

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

**ÇOK TEPELİ YAPAY ARI KOLONİ OPTİMİZASYON ALGORİTMASI İLE
YER RADARI GÖRÜNTÜLERİNDEKİ SAÇILMA HİPERBOLLERİNİN
OTOMATİK BULUNMASI**

YÜKSEK LİSANS TEZİ

Taymaz RAHKAR FARSHI

Temmuz 2013
TRABZON

KARADENİZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

İSTATİSTİK VE BİLGİSAYAR BİLİMLERİ ANABİLİM DALI

ÇOK TEPELİ YAPAY ARI KOLONİ OPTİMİZASYON ALGORİTMASI İLE
YER RADARI GÖRÜNTÜLERİNDEKİ SAÇILMA HİPERBOLLERİNİN
OTOMATİK BULUNMASI

Taymaz RAHKAR FARSHI

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsünde
"YÜKSEK LİSANS (İSTATİSTİK)"
Unvanı Verilmesi İçin Kabul Edilen Tezdir.

Tezin Enstitüye Verildiği Tarih : 25.06.2013
Tezin Savunma Tarihi : 12.07.2013

Tez Danışmanı : Yrd. Doç. Dr. Orhan KESEMEN

Trabzon 2013

Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü
İstatistik ve Bilgisayar Bilimleri Anabilim Dalında
Taymaz RAHKAR FARSHI tarafından hazırlanan

ÇOK TEPELİ YAPAY ARI KOLONİ OPTİMİZASYON ALGORİTMASI İLE
YER RADARI GÖRÜNTÜLERİNDEKİ SAÇILMA HİPERBOLLERİNİN
OTOMATİK BULUNMASI

başlıklı bu çalışma, Enstitü Yönetim Kurulunun 25/ 06/ 2013 gün ve 1511 sayılı
kararıyla oluşturulan jüri tarafından yapılan sınavda
YÜKSEK LİSANS TEZİ
olarak kabul edilmiştir.

Jüri Üyeleri

Başkan	: (Unvan, Ad Soyad)	Doç.Dr.Türkan ERBAY DALKILIÇ
Üye	: (Unvan, Ad Soyad)	Doç. Dr. Selçuk Han AYDIN
Üye	: (Unvan, Ad Soyad)	Yrd. Doç. Dr. Orhan KESEMEN

Prof. Dr. Sadettin KORKMAZ
Enstitü Müdürü

ÖNSÖZ

Bu tez, Karadeniz Teknik Üniversitesi Fen Bilimleri Enstitüsü İstatistik ve Bilgisayar Bilimleri Anabilim Dalı, Yüksek Lisans Programı'nda hazırlanmıştır. Bu tez, Çok Tepeli Yapay Arı Koloni Optimizasyon Algoritması ile Yer Radarı Görüntülerindeki Saçılma Hiperbollerinin Otomatik Bulunması için yapılmıştır. Tez çalışma süresinde değerli yardım ve katkılarıyla beni yönlendiren danışman hocam Yrd. Doç. Dr. Orhan KESEMEN 'ne, teşekkürlü bir borç bilirim.

Çalışmalarım boyunca beni hiçbir zaman yalnız bırakmayan eşim Sara'ya maddi ve manevi desteklerini esirgemeyen anne ve babama teşekkür ederim. Bu tezin, bundan sonraki çalışmalara katkı sağlamasını temenni ederim.

Taymaz RAHKAR FARSHI

Trabzon 2013

TEZ BEYANNAMESİ

Yüksek Lisans Tezi olarak sunduğum “Çok Tepeli Yapay Arı Koloni Optimizasyon Algoritması ile Yer Radarı Görüntülerindeki Saçılma Hiperbollerinin Otomatik Bulunması” başlıklı bu çalışmayı baştan sona kadar danışmanım Yrd. Doç. Dr. Orhan KESEMEN’in sorumluluğunda tamamladığımı, verileri/örnekleri kendim topladığımı, deneyleri/analizleri ilgili laboratuvarlarda yaptığımı/yaptırdığımı, başka kaynaklardan aldığım bilgileri metinde ve kaynakçada eksiksiz olarak gösterdiğimi, çalışma sürecinde bilimsel araştırma ve etik kurallara uygun olarak davrandığımı ve aksinin ortaya çıkması durumunda her türlü yasal sonucu kabul ettiğimi beyan ederim. 25/06/2013

Taymaz RAHKAR FARSHI

İÇİNDEKİLER

	Sayfa No
ÖNSÖZ	III
TEZ BEYANNAMESİ.....	IV
İÇİNDEKİLER.....	V
ÖZET	VIII
SUMMARY	IX
ŞEKİLLER DİZİNİ	X
SEMBOLLER DİZİNİ	XIII
1. GENEL BİLGİLER.....	1
1.1. Giriş	1
1.2. Yer Radarı Verilerinin Toplanması	2
1.2.1. Ortak-Orta Nokta Ölçümleri.....	2
1.2.2. Ortak-Açılım Ölçümleri.....	3
1.3. Hough Dönüşümü	3
1.3.1. Düz Çizgiler İçin Hough Dönüşümü	3
1.3.2. Çemberler İçin Hough Dönüşümü.....	5
1.3.3. Hiperboller İçin Hough Dönüşümü	6
1.4. Optimizasyon.....	8
1.4.1. Sürü Zekası	9
1.4.2. Parçacık Sürü Optimizasyon.....	10
1.4.3. PSO Üzerinde Çalışmış Bazı Çok Tepeli Optimizasyon Yöntemleri	11
1.5. Yapay Arı Koloni Optimizasyon	13

1.5.1.	Gerçek Arıların Yiyecek Arama Davranışları	13
1.5.2.	Yiyecek Kaynakları	14
1.5.3.	İşçi Arılar	14
1.5.4.	Gezen İşçi Arılar	14
1.5.5.	Yapay Arı Kolonisi Algoritması	15
1.5.6.	Başlangıç Yiyecek Kaynağı Bölgelerinin Üretilmesi	16
1.5.7.	İşçi Arıların Yiyecek Kaynağı Bölgelerine Gönderilmesi	17
1.5.8.	Gözcü Arıların Seçiminde Kullanacakları Olasılık Değerlerinin Hesaplanması	18
1.5.9.	Gözcü Arıların Yiyecek Kaynağı Bölgelerini Seçmesi	19
1.5.10.	Tükenen Kaynağı Bırakma ve Kaşif Arı Üretimi	19
1.5.11.	ABC Algoritmasının Adımları ve Özellikleri	20
1.5.12.	ABC Algoritmasının Temel Özellikleri	21
2.	YAPILAN ÇALIŞMALAR	23
2.1.	Giriş	23
2.2.	Çok Tepeli ABC Algoritması	24
2.2.1.	ABC Yöntemiyle Çember Algılama	25
2.3.	ABC Yöntemiyle Radargram Hiperbollerin Algılanması	29
2.3.1.	Yapay Radargramdaki Hiperbollerin Algılanması	29
2.3.2.	Gerçek Radargramdaki Hiperbollerin Algılanması	38
3.	BULGULAR VE SONUÇLAR	44
4.	ÖNERİLER	45
5.	KAYNAKLAR	46

ÖZGEÇMİŞ

Yüksek Lisans

ÖZET

ÇOK TEPELİ YAPAY ARI KOLONİ OPTİMİZASYON ALGORİTMASI İLE YER
RADARI GÖRÜNTÜLERİNDEKİ SAÇILMA HİPERBOLLERİNİN OTOMATİK
BULUNMASI

Taymaz RAHKAR FARSHI

Karadeniz Teknik Üniversitesi
Fen Bilimleri Enstitüsü
İstatistik ve Bilgisayar Bilimleri Anabilim Dalı
Danışman: Yrd. Doç. Dr. Orhan KESEMEN
2013, 48 Sayfa

Bu çalışmada, sığ yeraltının araştırılmasında kullanılan yer radarı aletiyle yapılan ölçümlerde elde edilen verilerin yer altı derinlik görüntüsüne dönüştürülmesinde kullanılacak hız bilgisinin elde edilmesi amaçlanmıştır. Bunun için kullanılan Hough dönüşümünün uzun zaman alması pratik çözümleri imkansız kılmaktadır. Daha hızlı bir çözüm için stokastik bir optimizasyon yöntemi olan ABC algoritması kullanılmıştır. Bu yöntemin probleme uyarlanması için çok tepeli ABC algoritması geliştirilmiş ve daireler üzerinde gücü test edilmiştir. Daireler üzerinde başarımını artırmak için daire kenarlarını bulanıklaştırmak için Gaussian süzgeci kullanılmıştır. Geliştirilen çok tepeli ABC algoritması radargram verilerine uygulandığında başarımın artırılması için, radargram verilerine ön işlemlerden sonra Hilbert dönüşümü uygulanmıştır. Algoritmanın uygulanmasından sonra elde edilen radargram görüntüsündeki hiperbollerin hem sayıları hem de parametreleri (konum-hız) otomatik olarak belirlenmiştir.

Anahtar Kelimeler: Çok Tepeli Optimizasyon, Yapay Arı Koloni Algoritması, Çoklu Çember Algılama, Yer Radarı Hiperboller.

Master Thesis

SUMMARY

AUTOMATIC DETECTION DIFFRACTION HYPERBOLAS IN GROUND
PENETRATING RADAR IMAGES WITH MULTIMODAL ARTIFICIAL BEE
COLONY OPTIMIZATION ALGORITHM

Taymaz RAHKAR FARSHI

Karadeniz Technical University
The Graduate School of Natural and Applied Sciences
Statistical and Computer Science Graduate Program
Supervisor: Assist. Prof. Orhan KESEMEN
2013, 48 Pages

In this study, ground penetrating radar tool used in the investigation of the shallow subsurface groundwater depth view of the data obtained from the measurements used in conversion is to obtain velocity information. Because of using Hough Transformation and using long time for that, practical solutions are impossible. For faster solution, a stochastic optimization method which is a ABC algorithm type is used. In order to adapt this method, Multimodal ABC Algorithm developed and the quality of that is tested on different apartments. In order to getting better performance in around the apartments blur filters are used. In developed multimodal ABC algorithm for improving the performance on radargram data, after pre-processing the Hilbert transform is applied to the radargram data. After applying the algorithm, in the obtained radargram image, the number of hyperbolas and also the parameters of them are determined automatically.

Key Words: Multimodal Optimization, Artificial Bee Colony Algorithm, Multi Circle Detection, Ground Penetrating Radar Hyperbolas.

ŞEKİLLER DİZİNİ

	Sayfa No
Şekil 1. Genel yer radarı düzeneğinin görünümü	2
Şekil 2. Doğru için Hough dönüşümü; (a) Görüntü uzayı; (b) Hough uzayı.....	4
Şekil 3. Sayısal görüntülerde Hough dönüşümü; (a) çemberler içeren sayısal görüntü; (b) yarıçapına göre en büyük toplam değerinin (a,b) merkez ikilsine göre Hough dönüşümü.....	6
Şekil 4. Yapay radargram görüntüsü	8
Şekil 5. Yapay hiperbollerin tepe noktasının hough dönüşümüyle bulunması.....	8
Şekil 6. ABC sözde-kod algoritması.....	21
Şekil 7. Sayısal görüntülerde dairelerin bulunması (a) örnek görüntü; (b) kenar algılama yapılmış görüntü; (c) gaussian bulanıklaştırma süzgeci uygulanmış görüntü; (d) ABC algoritması uygulandığında elde edilen verilerin diyagramı; (e) kümeleme ile iki sınıfa ayrılmış diyagram; (f) bulunan en iyi çözümlerin görüntü üzerine yerleştirilmesi.	27
Şekil 8. Eklentili sayısal görüntülerde dairelerin bulunması (a) örnek görüntü; (b) ABC algoritması uygulandığında elde edilen verilerin diyagramı; (c) kümeleme ile iki sınıfa ayrılmış diyagram; (d) bulunan en iyi çözümlerin görüntü üzerine yerleştirilmiş görüntü.....	28
Şekil 9. Gürültülü sayısal görüntülerde dairelerin bulunması (a) örnek görüntü; (b) ABC algoritması uygulandığında elde edilen verilerin diyagramı; (c) kümeleme ile iki sınıfa ayrılmış diyagram; (d) bulunan en iyi çözümlerin görüntü üzerine yerleştirilmiş görüntü.....	29
Şekil 10. Yapay yer radarı görüntüsü.	30
Şekil 11. Yapay yer radarı verilerine Hilbert dönüşümü uygulandıktan sonraki görüntüsü.	31
Şekil 12. Önerilen algoritmanın 1. çevrimdeki yiyecek bölgeleri yuvarlak noktalar biçiminde gösterimi.	31
Şekil 13. Önerilen algoritmanın 34. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.	32

