10^{-31}	3.402×10^{-14}	5.034×10^{-15}	8.520×10^{-31}	4.3033	1
DE	100	5.325×10^{-31}	3.197×10^{-14}	4.703×10^{-15}	1.145×10^{-30}	6.1102	1
PSO	77	4.050×10^{-31}	3.126×10^{-14}	4.324×10^{-15}	7.105×10^{-31}	9.9385	1
ABC	100	6.241×10^{-31}	3.299×10^{-14}	4.804×10^{-15}	1.601×10^{-30}	1.6786	1
OMP	83	5.416×10^{-09}	-	4.796×10^{-15}	4.934×10^{-08}	0.0029	1
CoSaMP	100	3.874×10^{-31}	-	4.582×10^{-15}	4.072×10^{-31}	0.0046	1
PDIP	100	1.415×10^{-11}	-	2.514×10^{-05}	1.898×10^{-11}	0.0178	1

Tablo 5.5b: $N=64$, $M=48$ ve $k=12$ olması durumunda başarılı ve başarısız tüm koşmalar için elde edilen sonuçlar.

Alg.	B.O (%)	$MSE(x, x')$ (ort.)	$MSE(x, x')$ (maks.)	$MSE(x, x')$ (min.)	$MSE(x, x')$ (Std. Sap.)	$MSE(x, x')$ (Var)	Korr. (x, x') (ort.)
GA	100	5.367×10^{-31}	5.687×10^{-30}	3.838×10^{-32}	8.520×10^{-31}	7.259×10^{-61}	1.0000
DE	100	5.325×10^{-31}	8.318×10^{-30}	2.806×10^{-32}	1.145×10^{-30}	1.311×10^{-60}	1.0000
PSO	77	9.627×10^{-05}	0.0025	3.839×10^{-32}	3.329×10^{-04}	1.108×10^{-07}	0.9997
ABC	100	6.241×10^{-31}	1.090×10^{-29}	2.227×10^{-32}	1.601×10^{-30}	2.563×10^{-60}	1.0000
OMP	83	1.605×10^{-04}	0.0060	4.278×10^{-32}	7.349×10^{-04}	5.402×10^{-07}	0.9996
CoSaMP	100	3.874×10^{-31}	3.394×10^{-30}	3.297×10^{-32}	4.072×10^{-31}	1.658×10^{-61}	1.0000
PDIP	100	1.415×10^{-11}	1.006×10^{-10}	2.597×10^{-13}	1.898×10^{-11}	3.604×10^{-22}	1.0000

Ayrıca ayrıntılı incelenen bu dört örnekten farklı olarak N , M ve k 'nin farklı durumları için de yüzde başarı oranları incelenmiştir. Kullanılan algoritmalar yine aynı koşullarda 100 kere koşulmuş ve elde edilen sonuçlar Tablo 5.6'da bir bütün olarak gösterilmiştir.

Tablo 5.6. Algoritmaların farklı durumlar için sinyali geri elde etmelerine ait yüzde başarı oranları.

N	M	k	GA	DE	PSO	ABC	OMP	CoSaMP	PDIP
64	24	8	70	98	81	100	33	60	62
		12	7	73	12	66	3	0	2
	48	8	100	100	87	100	97	100	100
		12	100	100	77	100	83	100	100
128	64	12	100	100	70	98	95	100	100
		16	98	100	55	88	77	100	100
256	64	16	80	85	34	54	57	85	65

Tablo 5.6'da sezgisel ve klasik algoritmaların sonuçlarının karşılaştırılması görülebilir. Bu elde edilen verilere göre sezgisel algoritmalar kendi aralarında karşılaştırılacak olursa; sinyalin uzunluğu N küçük olduğunda ve ölçüm sayısı da seyreklik seviyesine yakın olduğunda GA diğer algoritmalarla göre daha kötü sonuç vermiştir. Bu durum, sinyali elde etmek için deterministik bilginin az olduğu şeklinde yorumlanabilir. Bu durumda GA diğerlerine göre daha zayıf kalmaktadır. İşaretin uzunluğu arttığında sezgisel algoritmalar için önerilen metotta araştırma uzayı da artmakta ve bu da algoritmanın çözüme ulaşmasını zorlaştırabilmektedir. Bu koşullarda GA, ABC ve PSO algoritmalarına göre daha iyi performans sergilemiştir. Araştırma uzayının artması işareti geri oluşturmayı zorlaştırmasına rağmen GA'nın başarı oranı işaret uzunluğu artınca daha yüksek olmuştur. DE algoritması buradaki değerlendirmeye göre sinyali geri oluşturmada daha iyi performansa sahiptir. Tablo 5.6'da gösterilen test koşullarının hepsinde başarı oranı % 70'in üzerindedir. Bu değerlendirmede PSO algoritmasının başarı oranı diğer sezgisel algoritmalarından daha düşüktür. Ancak genel olarak bakıldığında, klasik algoritmalar içinde yer alan OMP gibi başarısı kanıtlanmış bir algoritmanın başarı oranına da yakındır. Sonuç olarak seyrek sinyali geri elde etmede kullanılabilir. ABC algoritması da başarı oranı açısından DE algoritmasının biraz gerisinde olmak kaydıyla genel olarak başarı oranı yüksektir. İşaretin uzunluğu artınca

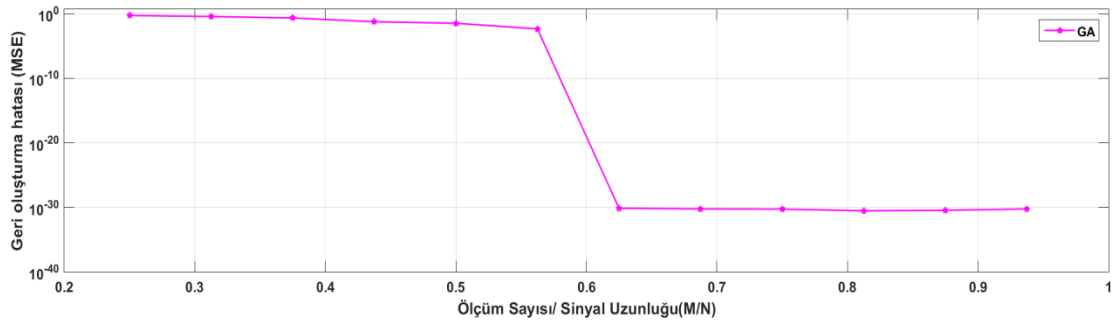
başarı oranı düşmüştür. Yani araştırma uzayı artınca çözümü bulma olasılığı DE ve GA algoritmalarına göre daha zayıftır. ABC algoritması Ölçüm sayısı ve seyreklik derecesi birbirine yakın olduğunda GA, PSO ve buradaki klasik yöntemlerden daha iyi bir başarı oranına sahiptir.

Farklı N , M ve k değerleri için algoritmalar test edilirken her bir test koşulu 100 defa çalıştırılmıştır. Elde edilen bütün sonuçlar değerlendirildiğinde OMP, CoSaMP ve PDIP klasik optimizasyon algoritmaları zaten oldukça başarılı algoritmalar ve hali hazırda CS uygulamalarında seyrek sinyalin geri oluşumunda sıklıkla kullanılmaktadırlar. Her çalışmada işaretin ve ölçüm matrisinin değeri değişmektedir. Fakat aynı şartlar altında popülasyon tabanlı algoritmalarla gerçekleştirilen örneklerden anlaşılabileceği gibi önerilen yöntem, CS’de seyrek sinyali geri oluşturmak için tercih edilebilir yöntem olarak değerlendirilebilir.

5.3. Test 2

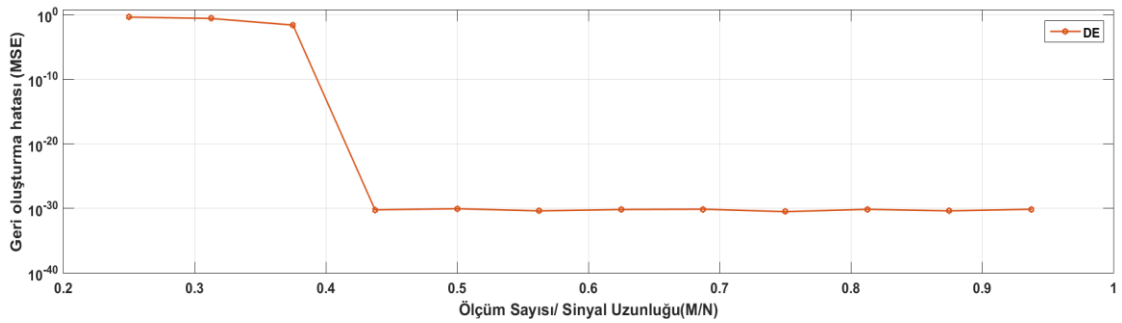
5.3.1 Popülasyon tabanlı algoritmalarla sinyalin ölçüm sayısına karşılık hatasının değişiminin incelenmesi

Benzetim sonuçlarının bu kısmında bir önceki kısım ile benzer olarak seyreklik seviyesi değişimine ve ölçüm sayısı değişimine karşılık incelemeler gerçekleştirilmiştir. Birinci kısımda verilen incelemelerden farklı olarak sezgisel algoritmalar aynı problem için 30 defa çalıştırılmıştır. Önceki kısımda her çalışmada problem de değişmekteydi ve algoritmalar 100 kere çalıştırılmıştı. Gerçekleştirilen benzetim çalışmaları MSE değeri temel alınarak değerlendirilmiştir. Seyrek sinyalin geri elde edilmesine ait sonuçlar, grafikler halinde verilmiştir. Bu grafikler iki gruba ayrılır: Birinci gruptaki grafiklerde algoritmaların her birinin geri elde etme hatasının ölçüm sayısına göre değişimi gösterilmektedir. Test sinyalinin uzunluğu $N=64$ ve seyreklik derecesi $k=12$ sabit olduğu durumda ölçüm sayısı M , 16’den başlayarak 60’a kadar 4’er 4’er artmaktadır. Şekil 5.6 ile Şekil 5.12 arasında konuyla ilgili elde edilen grafikler verilmektedir. Şekil 5.6’da GA için M/N ’ye bağlı olarak MSE hatasının yani yeniden oluşturulan sinyalin hatasının nasıl değiştiği gösterilmektedir.



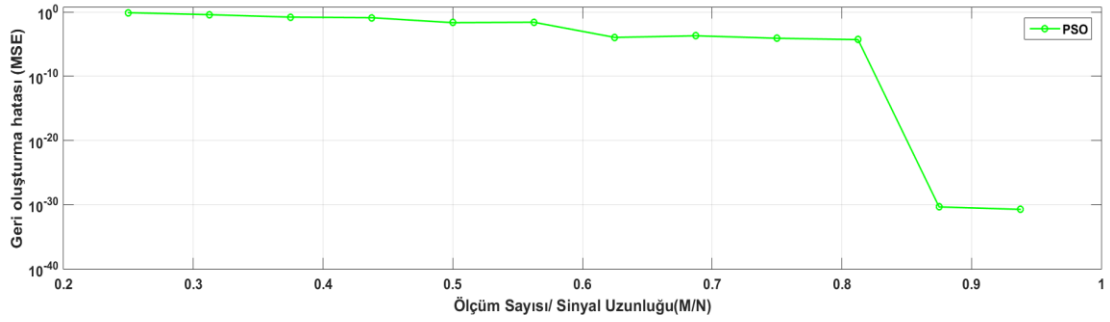
Şekil 5.6. GA için geri oluşturulan sinyalin ölçüm sayısına karşılık MSE hatasının değişimi.

Şekil 5.6'da görüldüğü gibi GA'ya ait çizdirilen grafikte hatanın tam olarak sıfır olduğu yer M 'nin 40 olduğu noktadır. Böylece $M/N=40/64=0.625$ 'dir. GA grafiğinde $M=40$ noktasından önceki ölçüm sayılarında da sıfır hataya ulaşılmıştır ancak bu grafikteki MSE değerleri 30 koşmanın ortalamasıdır. Bu sebeple grafikte MSE değerini sıfır olarak görebilmek için 30 koşmanın hepsinin sıfır olması gerekmektedir. Bu nokta da GA için ölçüm sayısının 40 olduğu durumdur.



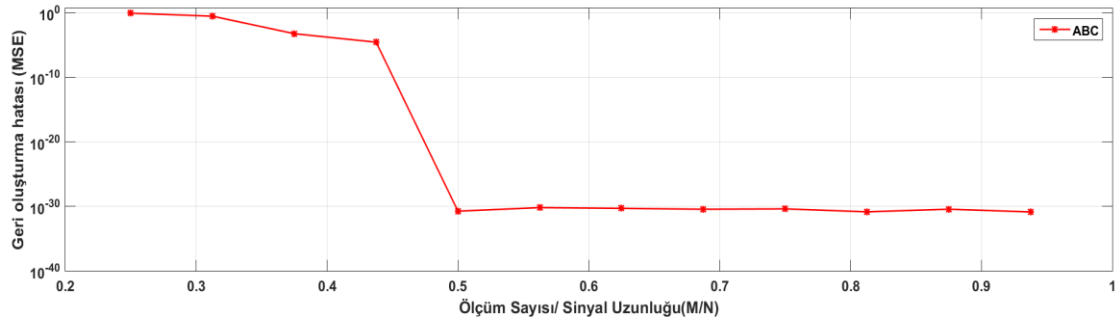
Şekil 5.7. DE algoritması için geri oluşturulan sinyalin ölçüm sayısına karşılık MSE hatasının değişimi.

Şekil 5.7. DE için geri oluşturulan sinyalin ölçüm sayısına karşılık MSE hatasının değişimi verilmektedir. Şekilde görüldüğü gibi DE algoritması için çizdirilen grafik değerlendirildiğinde hatanın tam olarak sıfır olduğu yer $M=28$ olarak elde edilmiştir. Yani, $M/N=28/64=0.4375$ 'dir. DE algoritmasının hatasının değişimi ile ilgili olarak $M=28$ 'den önceki ölçüm sayılarında da bazı koşmalarda sıfır hata ile sonucu ulaşıldığı belirtilebilir. Fakat bu grafikteki MSE değerleri 30 koşmanın ortalaması olduğundan, grafikte sıfır MSE görebilmek için 30 koşmanın hepsinin sıfır olması gerekir. Bu nokta da DE için $M=28$ noktasıdır. Bu noktadan sonraki ölçüm sayılarında da normal olarak hata sıfırdır.



Şekil 5.8. PSO algoritması için geri oluşturulan sinyalin ölçüm sayısına karşılık MSE hatasının değişimi.

Şekil 5.8’de PSO algoritması için geri oluşturulan sinyalin ölçüm sayısına karşılık MSE hatasının değişimi gösterilmektedir. Şekilden görüldüğü gibi PSO algoritması için çizdirilen grafikte hata $M=40$ ’dan sonra düşük olsa da tam olarak 30 koşmanın da sıfır olduğu yer 56 noktasıdır, böylece $M/N=56/64=0.875$ ’dir. Örneğin PSO $M=32$ noktasında $M/N=0.5$ ’dir. Bu durumda 30 koşmanın 15 tanesinin MSE değeri sıfırdır.

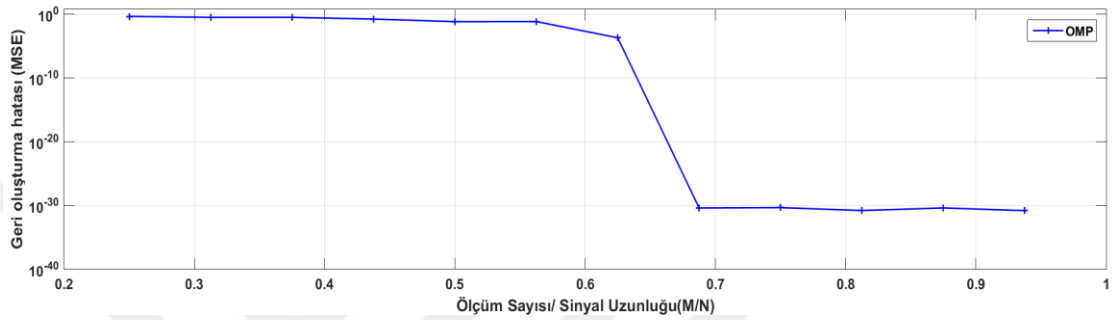


Şekil 5.9. ABC algoritması için geri oluşturulan sinyalin ölçüm sayısına karşılık MSE hatasının değişimi.

Şekil 5.9. ABC algoritması için geri oluşturulan sinyalin ölçüm sayısına karşı MSE hatasının değişimi verilmektedir. Şekilden anlaşılacağı üzere ABC algoritması için hata tam olarak $M=32$ ’de sıfır olmaktadır Bu nokta grafikte $M/N=32/64=0.5$ noktasına karşılık gelmektedir. ABC algoritması için de diğer kullanılan sezgisel algoritmalar için aynı durum söz konusudur. Yani algoritma ile 32’den önceki ölçüm sayılarında da bazı koşmalarda sıfır hataya ulaşılmıştır ancak bu grafikteki MSE değerleri 30 koşmanın ortalaması olduğundan grafikte sıfır MSE görülemez. Sıfır görebilmek için 30 koşmanın hepsinin sıfır olması gerekmektedir. ABC algoritması için 32 noktasından sonraki ölçüm sayılarında da normal olarak hata sıfırdır.

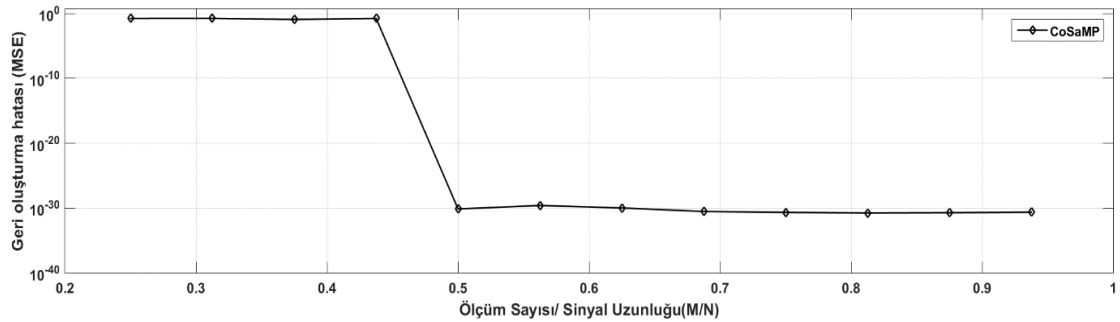
5.3.2 Klasik algoritmalarla sinyalin ölçüm sayısına karşılık hatasının değişiminin incelenmesi

Bu kısımda klasik algoritmalarla sinyaldeki ölçüm sayısına karşılık MSE hatasının değişimi incelenmiştir. Klasik yöntemler kullanıldığı durumda da çözülmesi hedeflenen test problemi sezgisel algoritmalarla da kullanılmıştır. Dolayısıyla aynı test problemi kullanıldığından algoritmalar bir defa koşturulmuştur.



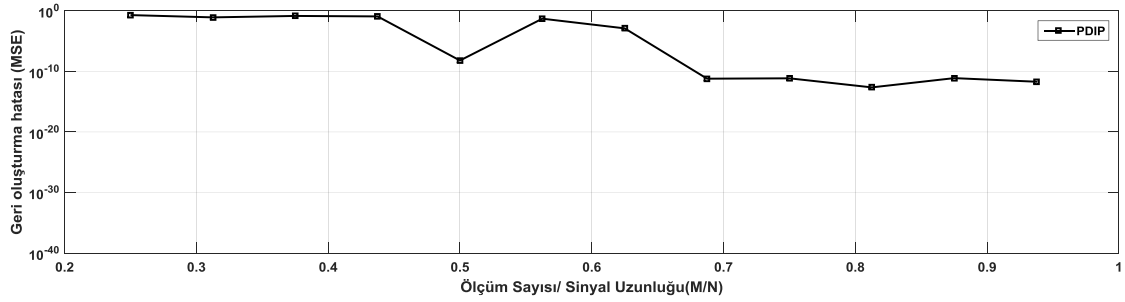
Şekil 5.10. OMP algoritması ile geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatası.

Şekil 5.10'da OMP algoritması kullanıldığı durumda geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatasındaki değişim gösterilmektedir. Şekil incelenecek olursa OMP algoritması için $M=44$ 'de hata sıfır olmaktadır. Bu nokta şekil üzerinde $M/N=44/64=0.6875$ noktasına karşılık gelmektedir. Sezgisel algoritmalarla farklı olarak test edilen durumların içinde, $M=44$ 'den önce sıfır olduğu nokta yoktur.



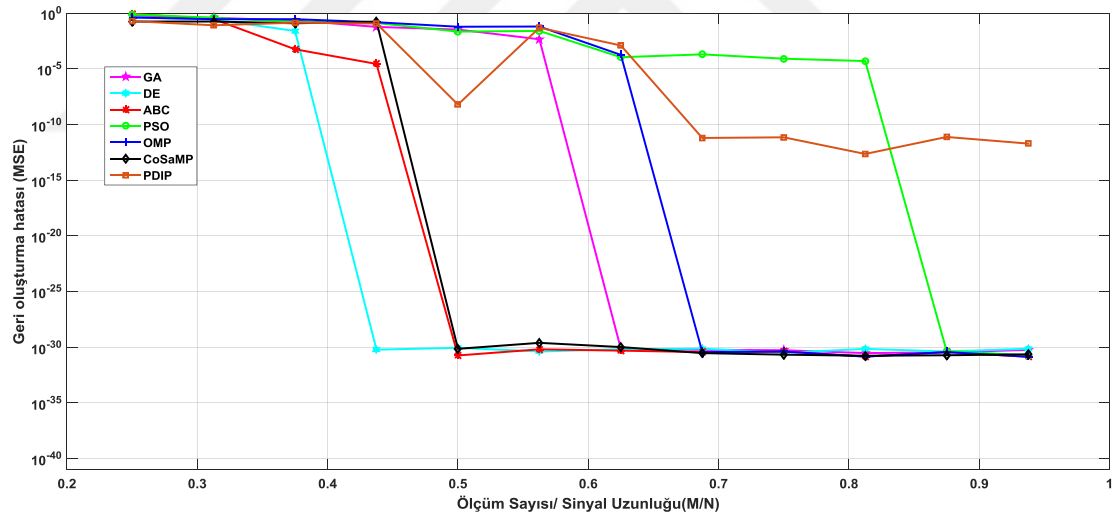
Şekil 5.11. CoSaMP algoritması ile geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatası.