Şekil 14.	Önerilen algoritmanın 71. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.	32
Şekil 15.	Çözüm listesinin ayıklanması; (a) çözüm listesi diyagramı; (b) çözüm listesinin kümelenmiş diyagramı.	33
Şekil 16.	Gürültülü yapay radargram görüntüsü.	34
Şekil 17.	Gürültülü yapay radargram görüntüsüne Hilbert dönüşümü uygulaması.	34
Şekil 18.	Önerilen algoritmanın 1. çevrimdeki yiyecek bölgeleri yuvarlak noktalar biçiminde gösterimi.	35
Şekil 19.	Önerilen algoritmanın 15. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.	35
Şekil 20.	Önerilen algoritmanın 25. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.	36
Şekil 21.	Önerilen algoritmanın 40. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.	36
Şekil 22.	Önerilen algoritmanın 62. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.	37
Şekil 23.	Önerilen algoritmanın 118. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.	37
Şekil 24.	Çözüm listesinin ayıklanması; (a) çözüm listesi diyagramı; (b) çözüm listesinin kümelenmiş diyagramı.	38
Şekil 25.	Gerçek radargram görüntüsünün 1. çevrimdeki yiyecek noktalarının gösterimi.	40
Şekil 26.	Gerçek radargram görüntüsünün 17. çevrimde ilk defa limit değeri aşılarak en iyi hiperbollerin gösterimi.	40
Şekil 27.	Gerçek radargram görüntüsünün 18. çevrimdeki en iyi hiperbollerin gösterimi.	41
Şekil 28.	Gerçek radargram görüntüsünde 75. çevrimdeki en iyi hiperbollerin gösterimi.	41
Şekil 29.	Gerçek radargram görüntüsünde 128. çevrimdeki en iyi hiperbollerin gösterimi.	42
Şekil 30.	Gerçek radargram görüntüsünün 163. çevrimdeki en iyi hiperbollerin gösterimi.	42

Şekil 31. Çözüm listesinin ayıklanması; (a) çözüm listesi diyagramı; (b) çözüm listesinin kümelenmiş diyagramı.	43
--	----

SEMBOLLER DİZİNİ

GPR	: Yer radarı
COG	: Ortak açılım dizilimi
CMP	: Ortak orta nokta dizilimi
$V_m d_m$: Verilen dizinin en büyük değeri
ABC	: Yapay arı koloni algoritması
PSO	: Parçacık sürü optimizasyonu
DE	: Farksal değişim algortması
rand(0,1)	: Düzgün dağılımdan [0,1) aralığında rastgele bir sayı

1. GENEL BİLGİLER

1.1. Giriş

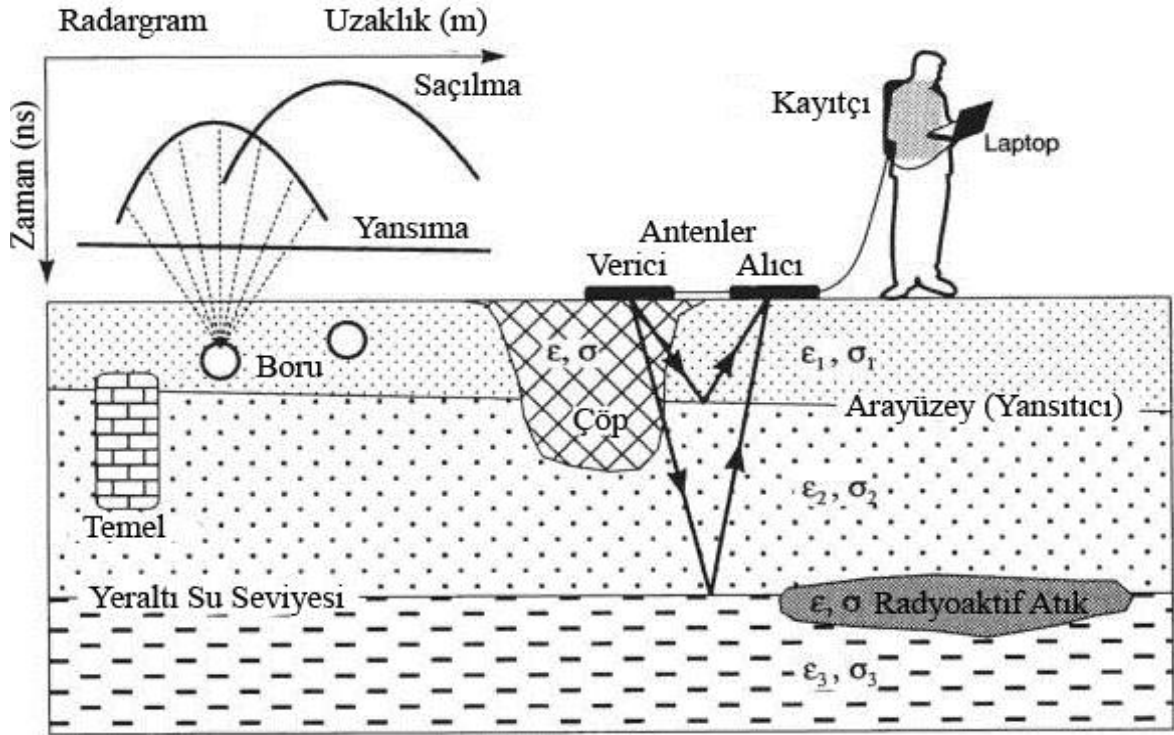
Günümüzde mühendislik çalışmalarında sığ yeraltının araştırılması giderek önem kazanmaktadır. Yer radarı (Ground Penetrating Radar, GPR), arkeolojik aramalarda, yapı sağlamlık araştırmasında, çevre kirliliği araştırmalarında özellikle tercih edilen bir mühendislik yöntemidir [1]. Kullanımı kolay, verilerin güvenilirliği yüksek, her türlü çevre ortamında kullanılabilir olması yöntemin giderek yaygınlaşmasını sağlamaktadır.

Yer radarı, 10 MHz ile 2 GHz arasında elektromanyetik dalgaları yer içine gönderen ve bunların yayılımını, yansımalarını ve saçılmasını inceleyen bir yöntemdir [2]. Yer radarı yüksek çözünürlüklü elektromanyetik bir teknik olarak yerin sığ alt yüzeyinin araştırılmasında birincil yöntem olarak kullanılır [3].

Yer radarı yönteminin esasları, sismik arama ve elektrik mühendisliğinin ilkelerine dayanır. Bu nedenle, yer radarı uygulamacısı, yeraltı arama bilgisine ve elektrik mühendisliği açısından yeterli derecede temel bilgiye sahip olmalıdır. Benzer şekilde havadan aramalar için de bu temel ilkeler geçerlidir. Bu ilkeler ışığında ortaya çıkarılan uygulama, uygulamacıyı sismik yansıma yöntemine götürür. Tüm bu uygulamaların temel ilkeleri Şekil 1’de verilmiştir. Şekil 1’de verici-alıcı anten çifti, kayıtçı ve kayıtçıya bağlı dizüstü bilgisayar gözükmektedir. Araştırılan yeraltında, atık çöpler, farklı arayüzeyler, yeraltı su seviyesi, gömülü borular, radyoaktif atıklar ve temel yapılar olabilmektedir [4].

Elektromanyetik dalga verici antenden yayılarak aynı materyal boyunca tek hızla yol aldığı varsayılır. Yeraltındaki materyalin elektriksel özellikleri (permittivity) yayılan bu dalganın hızını etkiler.

Bu dalgalar aşağı yönde yayılır ve yeraltında farklı özelliklere sahip bir ara yüzeye veya materyale çarptıklarında iki parçaya ayrılır. Bu parçalardan biri yukarı doğru yönelirken diğeri ise aşağı doğru yol almaya devam eder. Materyalden dönen bu yansıma alıcı anten tarafından yakalanır, daha sonra işlenmek ve yorumlanmak üzere elektronik ortamda sayısal sinyale çevrilerek kayıt edilir [5].



Şekil 1. Genel yer radarı düzeneğinin görünümü

1.2. Yer Radarı Verilerinin Toplanması

Uygulamada, yer radarı ölçümleri, yüzey boyunca ayrı noktalarda veya yeryüzünde antenleri sürekli taşıyarak yapılabilir. Bu iki tür ölçüm düzeneklerinden birincisi sabit ölçüm türü, antenlerin birbirinden bağımsız şekilde hareket etmesiyle, farklı noktalarda ve ayrı ölçüm alması için kurulan düzenektir. Kayan ölçüm türü ise, alıcı ve verici antenler sabit bir aralıkla yüzey boyunca elle veya bir araçla çekilerek yapılan ölçümlerdir. Sabit düzeneğinin üstünlüğü istenilen uzaklıkta ve istenilen türde veri toplanmasıyken, kayan düzeneğin üstünlüğü ise hızlı veri toplanmasıdır. Uygulamada, sabit ölçüm türü ile kayan ölçüm türü birleşik şekilde kullanılarak hem istenilen şekilde veri toplanırken hem de zamandan kazanım sağlanmış olunur. Sabit düzende ölçüm yaparak, kayan düzende en iyi uzaklık ve anten dizilimi belirlenebilir [3].

Yeraltının bilgisini elde etmede farklı veri toplama düzenekleri kullanılır. Bunlar ortak açılım (COG) ölçümleri ve ortak orta nokta (CMP) ölçümleri incelenecektir.

1.2.1. Ortak-Orta Nokta Ölçümleri

Ortak orta nokta (CMP) ölçüleriyle elektromanyetik dalga hızları bulunabilir. Bu ölçü düzeneğinde, kaynak ve alıcı arasındaki açılım (x_M) sabit orta noktaya göre (Δx) artarak gitmektedir. Böylelikle açılımdan yararlanılarak hız bilgisi elde edilebilir. Buradaki varsayım, yeraltını düz tabakalardan oluşmuş olması ve yeraltına giden elektromanyetik dalganın alıcı-verici ortak derinlik noktasından yansıdığıdır [5].

1.2.2. Ortak-Açılım Ölçümleri

Yeraltındaki nesnelere bulmak veya yeraltındaki süreksizlikleri belirlemek için, genelde ortak-açılım ölçü düzeneği yardımıyla yer radarı ölçüleri kullanılır. Burada kaynak ve alıcı antenleri (Δx) uzaklığında sabit bir aralıkla ayrılmıştır. Ölçümler, sistemin bir doğrultu boyunca sabit bir aralıkla kaydırılmasıyla yarı sürekli bir şekilde alınır ve aynı işlem tekrarlanarak doğrultu boyunca ölçüm alınır. Bu yöntem hızlı olması dolayısıyla ucuzdur. Fakat bu yöntemin temel eksikliği, ölçümlerde elde edilecek veriden yeraltındaki dalga hızı bilgisinin belirlenmesindeki zorluktur. Yine de, yeraltında bir nesne olduğunda radargramda bir saçılma hiperbolü görülmektedir. Bu hiperbolden yeraltındaki dalga hızı kestirilebilir [5]. Bu kestirim için yaygın olarak Hough dönüşümü kullanılmaktadır.

1.3. Hough Dönüşümü

Hough dönüşümü bir sayısal görüntüde bulunan doğrular ve çemberler gibi basit geometrik şekillerin bulunması için kullanılır [6]. Bu çalışmada ise yer radarı görüntülerindeki hiperbolik yaylar gibi geometrik yapıların bulunmasında kullanılmıştır.