Şekil 5.11'de geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatası CoSaMP algoritması ile incelenmiştir. Şekil değerlendirildiğinde hatanın tam olarak sıfır olduğu yer ABC algoritmasında bulunan değerle aynıdır; $M=32$ noktasıdır. Bu değer grafik üzerinde $M/N=32/64=0.5$ noktasına karşılık gelmektedir.



Şekil 5.12. PDIP algoritması ile geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatası.

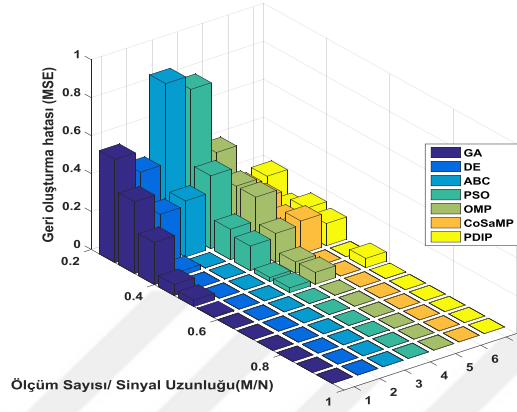
PDIP algoritması ile geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatası Şekil 5.12’de verilmektedir. Çizdirilen grafik incelendiğinde bu algoritmayla hatanın en düşük olduğu yer 44 olarak bulunmuştur. Yani grafikte bu, $M/N=44/64=0.6875$ noktasına karşılık gelmektedir. Bu noktadan sonra hata, 10^{-12} seviyelerinde olmaktadır.



Şekil 5.13. Klasik ve sezgisel algoritmalar için geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatalarındaki değişim.

Klasik ve sezgisel algoritmalar için geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatalarındaki değişim Şekil 5.13’de aynı grafik üzerinde gösterilmektedir. Şekil 5.13 ve Şekil 5.6 ile Şekil 5.12 arasında verilen tüm şekiller incelendiğinde aynı şartlar altında bu test için DE algoritması ile yeniden oluşturma hatası diğer algoritmalarla göre en önce sıfıra ulaştığından bu testte en iyi sonuç veren algoritma DE algoritması olmaktadır. DE algoritmasından sonra ABC algoritması ve CoSaMP

yaklaşık olarak aynı performansı göstermiştir. Bunlardan sonra sırasıyla GA, OMP, PDIP optimizasyon yöntemi ve PSO algoritması gelmektedir. Şekil 5.14’de geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatasının üç boyutlu çubuk grafikte gösterimi verilmektedir.

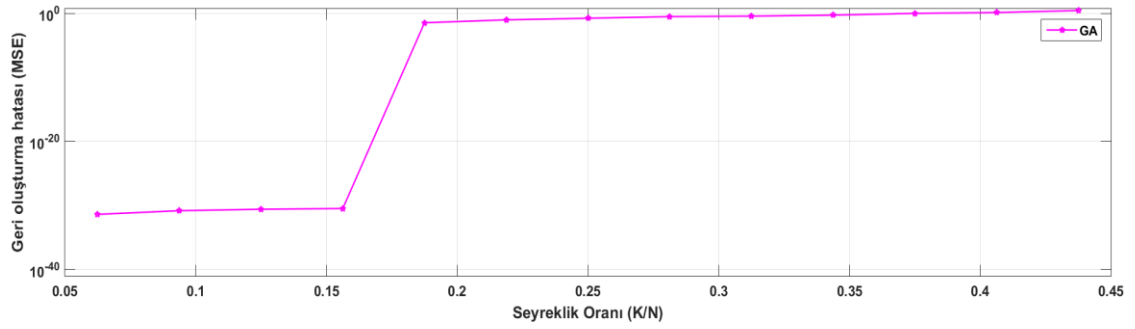


Şekil 5.14. Geri oluşturulan sinyalin ölçüm sayısına karşılık geri oluşturma hatasının üç boyutlu çubuk grafiğinde gösterilmesi.

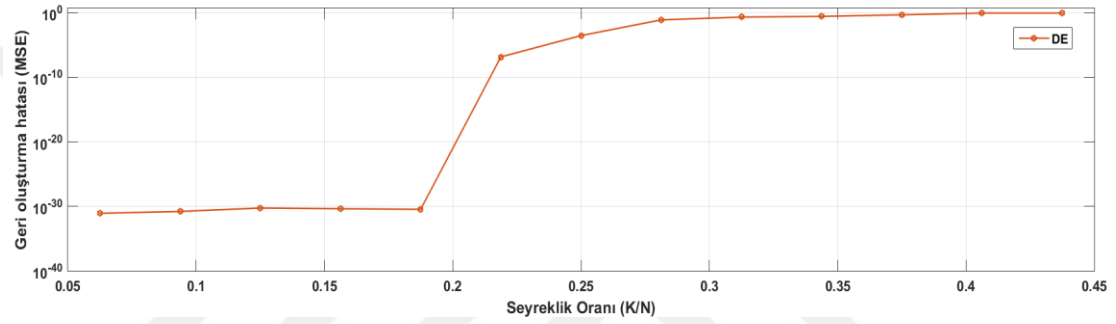
5.3.3 Popülasyon tabanlı algoritmalarla seyreklik seviyesine karşılık geri oluşturma hatasının incelenmesi

Bu kısımda test sinyalinin 64 uzunluğunda ve ölçüm sayısı da 32 olarak alınmış ve bu durumda popülasyon tabanlı algoritmalarla geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası incelenmiştir. Dört farklı optimizasyon algoritması ile gerçekleştirilen benzetimlerde sinyalin yeniden elde edilmesi aşamasındaki hataların değişimi Şekil 5.15 ile 5.21 arasında verilmiştir. Seyreklik seviyesi 4’ten başlayarak 28’e kadar 2’şer 2’şer artmaktadır. Şekil 5.15’de GA algoritması ile geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası çizdirilmiştir. Şekilden anlaşılabacağı üzere GA $k=10$ ’a kadar yani $k/N=10/64=0.15625$ noktasına kadar çok çok küçük sıfıra yakın bir hata ile sinyali geri oluşturmaktadır.

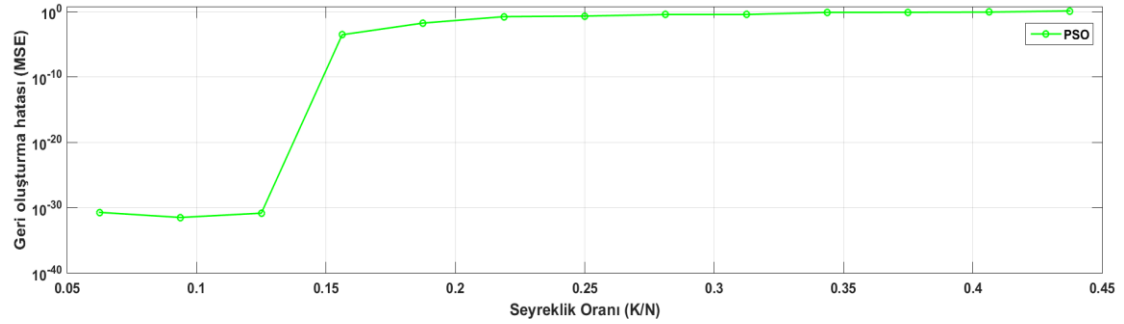
Şekil 5.16. DE algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası verilmektedir. Şekilden görüldüğü gibi, DE algoritması $k=12$ ’ye kadar sıfıra yakın hata ile sinyali geri oluşturmaktadır. Bu değer grafikte $k/N=12/64=0.1875$ noktasına karşılık gelmektedir.



Şekil 5.15. GA algoritması ile geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası.

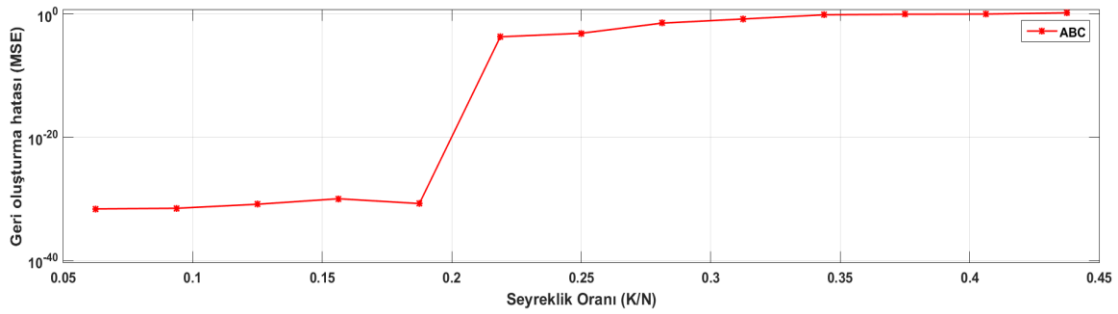


Şekil 5.16. DE algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası.



Şekil 5.17. PSO algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası.

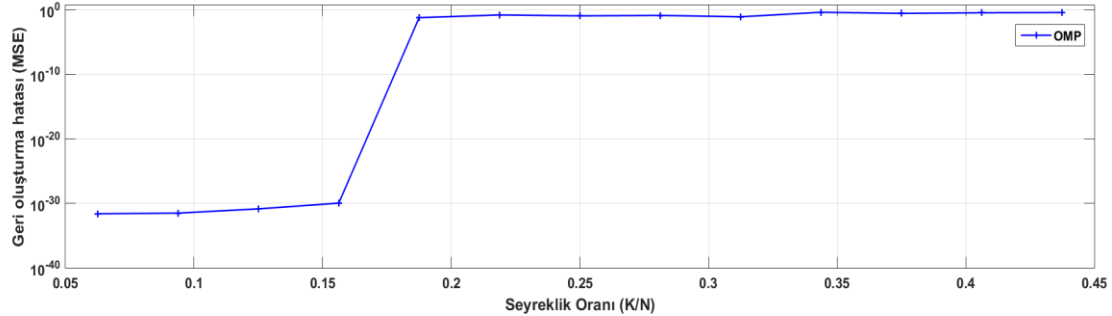
Geri oluşturulan sinyalin seyreklik seviyesine karşılık geri oluşturma hatası PSO algoritması kullanıldığı durumda Şekil 5.17'de verildiği gibi elde edilmiştir. Şekil 5.17'de incelenecek olursa, algoritmanın 8'e kadar yani $k/N=8/64=0.125$ noktasına kadar yaklaşık 10^{-30} seviyelerinde sıfıra çok yakın bir hata ile sinyali geri oluşturduğu anlaşılmaktadır.



Şekil 5.18. ABC algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası.

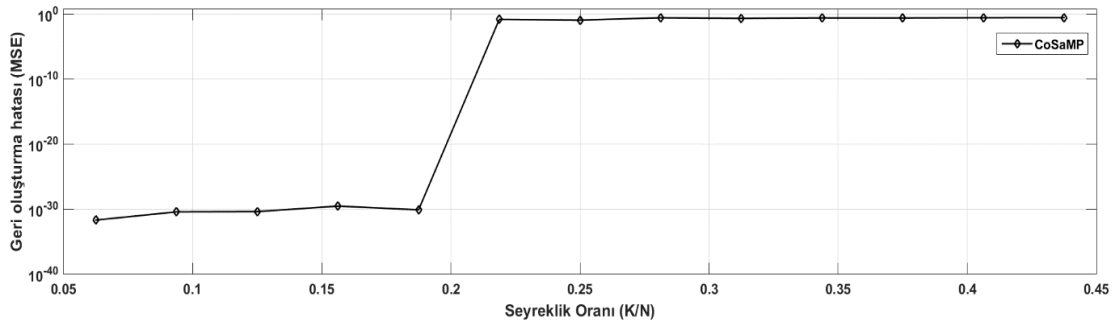
ABC algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası Şekil 5.18 ile verilmektedir. Bu şekil değerlendirildiğinde, ABC algoritması kullanılması durumunda da 12'ye kadar yani $k/N=12/64=0.1875$ noktasına kadar yine sıfıra çok yakın bir hata ile sinyalin geri oluşturulduğu belirlenebilmektedir.

5.3.4 Klasik algoritmalarla seyreklik seviyesine karşılık geri oluşturma hatasının incelenmesi



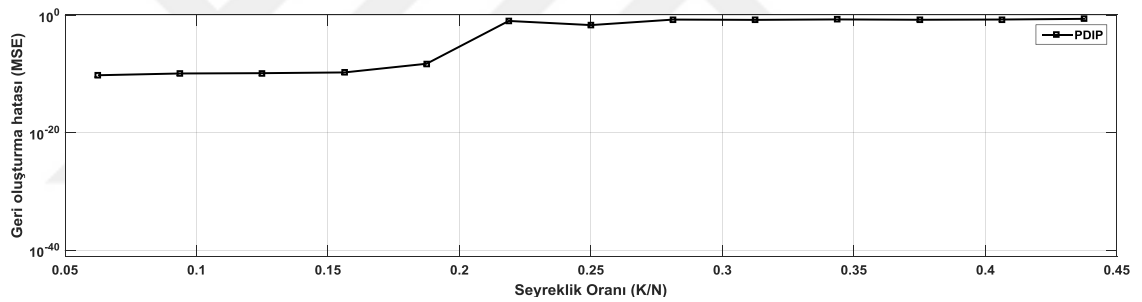
Şekil 5.19. OMP algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası.

Şekil 5.19'da OMP algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası gösterilmektedir. Şekil incelendiğinde anlaşılabacağı gibi OMP algoritması kullanıldığı durumda da $k=10$ 'a kadar yani $k/N=10/64=0.115625$ noktasına kadar 10^{-30} 'dan daha küçük yani neredeyse sıfır olan bir değerle, sinyalin yeniden oluşturulduğu anlaşılmaktadır.



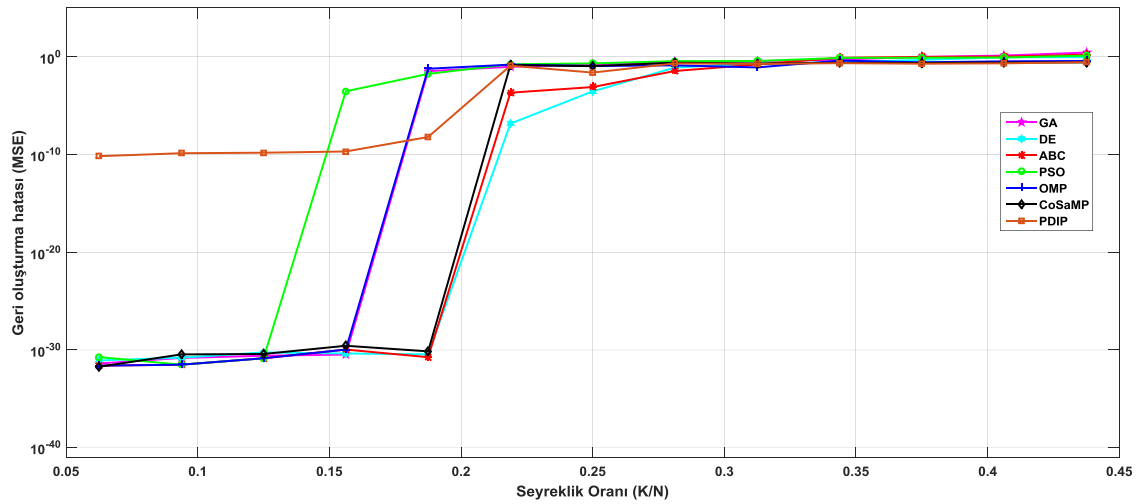
Şekil 5.20. CoSaMP algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası.

CoSaMP algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası, Şekil 5.20 ile verilmektedir. Şekil incelenirse CoSaMP algoritması da $k/N=12/64=0.1875$ noktasına kadar yani $k=12$ 'ye kadar 10^{-30} 'dan daha küçük olan bir hata değeriyle sinyali geri oluşturmaktadır.

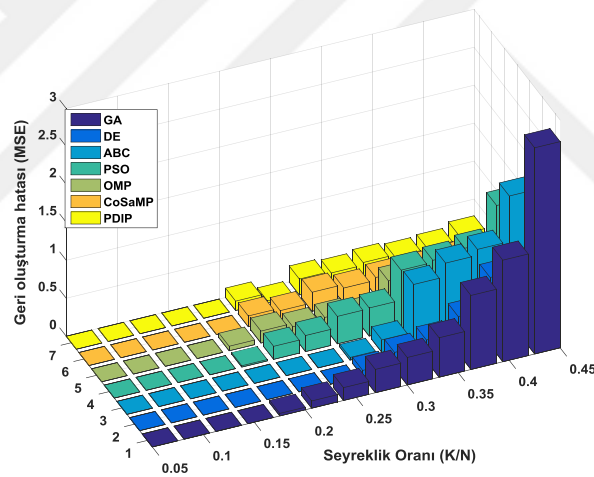


Şekil 5.21. PDIP algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası.

PDIP algoritması kullanılarak geri oluşturulan seyreklik seviyesine karşılık geri oluşturma hatası Şekil 5.21'de gösterilmektedir. Bu şekil değerlendirildiğinde PDIP algoritması ile seyreklik seviyesinin 12 olmasına kadar yani diğer bir ifadeyle $k/N=12/64=0.1875$ noktasına kadar minimum hata ile sinyal geri oluşturulmaktadır. 0.1875 noktasından sonra hata daha da artmaktadır. Şekil 5.22'de klasik ve sezgisel algoritmalar için seyreklik oranı ve MSE'ye bağlı olarak algoritma sonuçlarının değişimi, Şekil 5.23'de de seyreklik oranı ve MSE'ye bağlı olarak algoritmaların sonuçlarının 3 boyutlu çubuk grafikte gösterimi verilmektedir.



Şekil 5.22. Klasik ve sezgisel algoritmalar için seyreklik oranı ve MSE'ye bağlı olarak algoritma sonuçlarının değişimi.



Şekil 5.23. Seyreklik oranı ve MSE'ye bağlı olarak algoritmaların sonuçlarının 3 boyutlu çubuk grafiğinde gösterilmesi.

Bu kısımda test edilen problem için klasik ve sezgisel algoritmalar birlikte değerlendirildiklerinde elde edilen Şekil 5.15 ve 5.23 arasında verilen tüm şekiller karşılaştırıldığında DE, ABC ve CoSaMP algoritmalarının seyrek işareti geri elde etme işleminde birbirine yakın performans gösterdiği anlaşılmıştır. Bu algoritmalarla; $N=64$, $M=32$ ve $k=12$ durumuna kadar bütün koşmalarda sıfıra yakın hata ile işaret geri elde edilmiştir. Diğer test için de belirtildiği gibi, DE ve ABC algoritmaları seyreklik seviyesi k 'nın 12'den büyük olduğu durumlarda da sıfıra yakın hata ile işareti geri oluşturduğu koşmalara sahiptir. Ancak bütün koşmaların başarılı olduğu nokta $k=12$ 'ye kadardır. Bu algoritmalarından sonra GA ve OMP algoritmalarının performanslarının iyi

olduğu söylenebilir. Bu algoritmalarla; $N=64$, $M=32$ ve $k=10$ durumuna kadar bütün koşmalarda sifıra yakın hata ile işaret geri elde edilmiştir. Daha sonra PSO algoritması ve son olarak da PDIP algoritması minimum hata ile sonuca yakınsayan algoritmalar olarak sayılabilir.

Bu değerlendirmede sezgisel algoritmalarla klasik algoritmaları kıyaslarken sezgisel olanların sonuçlarının 30 koşmanın ortalaması olarak klasiklerin ise bir defa koşmayla elde edildiği göz önünde bulundurulmalıdır. Örneğin, 30 koşmanın 29'unda MSE değeri, 10^{-30} civarında olmasına rağmen sadece bir koşmada 10^{-2} gibi bir değer alırsa bunların ortalaması 10^{-4} seviyesinde olacaktır. Ayrıca Şekil 5.14 ve Şekil 5.23'de verilen çubuk grafiklerinde de görüldüğü gibi genel olarak sezgisel algoritmalarla geri elde edilen sinyalin MSE'si sıfır olmayan noktalarda klasiklerden daha yüksek olmaktadır. Bu durum sezgisel algoritmaların amaç fonksiyonlarıyla alakalıdır. Şöyle ki, CS teorisinde Φ ölçüm matrisi ve y gözlem vektörü bilinmektedir ve x işareti algoritmalar yardımıyla geri elde edilmeye çalışılmaktadır. Önerilen metotta sezgisel algoritmaların amacı y vektörünü sağlayan Φ matrisinin sütun vektörlerinin indislerini bulmaktır. X işareti bu doğru indisler kullanılarak hesaplanır. Dolayısıyla sezgisel algoritmaların amaç fonksiyonu $(y - y')$ matrisini minimize etmek üzerinedir. $\|y - y'\|_2$ ifadesinde minimumu sağlayan indisler tam olarak bulunamadığında, yani hatanın sıfır olmadığı durumda, doğru olmayan herhangi bir indis MSE (x, x') değerinin sezgisel yöntemlerde klasik yöntemlere göre daha yüksek olmasını sağlamıştır. Ancak genel olarak değerlendirilecek olursa, buraya kadar gerçekleştirilen testlerde sezgisel algoritmaların sinyali sifıra çok yakın bir hata ile geri elde ettiği ölçüm ve seyreklik değişimleri aralığı klasik algoritmalara göre daha geniştir. Gürültülü ölçümün söz konusu olmadığı durumlarda yani yalnızca test koşullarında sinyali az hata ile elde etme kavramı yoktur. Gürültünün olmadığı durumda, elde edilen sinyal orijinal sinyalle kıyaslandığında hata varsa sinyal doğru olarak elde edilememiş demektir. Dolayısıyla test problemlerinde elde edilen sonuçları buna göre değerlendirmek gerekir.