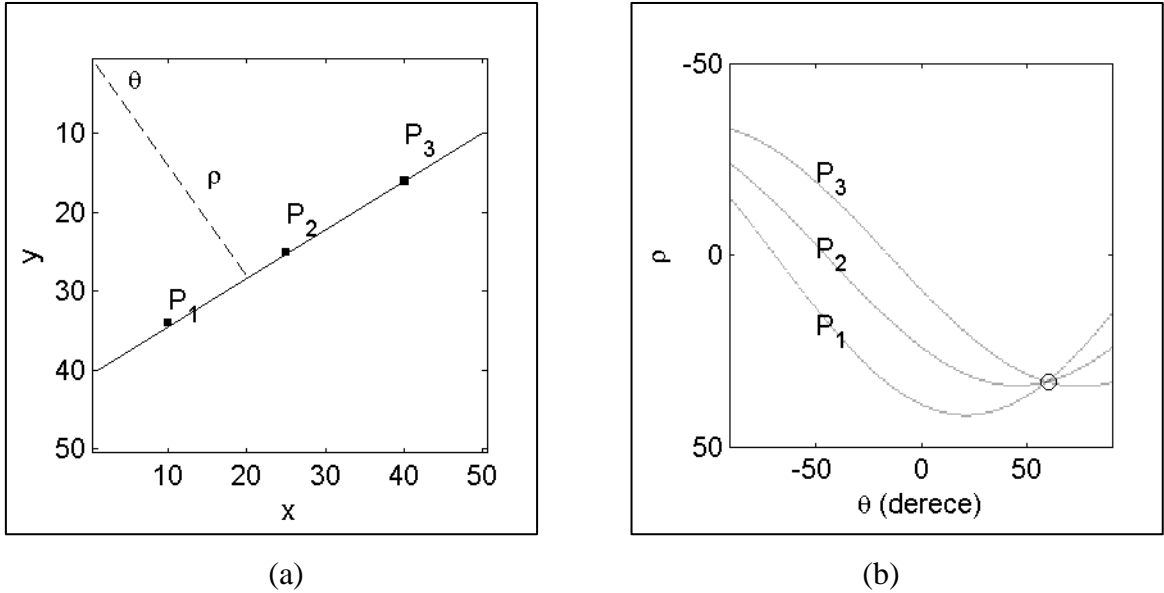
1.3.1. Düz Çizgiler İçin Hough Dönüşümü

Hough dönüşümü ile düz çizgilerin bulunması, birçok alanda başarıyla uygulanmaktadır. Hough dönüşümü genel olarak sayısal görüntülerdeki düz çizgilerin bulunması için kullanılır. Sayısal görüntülerde, görüntünün sol üst köşesi başlangıç noktası olacak şekilde sırasıyla yatay ve düşey (x_k, y_k) şeklinde iki koordinatla tanımlandığı düşünülürse; düz bir çizgi üzerine düşen herhangi bir (x_1, y_1) noktasının başlangıç noktasına göre alması beklenen değer, aşağıda verilen denklemle gösterilmektedir [6].

$$\rho = x_1 \cdot \cos \theta + y_1 \cdot \sin \theta \quad (1)$$

Burada ρ ve θ parametrelerinin olduğu yer, düz çizgiyi tanımlayan kutupsal koordinatlardır. ρ değeri sayısal görüntüdeki bir doğrunun başlangıç noktasına olan dik uzaklığını verirken θ değeri doğrunun yatayla yaptığı açıyı vermektedir. Görüntüdeki her bir parlak gözeğin konumu x_k ve y_k ikilisi ile tanımlanırsa x_k ve y_k değerleri (1) denkleminde yerine yazılarak bir ρ değeri elde edilir[6].

Her doğru için hesaplanan ρ ve θ değerleri Hough uzayında bir sinüsel eğriye karşılık gelir. Aynı doğruya ait olan tüm $P_k(x_k, y_k)$ noktaları için karşılık gelen sinüsel eğrilerin, düz çizgiye karşılık gelen aynı (ρ_k, θ_k) noktalarında kesişmektedir (Şekil 2).



Şekil 2. Doğru için Hough dönüşümü; (a) Görüntü uzayı; (b) Hough uzayı

Her bir (ρ, θ) ikilisine karşılık gelen tüm (x, y) ikili değerleri hesaplanır ve bunların gösterdiği noktanın değeri iki boyutlu bir dizide biriktirilir. Biriktirilen bu dizide dağılımın tek bir büyük tepe oluşturması beklenir. Ancak pratiktebirden çok doğrunun olması durumunda veya verinin gürültülü olması durumunda birden fazla tepe de oluşabilir [6]. Bu dağılımdaki yüksek noktalar, birçok gözeğin ρ ve θ değer çiftine katkıda bulunduğu noktayı meydana getirir. Sonuç olarak, bu tepelerin en yüksek değere sahip olanı, görüntüdeki en iyi çizgiye karşılık gelir. Birçok çizginin olması durumunda ise bir çok tepe bölgesinin oluşumu gözelenir.

1.3.2. Çemberler İçin Hough Dönüşümü

Hough dönüşümü düz çizgilerin bulunması için kullanıldığı gibi dairelerin bulunmasında da kullanılmaktadır. Hough yöntemiyle daire bulunmasının anlatılmasının nedeni, hiperbollerin bulunması işleminin hemen hemen çember bulunması ile aynı olmasındandır.

(a, b) çemberin merkezinin koordinatı ve r yarıçapı varsayılırsa, her hangi bir (x_i, y_i) noktası, (a, b, r) parametre uzayında bir dik dairesel koni yüzeyi üretmektedir. Tüm görüntü noktaları bir çember üzerine yerleştiğinde (a, b, r) noktası, koni yüzeyinde en fazla yoğunlaşmanın olduğu bölgeye karşılık gelir [18]. Bu (a, b, r) parametreleri kullanılarak bir dairenin denklemi şu şekilde ifade edilebilir,

$$(x - a)^2 + (y - b)^2 = r^2 \quad (2)$$

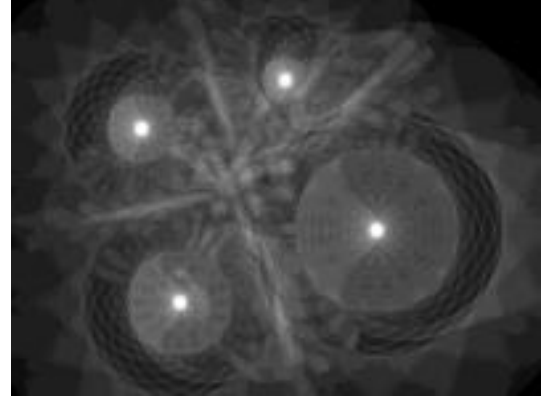
Hough uzayındaki en fazla yoğunlaşmanın olduğu alan kenar parametreleri tarafından belirlenmiş en iyi çember merkez koordinatları ve yarıçaplarına sahiptir [7]. Yarıçapı belirli olmayan bir dairenin tespiti üç boyutlu bir matris kullanılarak yapılabilir. Eğer yarıçap belirli ise iki boyutlu matris kullanılabilir. (2) denkleminde verilen eşitliğin parametrik denklemleri yazılırsa,

$$\begin{aligned} a &= x - r \cos(\theta) \\ b &= y - r \sin(\theta) \end{aligned} \quad (3)$$

eşitlikleri elde edilir. Üç boyutlu konik yüzeyini görüntülemekteki zorluk nedeniyle Hough uzayında her (a, b) noktasındaki tüm yarıçaplara karşılık gelen toplam değer en büyük değeri seçilerek Hough dönüşümü görüntülenmiş olacaktır (Şekil 3). Şekil 3 (b)'de küçük beyaz daireler çemberlerin merkezlerini vermektedir. Çemberlerin yarıçapları ise en büyük genlik değerini veren yarıçap olarak Hough dönüşümü hesaplanırken bulunur.



(a)



(b)

Şekil 3. Sayısal görüntülerde Hough dönüşümü; (a) çemberler içeren sayısal görüntü; (b) yarıçapına göre en büyük toplam değerinin (a, b) merkez ikilsine göre Hough dönüşümü

1.3.3. Hiperboller İçin Hough Dönüşümü

Hiperbolik yaylar için yöntem daha karmaşık olmamakla birlikte, uzun bir hesaplama süresi içeren 3 boyutlu bir diziyeye ihtiyaç duyulduğundan, uygulaması çember bulma yönteminde olduğu gibi uzun zaman alan bir işlemdir. Yukarıda anlatılan konu Hough dönüşümüyle çember parametrelerinin belirlenmesi için kullanılmıştı. Çember algılamada çemberin merkez koordinatları ve çember yarıçapından oluşan (a, b, r) parametreleri ile tanımlama yapılır. Hiperbol algılamada ise hiperbolü belirleyecek tepe noktasının koordinatı (t_0, x_0) ve bunlara eşlik edecek hiperbolün şeklini belirleyen hız (v_0) parametresinin belirlenmesi gerekmektedir. Bu hız parametresi ortamda yayılması muhtemel elektromanyetik dalga hızlarından elde edilmektedir. Ortamda yayılan elektromanyetik dalganın hızı yükseldikçe hiperbolün kanatları açılmakta, hız düştükçe hiperbolün kanatları kapanmaktadır.

Radargramda hiperbol denklemi,

$$t^2 = t_0^2 + \frac{(x - x_0)^2}{v^2} \quad (4)$$

biçiminde verilmektedir. Burada (t_0, x_0) ikilisi hiperbolün tepe noktası koordinatlarını verirken, v araştırma hızını vermektedir. Her x değeri için bir t zaman değeri hesaplanmaktadır [6]. Bu hesaplardan yola çıkılarak (t_0, x_0) noktası için en büyük toplam değeri veren hız değerine Hough dönüşümü,

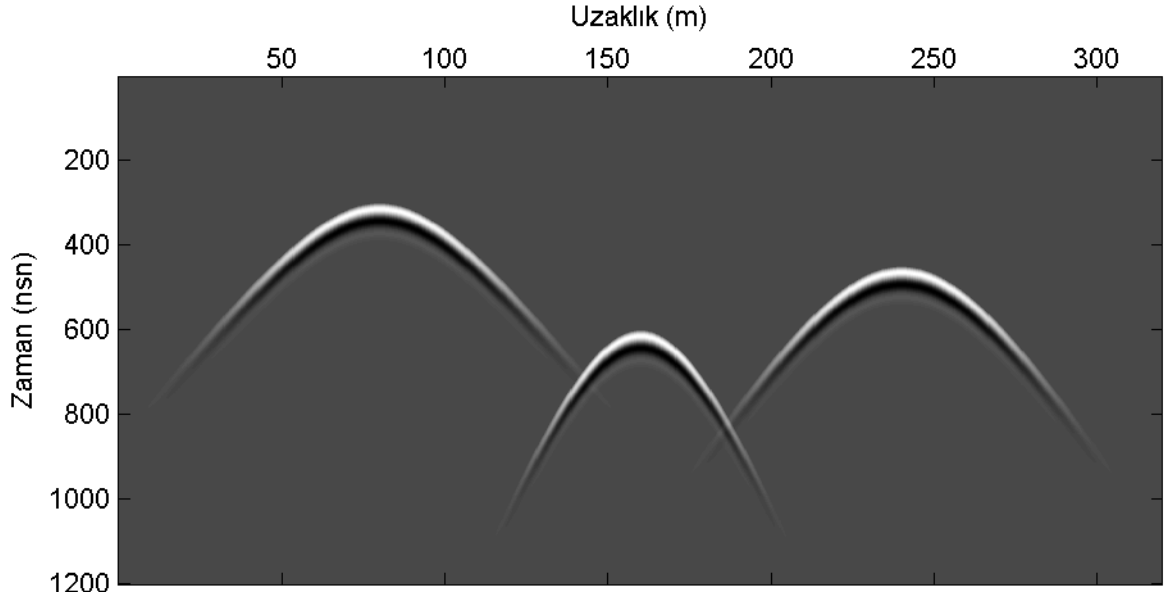
$$H(t_0, x_0, v) = \sum_{x=x_{min}}^{x_{max}} s_v(t, x), \quad (v = v_{min} : \Delta v : v_{max}) \quad (5)$$

biçiminde hesaplanmaktadır. Burada $s(*,*)$ iki boyutlu radargram verisidir. Üç boyutlu bir dizi biçiminde hesaplanan Hough dönüşümü sonucu, iki boyutta görüntülenebilmek için,

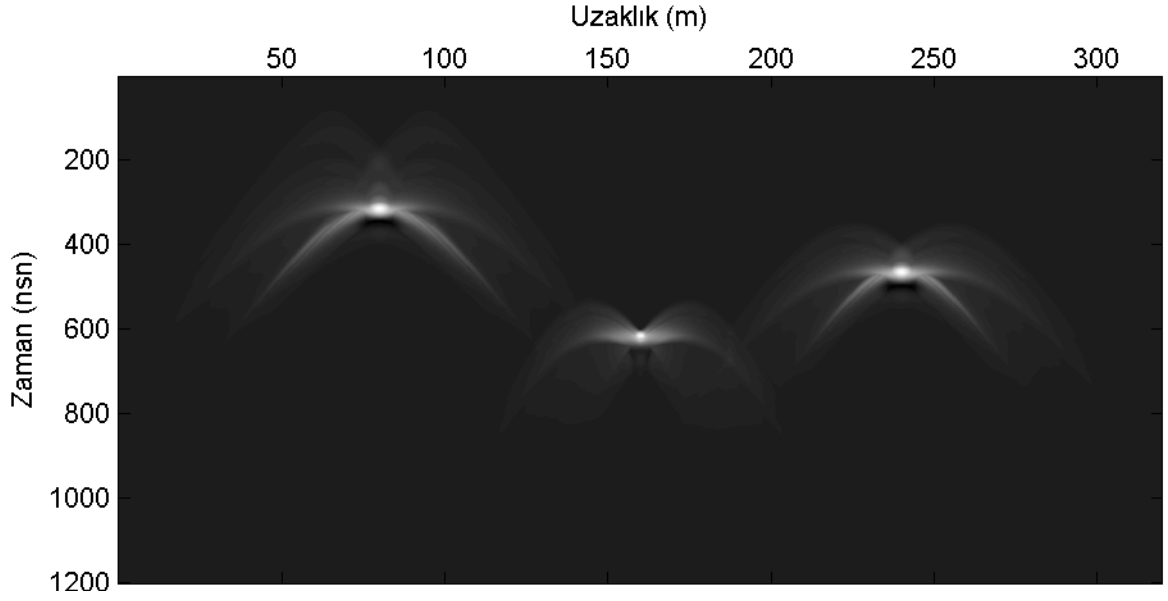
$$\hat{H}(t_0, x_0) = \bigvee_v H(t_0, x_0, v) \quad (6)$$

eşitliği kullanılarak $H(t_0, x_0, v)$ dizisinin hız yönündeki en büyük değeri alınarak iki boyutlu $\hat{H}(t_0, x_0)$ bir diziye dönüşmektedir. Aynı zamanda noktadaki en büyük dönüşüm değerini veren hız bilgisi de saklanmaktadır.