5.4 Gürültülü Ölçümden Sinyalin Geri Elde Edilmesi

Tez çalışmasında buraya kadar olan testlerde ölçümlerin gürültüsüz ortamda gerçekleştirildiği düşünülmüştür ve sonuçlar bu duruma göre verilmiştir. Ancak algoritmaların gürültü söz konusu olduğu durumlarda da iyi sonuç vermesi gerekir yani

algoritmaların gürbüz olması da istenen bir konudur. Pratik uygulamalar söz konusu olduğunda gürültü her zaman vardır. En azından, sinyali algılamak ölçüm yapan mekanizmalarda vardır. Gürültülü ortamlarda, gürültü terimi aşağıdaki denklemde olduğu gibi ölçüme eklenir:

$$y = \Phi x + n \quad (5.1)$$

Bu tez çalışmasında n , “beyaz gauss gürültüsü” olarak sinyalin gürültüye oranı (Signal to Noise Ratio-SNR) mertebelerinde, elde edilen y vektörüne eklenmiştir. SNR aşağıdaki formülde verildiği gibi hesaplanmaktadır.

$$SNR = 20 * \log\left(\frac{\|\Phi x\|_2}{\|n\|_2}\right) \quad (5.2)$$

Gürültü içeren ölçümlerden sinyalin geri elde edilmesi incelemesi için üretilen sinyal $N=256$, $M=64$ ve $k=16$ özelliğine sahiptir. y gözlem vektörüne yani ölçümlere sırasıyla 10 dB., 15 dB., 20 dB., 25 dB. SNR seviyelerine sahip beyaz gauss gürültüsü eklenmiştir. Daha sonra bu gürültü içeren ölçümlerden sinyali geri elde etmek için üst kısımlarda değerlendirilen aynı algoritmalar kullanılmıştır. Her bir gürültü seviyesi için kullanılan yedi algoritma 30 defa koşturulmuştur. Test sinyali ve eklenen gürültüler bütün algoritmalar için aynı olarak alınmıştır. Bu teste ait algoritmaların performans sonuçları Tablo 5.7’de verilmiştir. Tabloda kullanılan tüm algoritmalar için 30 koşmaya ait MSE’nin ortalaması, standart sapması ve geri oluşturulan sinyalle orijinal sinyal arasındaki korelasyonun ortalaması verilmiştir.

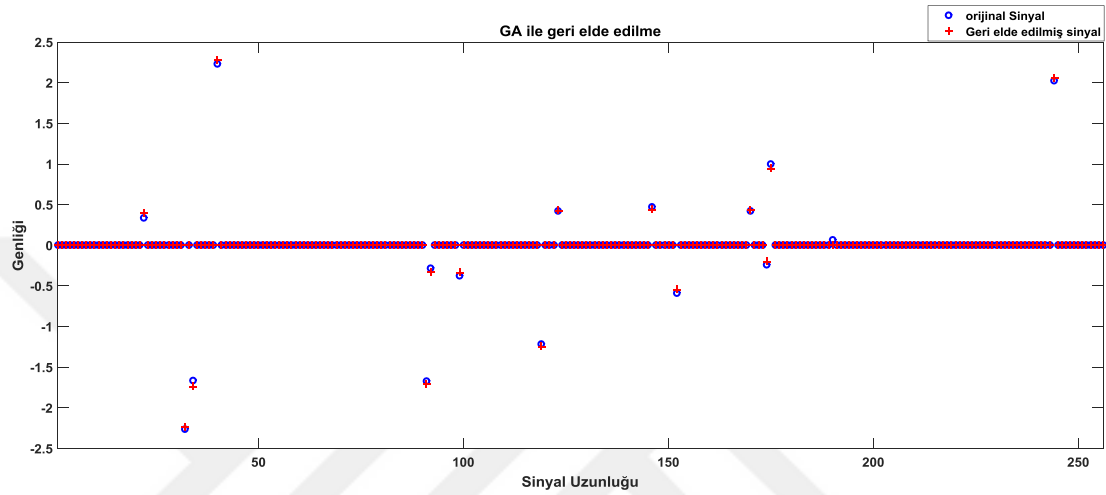
Tablo 5.7 incelenecek olursa verilen değerlere göre, algoritmaların gürültülü ölçümden sinyali geri oluşturma değerlerinin birbirine yakın olduğu görülecektir. Klasik ve popülasyon tabanlı algoritmalar birbirleriyle genel bir değerlendirmeye tabii tutulursa OMP algoritmasının bütün gürültü seviyelerinde diğer algoritmalara göre daha iyi sonuç verdiği söylenebilir. OMP’den sonra DE ve ABC algoritmaları da yakın performansa sahiptir. Aralarındaki fark az olmakla beraber, bazı gürültü seviyelerinde ABC algoritması bazılarında ise DE algoritması daha iyi sonuç vermektedir. Bu algoritmalarından sonra diğerlerindeki sıralama gürültü seviyesine göre değişmektedir. Bu kısımda yapılan incelemelerde sonular birbirlerine çok yakın olarak çıkmıştır. Dolayısıyla gürültülü ölçümlerde sezgisel algoritmaların, çok daha yaygın bir kullanım alanı bulan klasik algoritmalara alternatif olarak sinyalin geri edilmesinde

kullanılabileceği söylenebilir. Tablo 5.7’de farklı gürültü seviyeleri için seyrek sinyalin geri oluşturulması işleminde elde edilen sonuçlar verilmektedir.

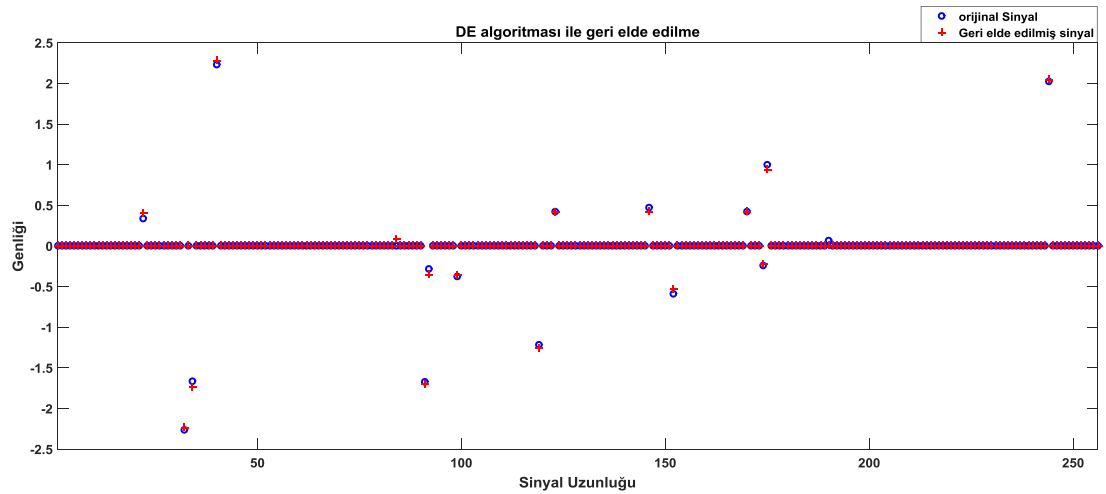
Tablo 5.7. Gürültü içeren ölçümlerde seyrek sinyalin geri oluşturulması aşaması için elde edilen sonuçlar.

SNR=10 dB	MSE(x, x') (Ortalama)	MSE(x, x') (Standart sapma)	Korelasyon (Ortalama)
GA	2.840×10^{-2}	1.090×10^{-2}	0.8674
DE	2.130×10^{-2}	3.000×10^{-3}	0.9011
PSO	2.620×10^{-2}	1.452×10^{-2}	0.8785
ABC	2.070×10^{-2}	4.582×10^{-3}	0.9011
OMP	1.260×10^{-2}	5.312×10^{-18}	0.9424
CoSaMP	2.380×10^{-2}	7.011×10^{-18}	0.8776
PDIP	1.440×10^{-2}	5.320×10^{-18}	0.7514
SNR=15 dB	MSE(x, x') (Ortalama)	MSE(x, x') (Standart sapma)	Korelasyon (Ortalama)
GA	9.200×10^{-3}	8.100×10^{-3}	0.9544
DE	4.800×10^{-3}	2.000×10^{-3}	0.9787
PSO	1.119×10^{-2}	2.284×10^{-2}	0.9486
ABC	3.681×10^{-3}	1.480×10^{-3}	0.9830
OMP	2.901×10^{-3}	4.412×10^{-19}	0.9887
CoSaMP	7.282×10^{-3}	4.443×10^{-18}	0.9599
PDIP	4.565×10^{-3}	3.512×10^{-18}	0.9499
SNR=20 dB	MSE(x, x') (Ortalama)	MSE(x, x') (Standart sapma)	Korelasyon (Ortalama)
GA	3.400×10^{-3}	6.800×10^{-3}	0.9818
DE	7.015×10^{-4}	1.086×10^{-4}	0.9966
PSO	2.792×10^{-3}	1.060×10^{-2}	0.9851
ABC	7.931×10^{-4}	3.912×10^{-4}	0.9963
OMP	7.214×10^{-4}	2.213×10^{-19}	0.9963
CoSaMP	1.130×10^{-2}	1.812×10^{-18}	0.9363
PDIP	3.690×10^{-3}	2.641×10^{-18}	0.9866
SNR=25 dB	MSE(x, x') (Ortalama)	MSE(x, x') (Standart sapma)	Korelasyon (Ortalama)
GA	6.600×10^{-3}	2.450×10^{-2}	0.9656
DE	1.661×10^{-4}	2.756×10^{-20}	0.9992
PSO	3.331×10^{-3}	1.510×10^{-2}	0.9827
ABC	2.145×10^{-4}	1.621×10^{-4}	0.9989
OMP	1.612×10^{-4}	2.742×10^{-20}	0.9993
CoSaMP	5.614×10^{-4}	4.423×10^{-19}	0.9969
PDIP	2.445×10^{-3}	1.821×10^{-18}	0.9805