Şekil 4'te üç hiperbolden oluşan bir yapay radargram görüntüsü verilmektedir. Bu görüntü verisinde beyaz renkli gözetler pozitif değerlere karşılık gelirken, siyah renkli gözetler negatif değerlere karşılık gelmektedir. Bu görüntüye hiperbolik Hough dönüşümü uygulanırsa Şekil 5'teki görüntü elde edilir. Bu görüntüde beyaz topaklanmanın bulunduğu yerler hiperbollerin tepe noktasını vermektedir. Gerçekte hiperbolik Hough dönüşümü merkez koordinatları ve bu koordinatlardaki tüm hızlara göre hesaplanan hough dönüşümü üç boyutlu bir dizi vermektedir. Üç boyutlu dizinin gösterimindeki zorluktan dolayı dizinin hız vektörüne göre en yüksek genlik değerini veren değere göre çizimi Şekil 5'te verilmiştir.



Şekil 4. Yapay radargram görüntüsü



Şekil 5. Yapay hiperbollerin tepe noktasının hough dönüşümüyle bulunması.

1.4. Optimizasyon

Mühendisler ve bilim insanları karmaşıklık düzeyleri her geçen gün artan problemlerle karşılaşır. Bu problemler, yöneylem araştırmaları, mekanik sistemlerin

tasarımı, görüntü işleme uygulamaları, endüstriyel verimlilik gibi farklı alanlarda ortaya çıkmaktadır. Bu alanların temel problemi en iyi parametrelerin kestirilmesindeki zorluklardır. Bilinmeyen bu parametre değerlerinin bulunmasını içeren tüm bu problemler, bir optimizasyon problemi olarak tanımlanabilir. Optimizasyon problemlerinde bir yada birkaç amaç fonksiyonu tanımlanabilir. Aynı zamanda bütün parametreler göz önüne alınarak bu amaç fonksiyonu minimize ya da maksimize edilmeye çalışılır. Optimizasyon süreci değişken türüne bağlı olarak ayrık ve sürekli optimizasyon olmak üzere ikiye ayrılır [8].

Optimizasyon, en iyi sonucun bulunması için yürütülen bir süreçtir. Bir bilim insanının veya bir mühendisin önerdiği fikir, düşünce ya da plan, optimizasyon süreci sonucunda daha düzenli bir hale gelmiş olur [9]. Optimizasyon sürecinde başlangıç koşulları farklı yöntemlerle değerlendirilir. Elde edilen bu veriler, söz konusu fikir, plan ya da düşüncenin iyileştirilmesi için kullanılır. Optimizasyon farklı problemlerin nasıl çözüme kavuşturulduğunun bulunması için kullanılan matematiksel bir araçtır [10-11].

Optimizasyonda, herhangi bir problemin çözümünde birbirinden farklı iyi değerler arasında en iyisini bulmayı amaçlar. Problemin değerlendirilmesinde en iyi cevabın bulunması, kullanılan yöntem ve hata sınır değerine bağlıdır. Bu nedenle problemin formülasyonu en iyi cevabın bulunmasına doğrudan etkilemektedir.

1.4.1.Sürü Zekası

Sürü zekası, problemlere çözüm getirirken termitler, arılar, karıncalar, kuşlar, balık sürüleri gibi davranışlarını içinde yaşadıkları topluluğa göre ayarlayan hayvanların bu tutumlarını örnek alan bir yapay zeka tekniğidir. Arı kolonilerinin kovan etrafında dolaşarak birbirlerine bilgi aktarımları, karıncaların geçtikleri yollara kimyasal madde bırakarak diğer karıncalara bilgi aktarımları, kuş sürülerinin ve balık sürülerinin konum ve hızlarını ayarlayarak ilerlemeleri bu hayvanların topluluk bilincine göre yaptıkları davranışlara verilecek en iyi örneklerdendir [12].

Bir sürünün başlıca iki işlevi vardır. Bunlardan biri kendi başına organize olabilme (self-organization), diğeri ise iş bölümüdür. Kendi başına organize olmada kasıt, sürüdeki temel birimlerin, diğer birimlerle olan etkileşimleri sonucu aldıkları bilgilerle kendi başlarına işlev görmeleri ve sonucunda sistemin bütünü etkilemeleridir. Sistemin diğer birimleri ile etkileşimlerinde söz konusu olan, sistemin bir bütün olarak genel başarımı ile

ilgili bilgiler değil, temel komşuluk bilgileridir. Bonabeau ve arkadaşları kendi başına organize olabilmeyi dört özellik ile karakterize etmişlerdir [12-13]. Bunlar; pozitif geri besleme (positive feedback), negatif geri besleme (negative feedback), salınımlar (fluctuation) ve çoklu etkileşimdir. Pozitif geri besleme daha uygun yapıların bulunmasını sağlar. Karıncaların geçtikleri noktalara kimyasal maddeler bırakması, arıların daha zengin nektar kaynakları hakkındaki bilgileri diğer arılara iletmek için dans etmeleri pozitif geri beslemeye örnek olarak gösterilebilir. Negatif geri beslemede esas olan toplanan bilgilerin daha kararlı hale getirilmeye çalışılmasıdır. Salınımlar yeni kaynak keşiflerinin yapılabilmesi için yapılan rastgele dolaşımalar olarak tanımlanabilir. Çoklu iletişim ise bir bireyin diğer bireye ait bilgiyi kullanabilmesini ifade etmektedir. İş bölümü, topluluktaki her bireyin aynı anda farklı işler yapmasıdır. Bu şekilde çalışılması, tabi olmayan bireylerin gösterdikleri performanstan daha üstün bir performans ortaya konmasını sağlar ve bu özellik araştırma uzayındaki değişimlere cevap verilmesini sağlar [12].

Literatürde yapay arı kolonisi algoritmasıyla birlikte sürü zekasındaki bu özellikler üzerinde çalışan ve arıların davranışlarını model alan farklı yöntemler bulunsa da bunlardan yalnızca Yang'ın geliştirdiği Virtual Bee Algoritması[14] ayrık problemlerin çözümünde kullanılmaktadır. Bu problem yalnızca problemlerin iki boyutlu olduğu durumlarda kullanılmaktadır. [14-15]. Yapay Arı Kolonisi Algoritması Karaboğa tarafından geliştirilmiş [16] ve literatürde var olan algoritmalarla kıyaslandığında oldukça iyi sonuçlar üretmiştir. Yapay Arı Kolonisi Algoritmasının sinyal işlemede sıkça kullanılan yapay sinir ağlarının eğitiminde gösterdiği performans analiz edilmiş ve bu performans Farksal Gelişim (Differential Evolution, DE) ve Parçacık Sürüsü Optimizasyon (Particle Swarm Optimizasyon, PSO) algoritmalarıyla kıyaslanmıştır [16-18].

Bundan sonraki bölümlerde yaygın bir sürü optimizasyon algoritması olan Parçacık Sürü Optimizasyonu incelenecek ve bu çalışmaya temel teşkil eden Yapay Arı Koloni algoritması hakkında daha detaylı bilgi verilecektir.

1.4.2. Parçacık Sürü Optimizasyon

Parçacık sistemi bir noktalar dizisi oluşturur ve her bir noktaya önceden belirlenmiş başlangıç değerleri atanır. Aynı zamanda her parçacığa bir hız vektörü tanımlanır. Hız vektörü rastgele değerler alan değişkenlere göre belirlenir. Her parçacık kendisi için belirlenmiş hız vektörüne göre belirli bir açıyla hareket ettirilir. Reynolds, sürü algoritmasına temel olarak Reeves'in tanımladığı parçacık sistemini kullanmış ve bu

algoritmaya oryantasyon ve iç-cisim iletişimini dahil etmiştir. Reynolds'ın eklediği bu davranışlar kuş benzeri cisimlerin basit sürü kurallarını takip etmesine olanak sunmuştur. Basit sürü kurallarına göre bireyler hızlarını birbirine göre ayarlar ve birbirleriyle olan mesafelerini kontrol ederek birbirlerine çarpmamaya çalışırlar. Aynı zamanda birbirlerine yakın kalmaya çalışırlar. Bu davranışları bireysel zekaları artırarak daha iyi sonuç almalarını sağlar [8].

1.4.3. PSO Üzerinde Çalışmış Bazı Çok Tepeli Optimizasyon Yöntemleri

Li [19-20] genetik algoritmasından ilham alıp bir öneri sunmuş ve daha sonrasında Iwamatsu [19, 21-22] bu öneriyi geliştirmiştir. Bu yöntem esasen Jiang-Ping Li [19-23] tarafından önerilen bir PSO uyarlamasıdır. Bir parçacığın kesin bir türe ait olup olmadığını belirlemek için “yarıçap” adı verilen bir parametre değerinin atanmasını gerektirir. Bu değer için uygun seçimi iyi sonuçların ve çoğu durumda, optimize edilen amaç fonksiyona dayanan bu parametre için optimal değerin elde edilmesi için önemlidir. Kapsama (bir gruba üye seçimi işlemi) kullanımı da PSO için önerilmiştir [19, 24]. Brits [25] bir kapsama yarıçapının alt sürüdeki parçacıklar arasındaki ortalama uzaklıkların hesaplanmasıyla bulunduğu bir kapsamalı PSO algoritması önermiştir. Bu yaklaşım çeşitli parametrelerin kestirilmesini gerektirir. Bir alt sürü oluşturmak için, algoritmanın her tekrarlanmasında her bir parçacığın amaç fonksiyonunun varyansı ölçülmelidir. Varyansı hesaplamak için, her ϵ tekrarı süresince parçacıkların amaç fonksiyonları gözlemlenir ve her amaç fonksiyonu değişimi bir δ eşik değerinden düşükse bir alt sürü oluşturulmuş olur. Yaklaşım ayrıca iki alt sürünün çakışmasını engellemek için kullanıcı tarafından belirlenmiş bir μ eşik değerini de içermektedir [19, 25].

Bird ve arkadaşları [26], birden çok parametrenin atanmasından kaçınmak için uyumlu bir yöntem sunmuşlardır. Bu durumda, yarıçap da bir sürünün parçacıkları arasındaki uzaklıkların ortalaması olarak hesaplanmış olur, ama onlar yarıçaptan daha küçük bir uzaklıkta bulunan parçacıkların bilgilerini toplarlar. Eğer iki parçacık, tekrarlanmanın ϵ sayısı için birbirine yakınsa (hesaplanan yarıçapa göre), bir alt sürü oluşturulur. Bu durumda, her alt sürü için izin veren bir m parçacık limiti de mevcuttur [19, 26].

Passaro ve Sarita [27], Kennedy'nin [28] çalışmalarını takip etmişler ve ana sürünün alt sürülere ayrılması için kümeleme işlemini kullanmışlardır. Küme sayısı bulunurken

daha önceden belirlenmiş bir sayı miktarınca küme bulunmasından kaçınmak için Pelleg ve arkadaşlarının [29] algoritmasından esinlenerek k-ortalama yöntemi uygulanır ve optimal bir k sayısı belirlenir. k-ortalama algoritması optimal değerin belirlenmesi için verilen aralıktaki her bir k değeri için istatistiksel bir değer hesapladığından, bir minimum ve maksimum k sayısı atanmalıdır.

L_i , birden çok optimum değerin yerleştirilmesinde kullanılabilir ve herhangi bir ek parametre gerektirmeyecek bir PSO yönteminin geliştirilmesi için bir girişimde bulunmuştur [30]. Bu çalışma, Peram ve arkadaşlarının [31] hızın yenilenmesi için denkleme yeni bir bileşenin eklenmesi önerisinde bulunmuştur. Bu bileşen parçacığın tekrardaki x_i konumu ve her bir parçacık için hesaplanan p_i konumu arasındaki farkı içerir. p_i vektörünün her bir koordinatı ayrı ayrı hesaplanır. Bu çalışmada her bir koordinatının ayrı ayrı hesaplanması yerine sürüdeki parçacıkların p vektörleri seçilmiştir. Bu seçim karşılaştırılan iki parçacığın amaç fonksiyon değerleri arasındaki fark ile aralarındaki uzaklıkların farkın oranının hesaplanması işlemine dayanır.

$$F = \frac{f_i - f_j}{\|p_i - p_j\|} \quad (7)$$

Yani, i parçacığının p_i konumunu, hesaplanan oranı maksimize eden j parçacığının p_j vektörüdür. Elde edilen p_j konumu en iyi genel konumunun yerini alır ve böylece hız yenilenir. Bu düzenlemelerle birlikte, yöntem daha iyi bir şekilde çalışır, fakat bazı problemlerde sonuç dengesizdir.

Julio Barrera ve Carlos A. Coello elektrostatik etkileşime dayalı çok tepeli optimizasyon için parçacık sürü optimizasyon yöntemi üzerine çalışmışlardır [20]. Onların yönteminde, yukarıda anlatılan yöntemlerdekine benzer yaklaşımlar izlenir. Bir parçacığın hızı yenilediğine, sürüdeki her bir parçacık için bir değer hesaplanarak ve maksimize edilerek parçacığın g konumu belirlenir. Ancak, bu durumda, Coulomb'un kanununa göre elektrostatikten ilham alınarak, yüklü iki parçacık arasındaki kuvvet (8) denklemi kullanılarak hesaplanabilir [18].

$$F = \frac{1}{4\pi\epsilon_0} \frac{Q_1 Q_2}{R^2} \quad (8)$$

Burada F iki parçacık arasındaki kuvvet, Q_1 ve Q_2 parçacıkların yükleri, ϵ_0 elektrik sabiti (vakum geçirgenliği), R ise iki parçacık arasındaki uzaklığı göstermektedir. Önerilen yöntemde $f(p)$ parçacığı tarafından o ana kadar kaydedilmiş en iyi amaç fonksiyonunun bir parçacığın Q yükünün yerini almasıyla, yenilenen i parçacığı ve sürü içerisindeki diğer parçacıklar arasındaki bir F_{ij} kuvveti hesaplanır.

Ayrıca F denkleminde belirtilen $1/(4\pi\epsilon_0)$ değerinin yerine (9)'daki gibi hesaplanan α değeri koyulur.

$$\alpha = \frac{\|s\|}{Q_g - Q_w} \quad (9)$$

Burada $\|s\|$ araştırma alanının boyutu, Q_g en iyi genel optimum değeri, ve Q_w ise en kötü genel optimum değeridir. Yani i ve j parçacıkları arasındaki F_{ij} kuvvetinin değerini hesaplamak için, (4) denklemini kullanılır[18].

$$F = \alpha \frac{f(p_i)f(p_j)}{\|p_i - p_j\|^2} \quad (10)$$

Sayısal hataların önlenmesi için i ve j parçacıkları arasındaki uzaklık sıfır ise, F_{ij} değeri hesaplanmaz ve j parçacığı seçim için hesaba katılmaz. j parçacığı için F_{ij} 'nin maksimum değeri ile kaydedilen en iyi konum, mevcut genel amaç fonksiyonunun değerini alır.

1.5. Yapay Arı Koloni Optimizasyon

Yapay arı kolonisi (ABC) algoritması, Karaboğa (2005) tarafından arıların doğadaki davranışları temel alınarak geliştirilen popülasyon tabanlı bir optimizasyon tekniğidir. Öncelikle sayısal optimizasyon problemlerinde kullanılan algoritma, son zamanlarda çeşitli mühendislik problemlerine de başarıyla uygulanmıştır.

1.5.1. Gerçek Arıların Yiyecek Arama Davranışları

Doğal bir arı kolonisinde her bir işin yapılması için o iş için uzmanlaşmış arılar bulunmaktadır. Arılar yapacakları işlere göre bir iş bölümü oluştururlar ve bunu yaparken herhangi bir dış otoriteye gereksinim duymazlar. Bu yüzden kendi kendilerine organize olabildikleri söylenebilir. Sürü zekasının iki önemli özelliği de bu bahsedilen iş bölümü ve kendi kendine organize olabilmektir. Minimal yiyecek arama modelinde yiyecek kaynakları, işçi arılar ve gezen işçi arılar olmak üzere üç çeşit öge bulunmaktadır.

Bu ögeler devam eden bölümlerde ayrıntılı biçimde açıklanmıştır.

1.5.2. Yiyecek Kaynakları

Arıların polen, nektar veya bal gibi maddeler bulmak için gittikleri kaynaklarıdır. Bir yiyecek kaynağının kalitesini genel olarak yuvaya yakınlığı, içerdiği madde miktarı ve bu maddenin konsantrasyonu belirler. Yiyecek kaynağının kalitesini belirleyen bu ve bunlar gibi birçok faktör olsa da basit olarak kaynağının kalitesini belirleyen o kaynağın zenginliğidir [8].

1.5.3. İşçi Arılar

Arılar yiyecek çıkarılacak bir kaynak belirler ve yiyeceğin bu kaynaklardan yuvalara aktarımı işçi arıların görevidir. İşçi arıların bir diğer görevi ise gidip geldikleri bu kaynağın kalitesi ve konumuyla ilgili bilgileri diğer arılara iletmektir [8].

1.5.4. Gezen İşçi Arılar

Bu arılar sürekli nektar toplanabilecek kaynak ararlar. İki çeşit gezen arı vardır. Bunlar kaşif arılar ve gözcü arılar. Kaşif arılar rastgele olarak kaynak ararlar. Gözcü arılar ise işçi arıları izleyerek bu arılar tarafından paylaşılan bilgiyi kullanır ve yeni bir kaynağa yönelirler [8].

Arılar arasında bilgi paylaşımı kolektif bilginin oluşumu için önemli bir faktördür. Bir kovanda en önemli ortak alanlardan biri yiyecek kaynağının kalitesi ve yeri ile ilgili bilginin paylaşımının gerçekleştiği dans alanıdır. Bilgi aktarımı diğer arıların dans eden arıya antenlerini dokundurmaları ile gerçekleşir [8].

Aktarılan bilgiye göre yapılan dans da farklılık gösterir. Bunlardan en önemlisi nektarın tatlılığıdır. Dans eşik değerini bu üstünlük belirler. Dansı etkileyen diğer faktörler,

kaynağın kovana yakınlığı, çiçek kenarının kıvamı, besinin genel durumu ve hava koşullarıdır.

Yiyecek getiricilerin uzaktaki bir kaynağa yönlendirilmesi için diğer yiyecek getiricilerden yön bilgisi alması gerekmektedir. Arılar hedefe ulaşırken güneşten yararlanır ve bileşik göz yapıları sayesinde güneşle aralarındaki açıyı hesaplayabilmektedirler. Güneşin önünün kapanmış olması arıların bu hesaplamayı yapmalarına engel olmaz. Polarize güneş ışığından da güneşin konumunu belirleyebilirler. Arılar bir noktanın uzaklığını hesaplarken enerji tüketimini hesap ederler ve yiyeceği taşıırken, yüklerine bağlı olarak uçuş yüksekliklerini ayarlarlar. Bu, tükettikleri enerjiyi ayarlamalarını sağlar.

Danslar kaynağın kovana olan uzaklığına göre de farklılık gösterir. Bunlar; dairesel dans (round dance), kuyruk dansı (waggle dance) ve titreme dansı (tremble dance) gibi birkaç çeşittir. Dairesel dans ve kuyruk dansı birbirinden farklı uzaklıktaki kaynakları ayırmakta kullanılır [8].

Dairesel dans kovana maksimum 50-100 metre uzaklıktaki yiyecek kaynakları için yapıldığından, bu dans yön ve uzaklık bilgisi tam olarak vermemektedir.

Titreme dansında arılar petek üzerinde rastgele tarzda, yavaş ve düzensiz bir tempoda bacaklarını ileri, geri, sağa ve sola hareket ettirirler. Bu dans yalnızca dans alanında değil, her yerde yapılabilmektedir. Amacı, kovan kapasitesi ile kovana getirilecek yiyecek arasındaki dengeyi sağlamaktır [8].

Kuyruk dansı 100 metreden 10 kilometreye kadar olan uzaklıklardaki yiyecek kaynaklarının bilgi aktarımı için yapılır. Bu dans 8 figürüne benzeyen hareketler yapılmaktadır. Bu dansı yapan arı, kendisini izleyen arılar bu dansa eşlik etmeye başladığında dansı bırakır. Bu titreşim hareketinin 15 saniyedeki tekrarlanma sayısı ile kaynağın uzaklığı ilişkilidir. Daha az tekrarlanma, daha uzak mesafeleri ifade etmektedir.

Tüm zengin kaynaklarla ilgili bilgiler gözcü arılara iletilmiş olur ve gözcü arılar tüm bu bilgileri topladıktan sonra hangi kaynağa yöneleceğine karar verir. Zengin kaynaklarla ilgili daha çok bilgi aktarımı olur. Dolayısıyla bu kaynakların seçilme olasılığı daha yüksektir [8].

1.5.5. Yapay Arı Kolonisi Algoritması

Yapay arı kolonisi (ABC) algoritmasında; işçi arılar, gözcü arılar ve kaşif (scout) arılar olmak üzere üç grup arı bulunmaktadır. Modelin yarısını işçi, diğer yarısını da gözcü

arılar oluşturmaktadır. Nektar kaynağı sayısınınca işçi arı bulunmaktadır. Her bir nektar kaynağı için yalnızca bir işçi arı vardır [8].

Her çevrimin üç aşaması vardır; işçi ve gözcü arıların kaynaklara gönderilmesi, gidilen kaynakların nektar miktarlarının hesaplanması ve kaşif arının belirlenerek yeni bir kaynağa rastgele konumlanması. Burada yiyecek kaynakları, optimize edilmesi için üzerinde çalışılan problemin olası çözümlerine karşılık gelmektedir. Bir kaynağın nektar miktarı ise söz konusu kaynağın ifade ettiği çözümün kalite değerini temsil etmektedir. Gözcü arılar gidecekleri kaynağı belirlerken rulet tekerleği ilkesini temel alırlar. Her kolonide rastgele araştırma yapan, yiyecek ararken de herhangi bir önbilgi kullanmayan kaşif arılar bulunmaktadır. Bu nedenle arama maliyetleri düşüktür. Bu arıların buldukları nektar kaynakları da düşük olmakla birlikte zengin nektar kaynakları bulmaları da olasıdır. ABC algoritmasında “limit” parametresine göre işçi arılardan biri seçilir ve kâşif arı haline getirilir. Limit parametresiyle belirlenmiş sayıdaki denemeye çözüm geliştirilememişse söz konusu kaynak terk edilir ve bu kaynağa gidip gelen işçi arı kâşif arıya dönüşür [12].

Etkili bir arama sürecinde keşif (exploration) ve keşfedilenden faydalanma (exploitation) işlemleri eş zamanlı olarak yapılmalıdır. ABC algoritmasında keşif sürecinde kaşif arılar, keşfedilen kaynaklardan faydalanma işleminde ise gözcü ve işçi arılar görev almaktadır. Gerçek arılarda taşıma hızını koloninin bir kaynağı bulması ve onu kovana getirmesi belirler. Yapay arılar durumunda ise bulunan çözümün kalite değeri taşıma hızında etkin bir faktördür [12].

ABC algoritmasının temel adımları aşağıda verilmiştir [8]:

- Adım 1 : Başlangıç yiyecek kaynağı bölgelerinin üretilmesi.
- Adım 2 : İşçi arıların yiyecek kaynağı bölgelerine gönderilmesi.
- Adım 3 : Olasılıksal seçimde kullanılacak olasılık değerlerinin işçi arılardan gelen bilgiye göre hesaplanması
- Adım 4: Gözcü arıların olasılık değerlerine göre yiyecek kaynağı bölgesi seçmeleri
- Adım 5 : Bırakılacak kaynakların belirlenmesi ve bırakılması, kâşif arının yeni bir kaynak araması
- Adım 6 : Çevrim sayacı istenen çevrim sayısından küçükse Adım 2'ye git.

1.5.6. Başlangıç Yiyecek Kaynağı Bölgelerinin Üretilmesi

Arama uzayı yiyecek kaynaklarını içeren kovan çevresi olarak düşünülürse, algoritmanın çalışması rastgele yiyecek yerleri üretmeye başlaması şeklinde gerçekleşir.

Her bir parametrenin alt ve üst değerleri arasında değerler üretilir ve rastgele yer üretme süreci bu şekilde gerçekleşir.

$$x_{ij} = x_j^{min} + rand(0,1)(x_j^{max} - x_j^{min}) \quad (11)$$

Burada, SN yiyecek kaynağı sayısını, D optimize edilecek parametre sayısını ifade edecek şekilde tanımlanacak olursa, $i: 1 \dots SN$ ve $j: 1 \dots D$ olarak ifade edilebilir. x_j^{min} , j . parametrenin alt sınırını, x_j^{max} ise j . parametrenin üst sınırını ifade etmektedir. Aynı zamanda Deneme Sayaçları da sıfırlanmaktadır. Deneme Sayacı i . kaynağın geliştirilememe sayısını gösterir.