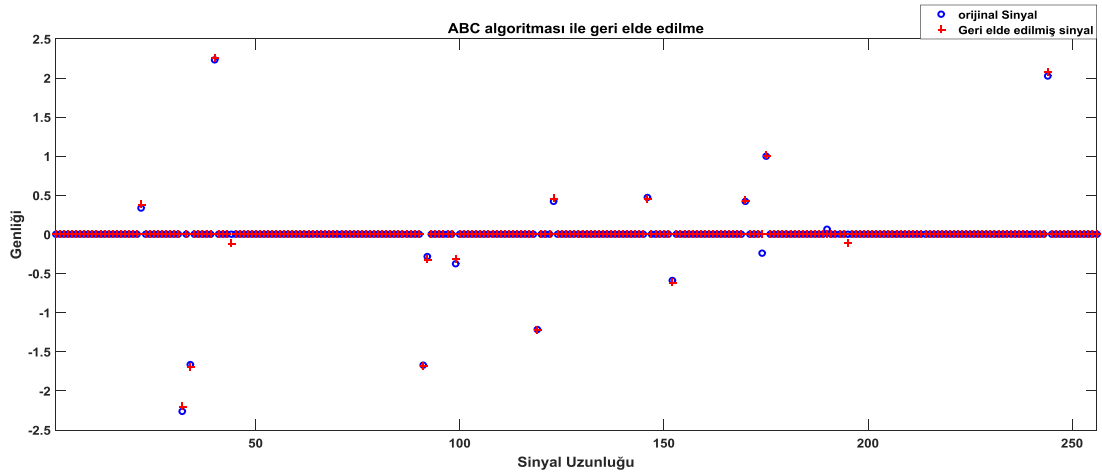
Şekil 5.24 ile 5.30 arasında 25 dB.'lik gürültü içeren ölçümlerden klasik ve sezgisel algoritmalar kullanılarak yeniden elde edilen sinyal ile orijinal sinyalin birlikte değerlendirildiği durumlar gösterilmiştir. Şekil 5.24'de gürültü içeren ölçümden GA, Şekil 5.25'de DE algoritması, Şekil 5.26'da ABC algoritması ve Şekil 5.27'de de PSO algoritması kullanılarak geri elde edilmiş sinyal ve orijinal sinyal gösterilmektedir.



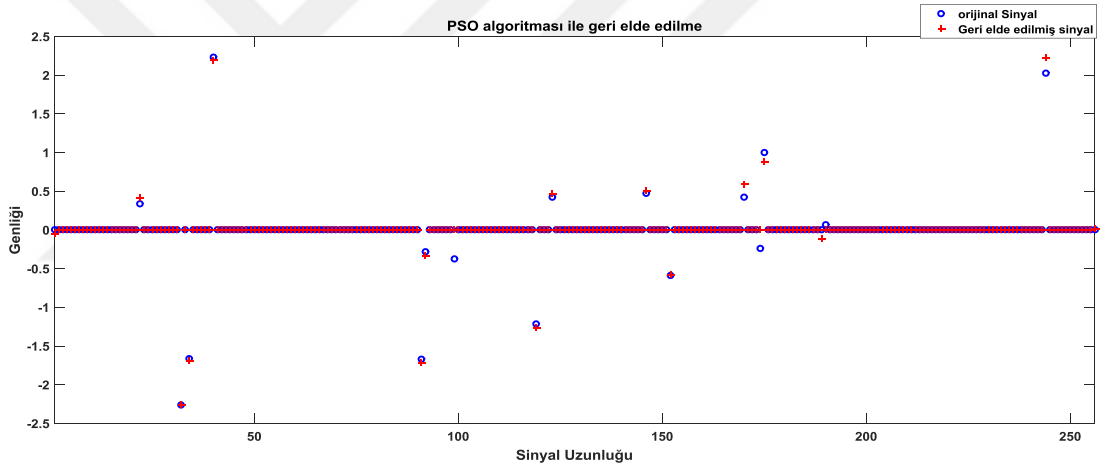
Şekil 5.24. GA geri ile geri elde edilmiş sinyal ve orijinal sinyal.



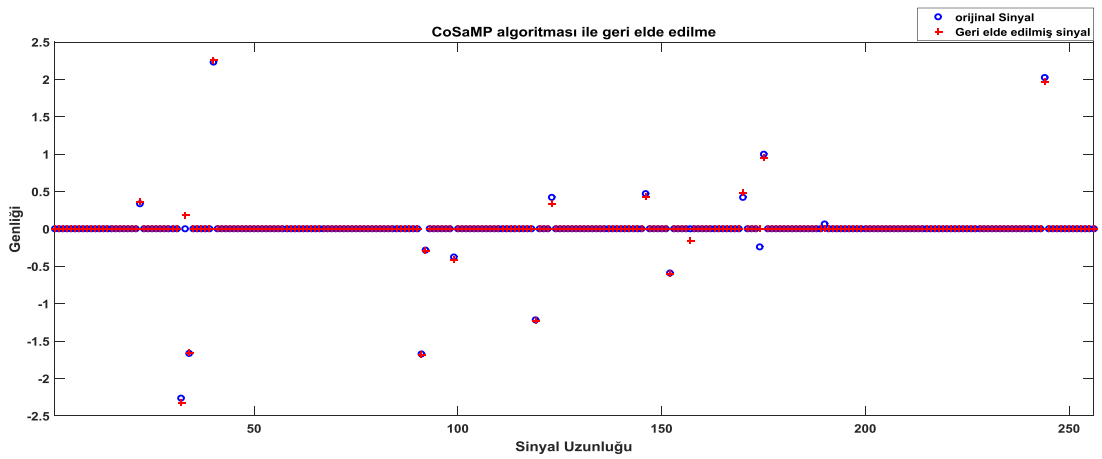
Şekil 5.25. DE algoritması ile geri elde edilmiş sinyal ve orijinal sinyal.



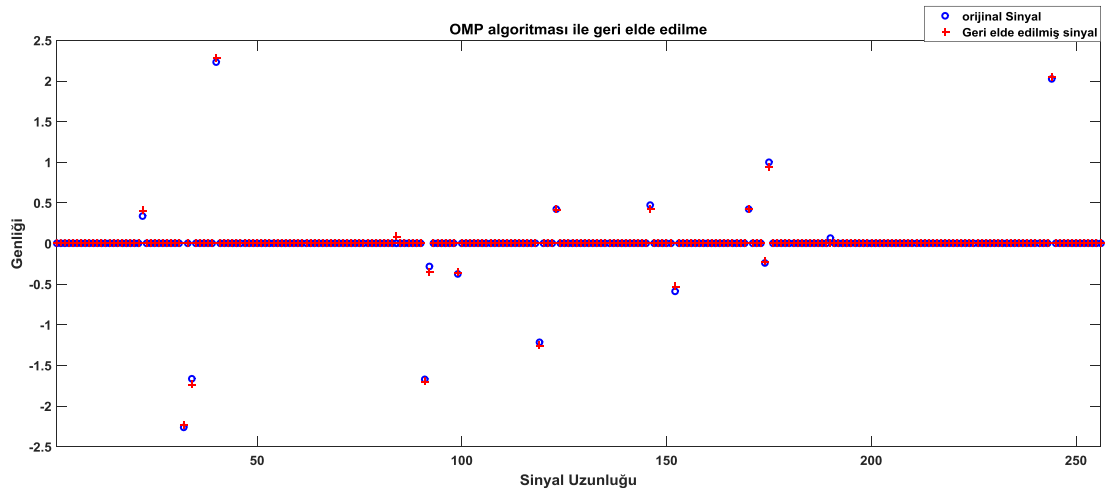
Şekil 5.26. ABC algoritması ile geri elde edilmiş sinyal ve orijinal sinyal.



Şekil 5.27. PSO algoritması ile geri elde edilmiş sinyal ve orijinal sinyal.

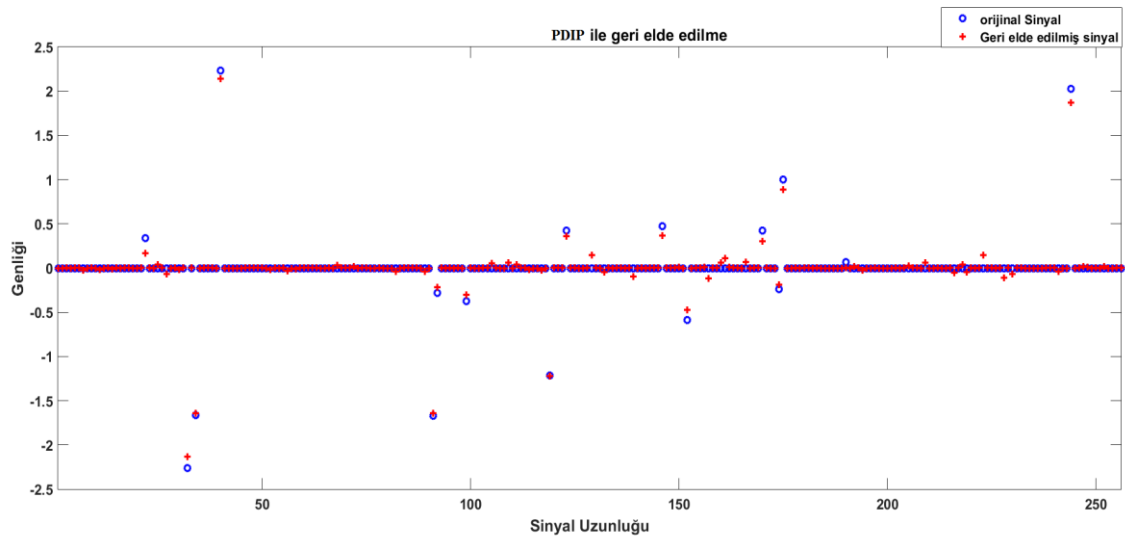


Şekil 5.28. CoSaMP algoritması ile geri elde edilmiş sinyal ve orijinal sinyal.



Şekil 5.29. OMP algoritması ile geri elde edilmiş sinyal ve orijinal sinyal.

Klasik algoritmaların gürültülü durumlar için sinyal uzunluğuna bağlı olarak genişliğinin nasıl değiştiğine ilişkin incelemeler; Şekil 5.28’de CoSaMP algoritması, Şekil 5.29’da OMP algoritması ve Şekil 5.30’da PDIP algoritması için verilmiştir. Bu şekillerde, algoritmalar ile geri elde edilmiş sinyal ve orijinal sinyal arasındaki değişim gösterilmektedir.



Şekil 5.30. PDIP algoritması ile geri elde edilmiş sinyal ve orijinal sinyal.

Bu verilen şekiller incelenecek de kullanılan algoritmaların hepsi 16 indisin 15 tanesini doğru olarak bulmuştur. Bir indisin yerine her algoritma yanlış olarak farklı bir indis bulmuştur. PDIP algoritması hariç Tablo 5.7’deki 25 dB’lik gürültü seviyesi için algoritmalarla ait performans değerlerinin farklı olması bu sebeptendir. Şekiller

incelenecek olursa bulunamayan indisin 190'ıncı indis olduğu görülür. Orijinal sinyalde 190'ıncı indise ait olan değer çok küçüktür ve indisin gözlem vektörüne katkısı gürültü olduğundan dolayı deterministik değildir. Bu yüzden algoritmalar tarafından ayırt edilememiştir.

5.5. Diğer çalışmalarla karşılaştırma

Seyrek işareti geri oluşturma işlemi, evrimsel algoritma tarafından literatürden Chakraborty'nin yapmış olduğu çalışma ile karşılaştırılmıştır. Chakraborty'nin önerdiği metot, bu tezde önerilen metottan farklıdır [14]. Makalede uygulanan test çalışmalarının aynısı önerilen metot için de uygulanmış ve elde edilen sonuçlar Tablo 5.8'de verilmiştir.