Başlangıç aşamasından sonra yiyecek kaynakları işçi arı, kâşif arı ve gözcü arı süreçlerinden geçirilir ve bu şekilde daha iyisi bulunmaya çalışılır. İstenen çevrim sayısı, kabul edilebilir bir hata değeri (ϵ) veya diğer optimizasyon algoritmaları için kullanılan bir durdurma kriteri, ABC algoritması için bir durdurma kriteri olarak kullanılabilir.

1.5.7. İşçi Arıların Yiyecek Kaynağı Bölgelerine Gönderilmesi

Daha önce de belirtildiği gibi yiyecek kaynağı sayısınca işçi arı mevcuttur. İşçi arı üzerinde çalıştığı kaynak komşuluğunda yeni bir kaynak belirleyerek bu kaynağın kalitesini değerlendirir. Yeni belirlediği kaynağın üzerinde çalıştığından daha kaliteli olması durumunda bu kaynağın kalitesini hafızasına alır. Aşağıdaki denklemle bu yeni kaynağın belirlenmesi durumu ifade edilmektedir.

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (12)$$

x_i komşuluğunda v_i kaynağının bulunması x_i ile gösterilen her bir kaynak için rastgele seçilen bir j parametresinin değiştirilmesiyle gerçekleşir. j parametresinin değiştirilmesinde x_k komşu çözümünün j . parametresi ile mevcut kaynağın j . parametresi arasındaki fark alınarak ϕ_{ij} sayısı ile ağırlıklandırılır ve mevcut kaynağın j . parametresine eklenir. Burada k sayısı $[1, SN]$ arasında bir rastgele sayı ($i \neq k$), ϕ sayısı ise $[-1, 1]$ arasında düzgün dağılımdan çekilmiş bir rastgele sayıdır. x_{ij} ve x_{kj} arasındaki farkın azalması çözümlerin birbirine daha çok benzemesini ifade eder ve bu durum x_{ij} parametresindeki değişim miktarının azalmasına yol açar. Yani optimal çözüme yaklaşılması, değişim miktarını da uyarlamalı olarak azaltmaktadır. Bu işlemle elde edilen

v_{ij} daha önceden belirlenmiş parametre sınırlarını aşmasıyla j . parametreye ait alt ve üst sınır değerlerine ötelenmesi durumu eşitlik (7) 'de ifade edilmiştir.

$$v_{ij} = \begin{cases} x_j^{min} & , v_{ij} < x_j^{min} \\ v_{ij} & , x_j^{min} \leq v_{ij} \leq x_j^{max} \\ x_j^{max} & , v_{ij} > x_j^{max} \end{cases} \quad (13)$$

Sınırlar aşılmaksızın elde edilen v_{ij} parametre vektörü yeni bir kaynağı ifade eder. Bu kaynağın kalitesi hesaplanır ve bir uygunluk değeri atanır [8].

$$u_i = \begin{cases} \frac{1}{1 + f_i} & , f_i \geq 0 \\ 1 + abs(f_i) & , f_i < 0 \end{cases} \quad (14)$$

Bu denklemde f_i ve u_i ile ifade edilen değerler çözümün maliyet ve uygunluk değerleridir. x_{ij} ve v_{ij} arasında belirlenen uygunluk değerine göre bir aç gözlü seçme işlemi uygulanır. Yeni bulunan v_{ij} değerinin daha iyi olması durumunda işçi arı eski kaynağın yerini hafızasından siler ve v_{ij} kaynağının yerini hafızasına alır. Bunu yapmaması durumunda i . arı x_i kaynağına gitmeye devam eder ve x_i çözümünün geliştirilememesinin sonucu olarak Deneme Sayacı bir artar. x_i çözümünün geliştirildiği durumda sayaç sıfırlanır.

1.5.8. Gözcü Arıların Seçiminde Kullanacakları Olasılık Değerlerinin Hesaplanması

İşçi arılar bir çevrimde araştırmalarını tamamladıktan sonra kovana döner ve bulunan kaynakla ilgili bilgileri dans aracılığıyla gözcü arılara iletirler. Gözcü arı bir kaynak seçerken bu bilgidan yararlanır ve bu bölge yiyecek kaynaklarının nektar miktarlarına orantılı bir olasılıkla seçilir. Bu durum ABC 'nin çoklu etkileşim sergilemesine örnek olarak gösterilebilir. Olasılıksal seçme işlemi yapılırken uygunluk değerleri kullanılır. Uygunluk değerine bağlı seçme işleminin gerçekleşmesi rulet tekerleği, stokastik örnekleme, turnuva yöntemi veya diğer seçim yöntemlerinden herhangi biriyle yapılabilir. Bu seçim işlemi yapılırken temel ABC algoritmasında rulet tekerleği kullanılmıştır.

Tekerlekteki her bir dilimin açısıyla uygunluk değerleri birbirine orantılıdır. Yani bir kaynağın uygunluk değeriyle diğer kaynakların tümünün toplamının birbirine oranı o kaynağın diğer kaynaklara göre nispi seçilme olasılığını vermektedir.

$$p_i = \frac{u_i}{\sum_{n=1}^{SN} u_n} \quad (15)$$

Burada uygunluk ile ifade edilen, i . kaynağın kalite değeri, SN ise işçi arı sayısıdır. Bu işleme göre kaynak bölgesini seçecek gözcü arı sayısı, uygunluk değerine bağlı olarak artacaktır. ABC algoritmasındaki bu durum, doğal arıların davranışlarında görünen pozitif geri besleme özelliğine benzemektedir.

1.5.9. Gözcü Arıların Yiyecek Kaynağı Bölgelerini Seçmesi

Algoritmada olasılık değerleri hesaplanır ve rulet tekerleğine göre seçim işleminde bulunan değerler kullanılarak $[0,1]$ aralığında rastgele bir sayı üretilir. p_i değeri bu sayıdan büyükse gözcü arı bu kaynak bölgesinde yeni bir çözüm üretir. Gözcü arı bu çözümü üretirken işçi arılar gibi eşitlik 12'yi kullanır. Yeni çözüm değerlendirilerek kalitesi belirlenir. Daha sonra açgözlü seçim işlemi uygulanır. Açgözlü seçim işleminde yeni çözümlerle eski çözümlerin uygunlukları karşılaştırılır ve iyi olan seçilir. Yeni çözümlerin daha iyi olması durumunda yeni çözüm eski çözümlerin yerini alır ve Deneme Sayacı sıfırlanır. Eski çözümlerin daha iyi olması durumunda ise mevcut çözüm korunur ve Deneme sayacı bir artırılır. Gözcü arıların tamamı yiyecek kaynağı bölgelerine dağılına kadar bu süreç devam eder.

1.5.10. Tüklenen Kaynağı Bırakma ve Kaşif Arı Üretimi

Tüm işçi ve gözcü arıların arama süreçlerini tamamlamasıyla Deneme Savaşları kontrol edilir. Çözüm geliştirememeye sayacı, bir arının gidip geldiği kaynaktan faydalanıp faydalanmadığını belirler. Çözüm geliştirememeye sayacının bir kaynak için belli bir eşik değerin üstünde olması durumu, işçi arının çözümü tüketmiş olan o kaynağı bırakıp başka bir çözüm arayışına girmesini gerektirir. Başka bir çözüm arayışına giren işçi arı, kaşif arı haline dönüşmüş olur. İşçi arının kaşif arıya dönüşmesiyle bu arı için rastgele çözüm arama

süreci başlamış olur. Kaynağın tükendiğini ifade eden eşik değeri “limit” olarak adlandırılır. Temel ABC algoritmasında her çevrim için yalnızca bir kaşif arının çıkmasına müsaade edilir.

1.5.11. ABC Algoritmasının Adımları ve Özellikleri

ABC algoritması araştırma sırasında uygulanan 4 farklı seçim işlemi aşağıdaki şekilde açıklanabilir;

- Küresel Seçim İşlemi: Eşitlik (15)'te tanımlanan olasılık değerleri hesaplanarak diğer kaynaklara kıyasla daha iyi kaynaklar elde edilmesinde kullanılır.
- Bölgesel Olasılık Tabanlı Seçim İşlemi: İşçi ve gözcü arıların nektar kaynağının türünü belirlemesine yardımcı olur yöresel bilgiyi kullanarak bir bölgeye ulaştıktan sonra o bölgedeki bir kaynağa karar verilmesine sebep olur
- Aç Gözlü Seçim İşlemi: İşçi ve gözcü arıların bu işlemi daha iyi olan kaynağın belirlenmesinde kullanır.
- Rastgele Seçim İşlemi: Eşitlik (11) ile tanımlanır ve kaşif arılar tarafından gerçekleştirilir.

Bütün bu seçimlerin bir arada kullanılması ABC algoritmasının hem iyi bir küresel araştırma, hem de bölgesel araştırma yapabilmesine olanak sağlar.

ABC algoritmasının adımları ve her bir adımda yapılan işlemler daha önceki bölümlerde belirtilmişti. Bu adımları sözde-kod şekline aşağıdaki gibi gösterebiliriz.

Eşitlik (11) aracılığıyla tüm x_{ij} , $i = 1, \dots, SN$, $j = 1, \dots, D$, çözümlerine başlangıç değerlerinin atanması ve çözüm geliştirememeye (Deneme) sayaçlarının sıfırlanması ($DenemeSayacı_i = 0$)

$f_i = f(x_i)$ maliyet fonksiyon değerinin ve bu değere karşılık gelen uygunluk değerinin (u_i) hesaplanması.

REPEAT

FOR $i=1$ to SN **DO**

Eşitlik (12)'yi kullanarak x_i çözümünün işçi arısı için yeni bir kaynak üret (v_i)
ve

$f(v_i)$ 'yi (13) eşitliğinde yerine koyarak bu çözümün uygunluk değerini hesapla.

v_i ve x_i arasında aç gözlü seçim işlemi uygula ve daha iyi olanı seç.
 x_i çözümü gelişmemişse çözüm geliştirememeye (Deneme) sayacını bir artır,
 $Deneme_i = Deneme_i + 1$,
 gelişmişse sayacı sıfırla, $Deneme_i = 0$

END

Eşitlik (14) ile gözcü arıların seçim yaparken kullanacakları uygunluk değerine dayalı olasılık değerlerini, (p_i), hesapla.

t=0

REPEAT

IF (random< p_i) **THEN**

Eşitlik (12)'yi kullanarak gözcü arı için yeni bir kaynak, v_i üret.

v_i ve x_i arasında aç gözlü seçim işlemi uygula ve daha iyi olanı seç.

x_i çözümü gelişmemişse çözüm geliştirememeye sayacını bir artır,
 $Deneme_i = Deneme_i + 1$,
 gelişmişse sayacı sıfırla, $Deneme_i = 0$

t=t+1

END IF

UNTIL (t=SN)

IF ($\max\{Deneme_i\} > limit$) **THEN**

x_i 'yi Eşitlik (11) ile üretilen rastgele bir çözümle değiştir.

END IF

En iyi çözümü hafızada tut.

UNTIL

Şekil 6. ABC sözde-kod algoritması

1.5.12. ABC Algoritmasının Temel Özellikleri

- Oldukça basit ve esnektir.
- Yiyecek aramada görevli doğal arıların davranışlarını çok yakın şekilde benzetir.
- Sürü zekasını esas alan bir algoritmadır.
- Sürekli problemler için geliştirilmiş olsa da ayrık problemlerin çözümünde de kullanılabilir.

- Oldukça az kontrol parametresi içerir.
- Kaşif arıların gerçekleştirdiği küresel araştırmaya ve gözcü arılar ile işçi arıların gerçekleştirdiği bölgesel araştırmayı içerir. Bu iki araştırmayı da dengeli bir şekilde yürütür.

Literatürde ABC algoritmasının koloni büyüklüğü için 20-50 arası bir değer, “limit” için ise parametre sayısı (D) ile koloni büyüklüğünün (SN) çarpımı civarında bir değer önerilir. Örneğin bazı çalışmalarda limit için $0,5 \times SN \times D$ değeri kullanılmıştır [6].

2. YAPILAN ÇALIŞMALAR

2.1. Giriş

Yer radarı, geniş alanlarda yüksek çözünürlük sağlayan bir arama yöntemi olarak günümüzde kullanılmaktadır. Yüksek frekanslı elektromanyetik dalgalar yardımıyla sığ yapıların araştırılması arkeolojik aramalar, su aramaları, gömülü nesnelerin aranmasında oldukça yaygın olarak kullanılmaktadır. Yöntemin sığ ve yüksek çözünürlüklü arama yapması, uzaysal ortamda sık örnekleme yapılmasını gerektirmektedir. Bu ise ölçüm zamanını artırmaktadır. Ölçüm zamanını azaltmak ve maliyeti düşürmek için, ölçüm tekniği açılım yapılarak değil, sabit bir açılım (~1m) seçilerek gerçekleştirilmektedir. Yani alıcı-verici arasında uzaklık 1m seçilerek sabitlenmektedir. Sabit açılım kullanılarak alıcı-verici çifti kaydırılarak hızlı bir ölçüm gerçekleştirilmektedir. Elde edilen veriler zaman-uzaklık görüntüsü biçiminde gösterilmektedir. Gömülü nesnelerin konumlarının tam olarak belirlenebilmesi için verilerin derinlik-uzaklık görüntüsüne dönüştürülmesi gerekir. Bunun için ortamın hız değişiminin bilmesi gerekmektedir. Hız bilgisi, sismik veri işleminde açılım yapılarak yansıma hiperbollerinden belirlenirken, yer radarında ise saçılma hiperbollerinden belirlenebilmektedir. Bu hiperbollerin tepesi sismik verilerde sıfır açılım noktasında iken yer radarında değişik noktalarda bulunmaktadır. Bu durumda Hough dönüşümü (3 boyutlu hız analizi) yardımıyla belirlenmektedir. Oysaki, yer radarı aramalarında çoğu zaman kısa zaman diliminde gömülü nesnelerin derinliklerinin bilinmesi istenebilir. Bu durumda Hough dönüşümü bu işlemi oldukça uzun bir zamanda gerçekleştirdiğinde uygulamada kullanılması zor gözükmektedir.

Bu çalışmada Hough dönüşümünün zaman açısından yetersiz kaldığı durumlarda stokastik optimizasyon yöntemlerine başvurulmuş ve sonuçlar irdelenmiştir. Yer radarı görüntülerinde birçok hiperbolün var olmasından dolayı çok tepeli bir optimizasyon algoritmasına ihtiyaç duyulmuştur. Bunun için PSO algoritması yardımıyla çok tepeli arama algoritması incelenmiş ve yetersiz kaldığı gözlenmiştir. Dolayısıyla problemin çözümü için çok tepeli yapay arı koloni algoritması geliştirilmiştir. Geliştirilen yöntem öncelikle sayısal görüntülerde çemberlerin bulunmasında kullanılmış ve ardından hem yapay hem de gerçek yer radarı verileri üzerine uygulanmış ve sonuçları irdelenmiştir.

2.2. Çok Tepeli ABC Algoritması

Bazı problemlerin yalnızca tek bir çözümü olmayıp birden çok çözüm içerebilirler. Bu tip problemlerin kaç tane çözüm içerdiğinin bulunması için çok tepeli optimizasyon yöntemi uygulanır. Bazı durumlarda yerel optimal noktalar bile küresel optimal noktalar gibi problemin çözümü olabilirler. Çok tepeli optimizasyon yöntemi uygulanırken şimdiye kadar genel olarak parçacık sürü optimizasyonu (PSO) yöntemine dayanarak yola çıkmıştır.

ABC algoritmasının özgün ve en yaygın formları ABC algoritmasında kullanılan genel seçim mekanizmasından dolayı çok tepeli optimizasyon problemlerini çözemeyip yalnızca tek bir genel noktanın bulunmasında kullanılabilir [19].

Son zamanlarda ABC algoritmasının çok tepeli optimizasyon problemlerinin çözümünde kullanılması için bir takım çalışmalar gerçekleştirilmektedir [18].

Klasik ABC algoritmasında bir arı, limit değerini (hareketsizlik sayacı) aştığında yerel çözüme düşmektedir. Yerel çözüme takılan bir arı genel çözümle karşılaştırıldıktan sonra kaşif arı tarafından başka bir konuma taşınmaktadır. Son zamanlarda geliştirilen çok tepeli ABC algoritmasında ise limit değerini aşan noktalar kaşif arılar yardımıyla yeni bir konuma taşımadan önce bir yerel çözüm listesine kayıt edilmektedir. Bu yöntemin en büyük eksikliği listenin çok fazla sayıda konumla dolması ve büyük bellek gerektirmesidir. Ayrıca şişen bu listedeki çoğu nokta birbirine yakın noktalardan oluşmaktadır. Bu durumda çözümü oluşturan noktaların seçimi imkansız duruma gelmektedir. Öte yandan listeye kayıtlı bir konuma düşen arının limit değerini aşana kadar bu konumda kalması ve arama işlemine katılmaması diğer çözümlerin bulunmasını zorlaştırmaktadır. Liste yönteminin bu sorununu çözmek için listeye kayıt edilen her konumun etrafında yarıçapı verilen bir çember konarak bu çemberin içine düşen her arı kaşif arıya dönüştürülerek yeni bir konumdan başlanması sağlanmaktadır. Bu durumda ise konacak her çemberin yarıçapının ne olacağı sorusu önemli bir sorun oluşturmaktadır [18].

Bu çalışmada, çok tepeli yeni bir ABC algoritması önerilmiştir. Bu algorithmada arama işlemi klasik ABC algoritmasında olduğu gibidir. Kaşif arı aşaması ise farklılık göstermektedir. Kaşif arı aşaması ise kendi içinde üç aşamadan oluşmaktadır.

Birincisi, her çevrimde deneme sayacı limiti aşınca, aşan yiyecek konumu bir listeye kayıt edilerek o noktadaki arı kaşif arıya dönüştürülür.

İkinci aşamada, listeye kaydedilen her konum ile tüm arıların konumları karşılaştırılır. Bu karşılaştırma için Coulomb'un kanunundan esinlenerek aşağıdaki eşitlik verilebilir.

$$F_{ij} = \frac{f(p_i) f(p_j)}{\|p_i - p_j\|^2} \quad (16)$$

Burada i . liste elemanını, j . ise karşılaştırılan arıyı vermektedir. Buradan elde edilen F_{ij} değerlerinin en büyük olanını veren j . arı kaşif arıya dönüştürülerek listedeki yerele düşmesi engellenir. Eğer $f(p_i)$ değeri $f(p_j)$ değerinden daha iyi ise liste değeri olduğu gibi kalır. Aksi durumda ise liste değeri j . konum bilgisiyle yer değiştirir.

Üçüncü aşamada listedeki tüm elemanlar birbiriyle karşılaştırılır. Bu durumda bütün elemanların elektrostatiğe dayalı F_{ij} değeri hesaplanır. Herhangi iki eleman çiftinin F_{ij} değeri bir önceki aşamada bulunan en büyük F_{max} değeri ile karşılaştırılır. Hesaplanan F_{ij} değeri karşılaştırılan F_{max} değerinden az olursa en iyi eleman hariç, diğerleri Listedenden çıkartılır.

Bu işlevler sayaç bitene kadar devam ettikten sonra cevap listesi oluşturulur. Ancak listede yer alan her eleman cevap değildir. Çok basit bir kümeleme yöntemiyle liste iki kümeye ayrılır ve bu kümelerden daha iyi değerli olanlar çözüm olarak kabul edilir. Diğerleri ise gereksiz koordinatlardan oluşur.

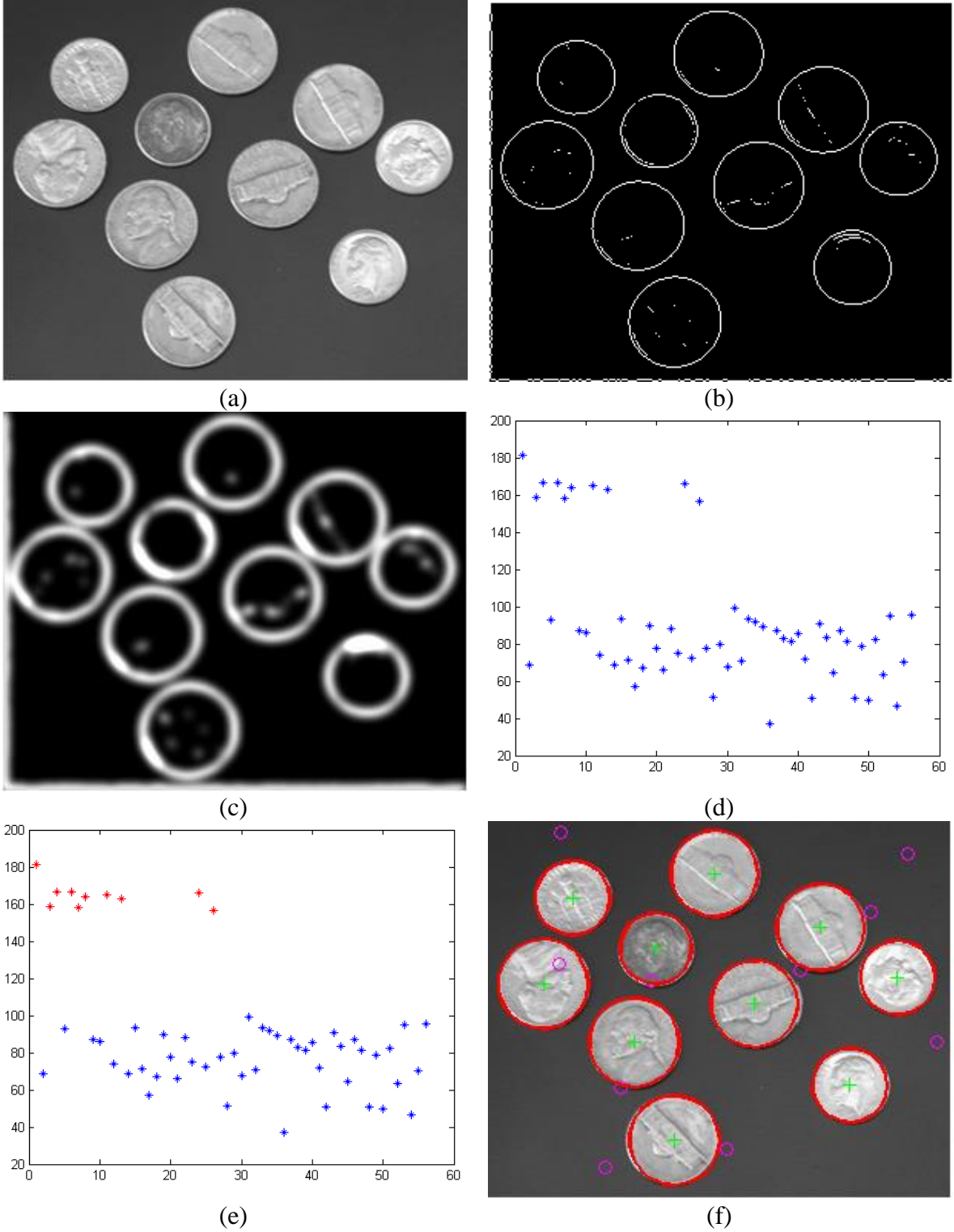
2.2.1. ABC Yöntemiyle Çember Algılama

Bu bölümde, geliştirilen algoritmanın bir problem üzerinde uygulaması verilmektedir. Bugüne kadar görüntü içerisindeki dairelerin parametrelerini elde etmek için çeşitli yöntemler geliştirilmiştir. Bunlardan en çok bilineni Hough dönüşümüdür. Hough dönüşümünün kendine özgü avantajları olduğu gibi dezavantajları da bulunmaktadır. Dezavantajlarından en önemli olanı hesaplama zamanı ve büyük bir belleğe ihtiyaç duymasıdır. Çünkü Hough dönüşümünde görüntü üzerindeki tüm gözetekler teker teker 360 derece taranır ve bu işlem oldukça uzun bir süre gerektirir. Tüm bu taramaların sonuçlarının kaydedilmesi için de büyük bir belleğe ihtiyaç vardır. Örneğin, bu dönüşüm 500×600 gözeteklik bir görüntü üzerinde uygulanırsa ve yarıçapı 10 gözetekten 300 gözeğe kadar alırsak 360 derecelik bir taramayla 31.320.000.000 gözeteklik bir hesaplamanın göze