Table 5.8. Önerilen metodun Chakraborty'nin çalışmasıyla karşılaştırılması.

N=50 M=30 K=8	Chakraborty [14]	GA	DE	PSO	ABC
50 üzerinden başarı oranı	34	49	50	44	50
$\ y - y'\ _2$ (ortalama)	2.89×10^{-04}	1.79×10^{-14}	1.98×10^{-14}	1.72×10^{-14}	1.83×10^{-14}
$\frac{\ y - y'\ _2}{\ y\ _2}$ (ortalama)	9.27×10^{-05}	1.67×10^{-15}	1.24×10^{-15}	1.17×10^{-15}	1.27×10^{-15}
$\ x - x'\ _2$ (ortalama)	3.78×10^{-04}	3.50×10^{-15}	3.73×10^{-15}	3.08×10^{-15}	3.38×10^{-15}
$\frac{\ x - x'\ _2}{\ x\ _2}$ (ortalama)	1.20×10^{-04}	1.50×10^{-15}	1.14×10^{-15}	1.14×10^{-15}	1.29×10^{-15}
N=50 M=30 K=8	Chakraborty	GA	DE	PSO	ABC
$\ y - y'\ _2$ (std. sap.)	4.31×10^{-04}	1.58×10^{-14}	1.32×10^{-14}	1.54×10^{-14}	1.75×10^{-14}
$\frac{\ y - y'\ _2}{\ y\ _2}$ (std. sap.)	3.34×10^{-05}	1.48×10^{-15}	8.34×10^{-16}	1.11×10^{-15}	1.04×10^{-15}
$\ x - x'\ _2$ (std. sap.)	6.20×10^{-04}	3.22×10^{-15}	2.62×10^{-15}	2.30×10^{-15}	2.64×10^{-15}
$\frac{\ x - x'\ _2}{\ x\ _2}$ (std. sap.)	1.46×10^{-02}	1.38×10^{-15}	8.03×10^{-16}	9.35×10^{-16}	9.38×10^{-16}

Chakraborty çalışmasında başarı oranını 10^{-4} olarak seçmiştir ve algoritmayı 50 kez çalıştırmıştır. Tablo 5.8'den görüldüğü gibi, bu tezde önerilen metot dört algoritma için de Chakraborty'nin önerdiği metottan daha iyi sonuç vermiştir. Bunun nedeni Chakraborty'nin çalışması aynı anda hem x sinyalinin sıfır olmayan elemanlarının indislerini ve bu indislere ait değerleri optimize etmeye çalışmıştır. Bu durumda da küresel optimum yerine bölgesel bir optimumu bulmuştur. Tezde önerilen metot ise sadece sıfır olmayan indisleri bulup daha sonra ölçüm matrisinin bu indislere karşılık gelen sütun vektörlerinin sözde tersini alarak sinyali geri oluşturduğu için çözüm küresel minimumdur.

6. BÖLÜM

SONUÇLAR TARTIŞMA VE ÖNERİLER

CS teoremi, geleneksel örnekleme metotlarından daha küçük örnekleme oranı kullanarak seyrek sinyali geri oluşturmaktadır. CS’de seyrek sinyali geri oluştururken çeşitli algoritmalar kullanılmaktadır. Seyrek sinyali geri oluşturmak için klasik yöntemlerde l_0 normu minimizasyon yöntemi temel alınmaktadır. Ancak bu çok zor bir optimizasyon problemi olarak bilinmektedir. Eğer algılama matrisi, sınırlı izometri özelliğini sağlarsa l_0 normu yerine l_1 minimizasyon yöntemleri kullanılabilir. l_1 minimizasyon problemi de doğrusal programlama yöntemleriyle çözülebilmektedir. Seyrek sinyalin geri oluşturulmasında klasik teknikler içerisinde değerlendirilen l_1 minimizasyon algoritmaları veya açgözlü algoritmalar tercih edilmektedirler. Son zamanlarda bu yöntemlere ek olarak, sezgisel algoritmalar da seyrek sinyalin geri oluşturulmasında kullanılmaktadır.

Tez çalışmasının ilk aşamasında da seyrek sinyali geri elde etmek için l_1 minimizasyon algoritmalarından PDIP algoritması ile açgözlü algoritma sınıfından OMP ve COSAMP algoritmaları temel alınmıştır. Bu algoritmalarla sinyalin yeniden edilmesindeki başarımlar incelenerek tüm algoritmaların kendi içlerinde performans karşılaştırması yapılmıştır.

Gerçekleştirilen çalışmalara ek olarak tez çalışmasının devamında, sıkıştırılmış algılamada seyrek sinyalin geri oluşturulmasında sezgisel yöntemlerin kullanılmasına dayalı bir yöntem önerilmiştir. Bu yöntemde sezgisel algoritmaların amaç fonksiyonu, CS’de kullanılan açgözlü algoritmalarla benzer şekilde tasarlanmıştır. Önerilen yöntem, GA, DE, PSO ve ABC algoritmalarıyla kullanılarak algoritmaların seyrek sinyali geri elde etmedeki başarımları farklı problemler üzerinde gösterilmiştir. Böylece popülasyon tabanlı sezgisel algoritmaların CS’deki performansları test edilmiştir. Elde edilen benzetim sonuçları aynı zamanda klasik algoritma ile elde edilen sonuçlarla

kıyaslanmıştır. Ayrıca bu tezde karşılaştırılan popülasyon tabanlı algoritmalar ve klasik metotlar gürültü içeren durumlar için de seyrek sinyali geri elde etmek için incelenmiştir. Gürültülü durumlar için de kullanılan sezgisel algoritmalar ve klasik algoritmaların sonuçları birbirine yakın olarak elde edilmiştir. CS’de problemin çözümü için önerilen popülasyon tabanlı algoritmalar, l_1 minimizasyonu ve açgözlü algoritmalarla göre sinyali daha uzun sürede geri elde etmektedirler. Yani yöntemler daha kısa sürede sinyalin geri edilmesini sağlamaktadırlar. Bu da önerilen yöntemin sezgisel algoritmalarla kullanılırken iterasyon sayısının yüksek olması dolayısıyladır. İterasyon sayısının yüksek olması, matrislerde çarpma işleminin sayısının da artmasına sebep olmaktadır Sayısal sistemler de ise çarpma işlemi zaman alan bir işlem ve mümkün olduğunca minimum olması tercih edilir.

Elde edilen bu performans sonuçlarına göre genel olarak sezgisel algoritmaların test probleminin ölçüm sayısına ve seyreklik seviyesine göre seyrek sinyali geri elde edebildiği aralık l_1 minimizasyonu yöntemlerinden ve açgözlü algoritmalarından daha geniştir. Yani klasik metotların seyrek sinyali geri elde edemediği noktalarda, önerilen metodu kullanan popülasyon tabanlı algoritmalar sinyali geri elde edebilmiştir. Sonuç olarak önerilen yöntem, CS’de seyrek sinyalin geri oluşturulması için kullanılabilir bir yöntemdir. Özellikle klasik yöntemlerin sinyali geri elde etmesinin mümkün olmadığı durumlarda kullanılması daha avantajlı olacaktır.

Gelecekte konuyla ilgili çalışmalarda, iterasyon sayısını azaltarak dolayısıyla matrislerde çarpma işlemi sayısını azaltmaya yönelik çalışmalar planlanmaktadır. Bu durumda seyrek sinyali geri oluşturma algoritmalarından bazıları başlangıç popülasyonunu oluşturmada kullanılabilir. Böylece algoritmanın araştırma uzayı küçülerek daha kısa sürede sonuca ulaşması mümkün olabilir. Yine farklı bir çalışma olarak matrislerin sözde tersi alınırken daha az matris çarpma işlemi kullanabilecek bir yöntem geliştirilebilir. Ayrıca konu ile ilgili geliştirilen algoritmalar farklı uygulamalar üzerinde test edilebilir.

KAYNAKÇA

1. Qaisar, S., Bilal, R., Iqbal, W., Naureen, M., Lee, S., 2013. Compressive sensing: From theory to applications, a survey. **Journal of Communication Networks**, **15** (3): 443–456.
2. Pinto, S., Mendoza, L., Flórez, E., 2016. Compressive sensing in FPGA and microcontroller. **International Journal of Engineering and Technology**, **7** (6):2202–2206.
3. Donoho, D., Compressed sensing, 2006. **IEEE Trans. Inf. Theory**, **52** (4): 1289–1306.
4. Candès, E., Romberg, J., Tao, T., 2006. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. **IEEE Trans. Inf. Theory**, **52**(2): 489–509.
5. Candès, E., 2008. The restricted isometry property and its implications for compressed sensing. **Comptes Rendus Math.** **346** (9–10): 589–592.
6. Arjoune, Y., Kaabouch, N., El Ghazi, H., Tamtaoui, A., 2017. Compressive sensing: Performance comparison of sparse recovery algorithms, pp. 1-7. *2017 IEEE 7th Annual Computing and Communication Workshop and Conference*, 9-11 January, 2017, Las Vegas, NV.
7. Baraniuk, R., 2007. Compressive sensing [Lecture Notes]. **IEEE Signal Processing Magazine**, **24** (4): 118–121.
8. Hayashi, K., Nagahara, M., Tanaka, T., 2013. A user's guide to compressed sensing for communications systems. **IEICE Transaction on Communication**, **E96.B** (3): 685–712.
9. Çelik, S., Başaran, M., Erküçük S., Çırpan, A., 2016. Seyrek işaret geri oluşturma için sıkıştırılmış algılama tabanlı algoritmaların karşılaştırılması, pp. 1441–1444. *2016 24th Signal Processing and Communication Application Conference*, 16-19 Mayıs 2016, Zonguldak, Türkiye.
10. Theodoridis, S., Kopsinis, Y., Slavakis, K. Sparsity-Aware Learning and Compressed Sensing: An Overview, pp. 1271–1377. *In: Academic Press Library in Signal Processing* (Eds. Paulo S. R. Diniz, A. K. Johan, R. Chellappa, S. Theodoridis). Elsevier, Poland.

11. Candes, E., Wakin, M., 2008. An introduction to compressive sampling. **IEEE Signal Processing Magazine**, 25 (2): 21–30.
12. Candes E., Tao, T., 2005. Decoding by linear programming. **IEEE Transaction on Information Theory**, 51 (12): 4203-4215.
13. Cagliyan, F., 2013. Compressive Sensing for Radar Target Detection. Middle East Technical University, The Graduate School of Natural and Applied Sciences, Master Thesis, Ankara, 64 p.
14. Chakraborty, U., 2015. Evolutionary algorithm for compressive sensing. **International Journal of Automation and Control**, 9 (1): 61–70.
15. Candes, E., Romberg, J., 2005. L_1 -magic : Recovery of Sparse Signals via Convex Programming.
16. Becker S., Bobin, J., Candès, E., 2011. NESTA: A fast and accurate first-order method for sparse recovery. **SIAM Journal on Imaging Sciences**, 4 (1):1–39.
17. Boyd, S., Vandenberghe, L., 2010. Convex Optimization, Cambridge University Press, New York, USA, 716 pp.
18. Beck, A., Teboulle, M., 2009. A fast iterative shrinkage-thresholding algorithm. **Society for Industrial and Applied Mathematics Journal on Imaging Sciences**, 2 (1): 183–202.
19. Hale, E., Yin, W., Zhang, Y., 2008. Fixed-point continuation for $-l_1$ minimization: Methodology and convergence. **SIAM Journal on Optimization**, 19 (3): 1107–1130.
20. Yin, W., Osher, S., Goldfarb, D. Darbon, J., 2008. Bregman iterative algorithms for l_1 -Minimization with applications to compressed sensing. **SIAM Journal on Imaging Sciences**, 1 (1): 143–168.
21. Figueiredo, M., Nowak, R., Wright, S., 2007. Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. **IEEE Journal of Selected Topics in Signal Processing**, 1 (4): 586–597, 2007.
22. Bioucas-Dias, J., Figueiredo, M., 2007. A New TwIST: Two-Step Iterative Shrinkage/Thresholding Algorithms for Image Restoration. **IEEE Transaction on Image Processing**, 16 (12): 2992–3004.
23. Wright, S., Nowak, R., Figueiredo, A., 2009. Sparse reconstruction by separable approximation. **IEEE Transaction on Signal Processing**, 57 (7): 2479–2493.

24. Mallat, S., Zhang, Z., 1993. Matching pursuits with time-frequency dictionaries. **IEEE Transaction on Signal Processing**, **41** (12): 3397–3415.
25. Tropp, J., Gilbert, A., 2007. Signal recovery from random measurements via orthogonal matching pursuit. **IEEE Transaction on Information Theory**, **53** (12): 4655–4666.
26. Blumensath, T., Davies, M., 2008. Gradient pursuits. **IEEE Transaction on Signal Processing**, **56** (6): 2370–2382.
27. Donoho, D., Tsaig, Y., Drori, I., Strack, J., 2012. Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit. **IEEE Transaction on Information Theory**, **58** (2): 1094–1121.
28. Needell, D., Vershynin, R., 2009. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. **Foundation of Computational Mathematics**, **9** (3): 317–334.
29. Needell, D., Vershynin, R., 2010. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. **IEEE Journal of Selected Topics in Signal Processing**, **4** (2): 310–316.
30. Needell, D., Tropp, J., 2009. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. **Applied and Computational Harmonic Analysis**, **26** (3): 301–321.
31. Liu, F., Lin, L., Jiao, L., Li, L., Yang, S., Hou, B., Ma, H., Yang, L., Xu, J., 2015. Nonconvex compressed sensing by nature-inspired optimization algorithms. **IEEE Transaction on Cybernetics**, **45** (5): 1028–1039.
32. Du, X., Cheng, L., Liu, L., 2013. A swarm intelligence algorithm for joint sparse recovery. **IEEE Signal Processing Letters**, **20** (6): 611–614.
33. Shah, J., Qureshi, I., Khaliq, A., Omer, H., 2014. Sparse signal recovery based on Hybrid Genetic Algorithm. **Research Journal of Recent Sciences**, **3** (5): 86–93.
34. Fengmin, X., Shanhe, W., 2013. A hybrid simulated annealing thresholding algorithm for compressed sensing. **Signal Processing**, **93** (6): 1577–1585.
35. Li, D., Li, M., Shen, Y., Wang, Y., Wang, Y., 2013. GA-BFO based signal reconstruction for compressive sensing pp. 1023–1028. *2013 IEEE International Conference on Information and Automation (ICIA)*, 26–28 August 2013, Yinchuan, China.

36. Du, X., Cheng, L., Cheng, G., 2014. A heuristic search algorithm for the multiple measurement vectors problem. **Signal Processing**, **100**: 1–8.
37. Lei, J., Liu, W., Liu, S., Liu, Q., 2014. Multiscale reconstruction algorithm for compressed sensing. **ISA Transactions**, **53** (4): 1152–1167.
38. Brajovi, M., Member, S., Lutovac, B., Orovi, I., 2016. Sparse signal recovery based on concentration measures and genetic algorithm, pp. 1-4. *2016 13th Symposium on Neural Networks and Applications (NEUREL)*, November 22-24, 2016, Belgrade, Serbia.
39. Ghadyani, M., Shahzadi, A., 2017. Adaptive joint sparse recovery algorithm based on Tabu Search, **Neurocomputing**, **224** (October 2016): 9–18.
40. Zhou, S., Liu, Y., Zhang, W., 2017. Compressed sensing of image signals with threshold processing, **Optik**, **131**: 671–677.
41. Feng, H., 2016. Multiple target localization in WSNs using compressed sensing reconstruction based on ABC algorithm, pp. 07008. *ICFST 2016 MATEC Web of Conferences*, 2016.
42. Li, L., Yao, X., Stolkin, R., Gong, M., He, S., 2014. An evolutionary multiobjective approach to sparse reconstruction. **IEEE Transaction on Evolutionary Computation**, **18** (6): 827–845.
43. Li, H., Fan, Y., Zhang, Q., Xu, Z., Deng, J., 2016. A multi-phase multiobjective approach based on decomposition for sparse reconstruction, pp. 601-608. *IEEE Congress on Evolutionary Computation (CEC)*, 24-29 July, 2016, Vancouver, BC.
44. Zhou, Y., Kwong, S., Guo, H., Zhang, X., Zhang, Q., 2017. A two-phase evolutionary approach for compressive sensing reconstruction. **IEEE Transaction on Cybernetics**, **47** (9): 2651–2663.
45. Available: <http://users.ece.gatech.edu/~justin/l1magic/>.
46. Tropp, J., Gilbert, A., 2007. Signal recovery from random measurements via orthogonal matching pursuit. **IEEE Transaction on Information Theory**, **53** (12): 4655–4666.
47. Davenport M., Wakin, M., 2010. Analysis of orthogonal matching pursuit using the restricted isometry property. **IEEE Transaction on Information Theory**, **56** (9): 4395–4401.

48. Karaboga, D., Akay, B., 2009. A comparative study of Artificial Bee Colony algorithm. **Applied Mathematics and Computation**, **214** (1): 108–132.
49. Srinivas, M., Patnaik, L., 1994. Genetic algorithms: a survey. **Computer**, **27** (6): 17–26.
50. Storn, R., Price, K., 1997. Differential Evolution-A simple and efficient heuristic for global optimization over continuous spaces. **Journal of Global Optimization**, **11**(4): 341–359.
51. Karaboga, D., Okdem, S., 2004. A simple and global optimization algorithm for engineering problems: Differential Evolution algorithm. **Turkish Journal of Electrical Engineering**, **12** (1): 53–60.
52. Karaboga, D., 2014. Yapay Zeka Optimizasyon Algoritmaları. Nobel Akademik Yayıncılık, Kayseri, 246 s.
53. Eberhart, R., Kennedy, J., 1995. A new optimizer using particle swarm theory, pp. 39-43. Sixth International Symposium on Micro Machine and Human Science, 4-6 October, 1995, Nagoya.
54. Poli, R., Kennedy, J., Blackwell, T., 2007. Particle swarm optimization. **Swarm Intelligence**, **1** (1): 33–57.
55. Karaboga, D., 2005. An idea based on honey bee swarm for numerical optimization, Technical report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005.
56. Baraniuk, R., Davenport, M., DeVore, R., Wakin, M., 2008. A simple proof of the Restricted Isometry Property for random matrices. **Constructive Approximation**, **28** (3): 253–263.
57. Needell, D., Tropp, J., Vershynin, R., 2008. Greedy signal recovery review, pp. 1048-1050. *2008 42nd Asilomar Conference on Signals, Systems and Computers*, 26-29 October, 2008, Pacific Grove, CA.
58. Available: <http://mf.erciyes.edu.tr/abc/software.html>.

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı, Soyadı: Murat Emre ERKOÇ

Uyruğu: Türkiye Cumhuriyeti

Doğum Tarihi ve Yeri: 1990, Sivrihisar

email: merkoc@erciyes.edu.tr

Yazışma Adresi: Erciyes Üniversitesi, Mühendislik Fakültesi 38039

Melikgazi/KAYSERİ

EĞİTİM

Derece	Kurum	Mezuniyet Tarihi
Lisans	Orta Doğu Teknik Üniversitesi, Kuzey Kıbrıs Kampüsü, Elektrik- Elektronik Mühendisliği	2015
Lise	Eskişehir Atatürk Lisesi, Eskişehir	2007

İŞ DENEYİMLERİ

Yıl	Kurum	Görev
2015- Halen	Erciyes Üniversitesi Mühendislik Fakültesi	Araştırma Görevlisi

YABANCI DİL

İngilizce

YAYINLAR

1. Erkoç M.E., Karaboğa N., "Comparison of Minimization and Greedy Algorithms for Recovering Sparse Frequency Domain Signals", International Workshop, "Mathematical Methods in Engineering, MME-2017 ", ANKARA, TÜRKİYE, 27-29 Nisan 2017, pp.67-67 [\(Link\)](#)

2. Erkoç M.E., Karaboğa N., "AYRIK DALGACIK ANALİZİ VE DALGACIK PAKET ANALİZİ YÖNTEMLERİNİ KULLANARAK EEG SİNYALİNDE GÜRÜLTÜ GİDERİMİ", Otomatik Kontrol Ulusal Toplantısı (TOK 2016), ESKİŞEHİR, TÜRKİYE, 29 Eylül - 1 Ekim 2016, ss.165-168.
3. Koçkanat S., Koza T., Karaboğa N., Erkoç M.E., "Noise cancellation application of ECG signal using artificial bee colony algorithm", 2016 24th Signal Processing and Communication Application Conference, SIU 2016 - Proceedings, ZONGULDAK, TÜRKİYE, 16 Mayıs - 19 Ekim 2016, ss.1017-1020.
4. Erkoç M.E., Karaboğa N., Küçüköz N., "EEG Sinyalindeki Gürültünün Giderilmesinde Ayırık Dalgacık Ve Ayırık Durağan Dalgacık Dönüşümlerinin Performanslarının İncelenmesi", Akıllı Sistemlerde Yenilikler ve Uygulamaları Sempozyumu (ASYU2016), DÜZCE, TÜRKİYE, 29 Eylül - 1 Ekim 2016, ss.272-275.
5. Erkoç M.E., Özdemir G., Karaboğa N., "Comparison of Reconstruction Algorithms for Sparse Signal Reconstruction from Noisy Measurement", International Advanced Researches and Engineering Congress-IAREC2017, OSMANİYE, TÜRKİYE, 16-18 Kasım 2017, pp.1183-1183 (Link)
6. Özdemir G., Erkoç M.E., Karaboğa N., "Analysis of the Effect of the Stop Band and Transition Band Frequency Values on the Performance of the Filter Bank Design ", International Advanced Researches and Engineering Congress, OSMANİYE, TÜRKİYE, 16-18 Kasım 2017, pp.1184-1184 (Link)