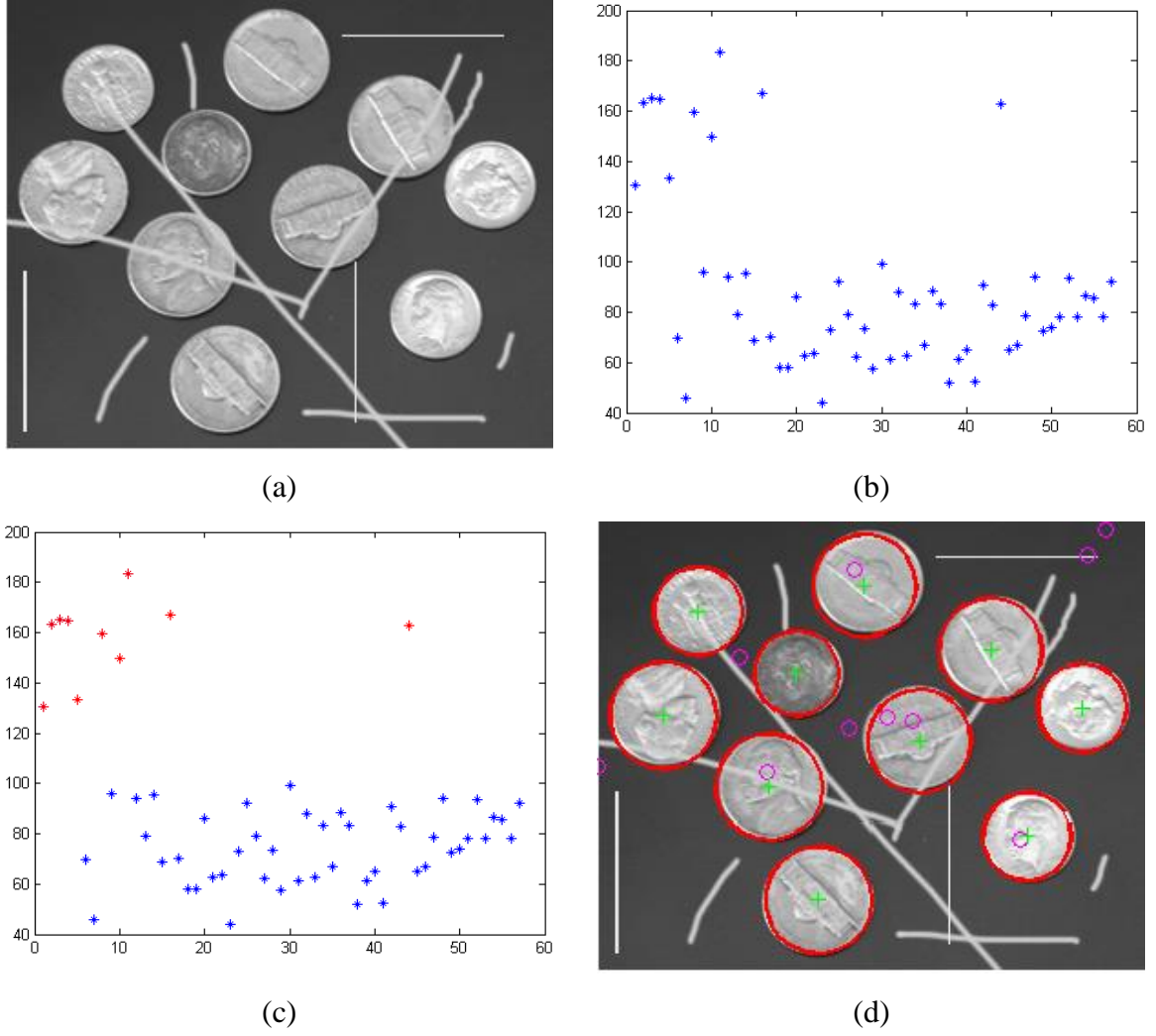
alınması gerekir. Hough dönüşüm fonksiyonu, bir uygunluk fonksiyonu olarak, geliştirilen algoritmada 3 boyutlu uzayda uygulanırsa bu üç boyut x , y ve yarıçap doğrultularından meydana gelir. x boyutu 1 ile 500, y boyutu 1 ile 600, yarıçap boyutu ise 10 ile 300 arasında tam sayı değerler olacaktır. Bu durumda 200 adımda 30 arıyla 360 derecelik bir taramanın x , y ve yarıçap doğrultuları için ayrı ayrı yapılması gerekir. Hesaplamanın tasarlanan algoritmada 3 boyutlu uzayda uygulanmasıyla, hesaplamada kullanılacak gözetki sayısı 6.480.000 'e düşürülmüş olur. Elde edilen bu sayı, Hough dönüşümünde ihtiyaç duyulanın %0,02'sidir.

Örnek olarak MATLAB yazılımındaki bir görüntü kullanılmıştır (Şekil 7(a)). Bu görüntünün ABC yönteminde başarılı olması için bazı ön işlemlere gerek duyulmaktadır. Bu işlemlere öncelikle Sobel kenar algılama yöntemi ile başlanarak Şekil 7(b)'deki kenar algılama görüntüsü elde edilir. Kenarların çıkarılmasının ardından Gaussian süzgeci uygulandıktan sonra Şekil 7(c) görüntüsü elde edilir. Elde edilen bu görüntüye algoritma uygulanarak 200 adım tamamlandıktan sonra algoritmadan çıkan değer listesi oluşmuş olur. Şekil 7(d)'de değer listedeki konumların maliyet değerlerinin diyagramı görülmektedir. Bu liste elde edildikten sonra k-ortalama kümeleme yöntemiyle, iki ayrı küme oluşturularak en iyi sonuçlar Şekil 7(e)'deki gibi elde edilir (kırmızı noktalar). Bu sonuçlardan yola çıkarak listedeki en iyi değerler belirlenmiş ve Şekil 7(f)'de başlangıç görüntü üzerine işlenerek gösterilmiştir. Bu görüntüde tüm daireleri hem konumları hem de yarıçapları iyi bir biçimde elde edildiği görülmektedir.

Algoritmayı gürültülü bir ortamda da denemek için Şekil 7(a)'da verilen görüntüye değişik eklentiler yapılarak Şekil 8(a) elde edilmiştir. Algoritma Şekil 8(a)'ya uygulandıktan sonra Şekil 8(b)'de görülen değer listesi diyagramı oluşmuş olur. Değer listesine k-ortalama kümeleme yöntemi uygulandıktan sonra en iyi değerler Şekil 8(c)'de görüldüğü gibi elde edilmiştir. En iyi değerlerin parametreleri Şekil 8(a)'da verilen görüntüye eklenirse Şekil 8(d) ortaya çıkmış olur.



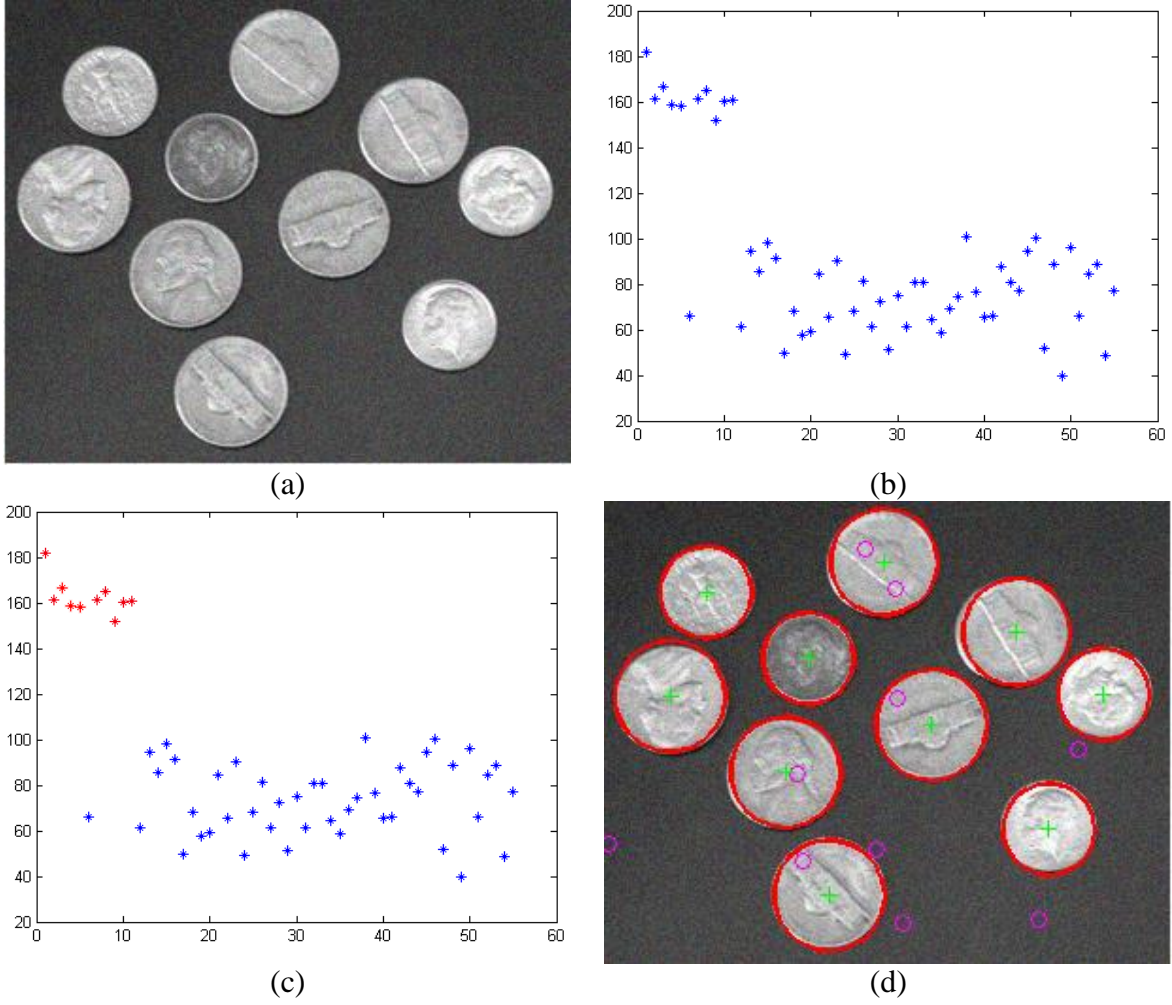
Şekil 7. Sayısal görüntülerde dairelerin bulunması (a) örnek görüntü; (b) kenar algılama yapılmış görüntü; (c) gaussian bulanıklaştırma süzgeci uygulanmış görüntü; (d) ABC algoritması uygulandığında elde edilen verilerin diyagramı; (e) kümeleme ile iki sınıfa ayrılmış diyagram; (f) bulunan en iyi çözümlerin görüntü üzerine yerleştirilmesi.



Şekil 8. Eklentili sayısal görüntülerde dairelerin bulunması (a) örnek görüntü; (b) ABC algoritması uygulandığında elde edilen verilerin diyagramı; (c) kümeleme ile iki sınıfa ayrılmış diyagram; (d) bulunan en iyi çözümlerin görüntü üzerine yerleştirilmiş görüntü.

Aynı işlemlerin Şekil 7(a)'ya tuz ve biber gürültüsü eklenmesi ve o şekilde algoritmaya sokulması halinde ortaya çıkan sonuçlar Şekil 9(a)-(d) gösterilmiştir.

Sonuçlara bakıldığında, algoritmanın gürültülü ortamda dahi iyi çalıştığı görülmektedir. Bunun başlıca nedeni görüntüdeki ayrırlılığın artmasından kaynaklanmaktadır.



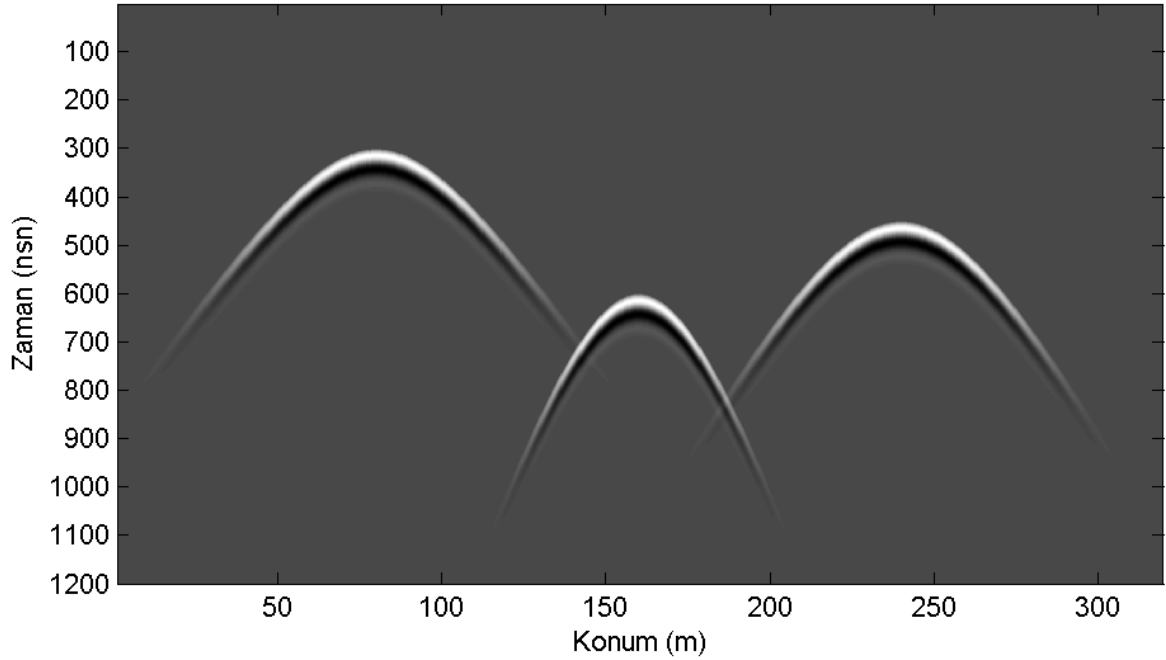
Şekil 9. Gürültülü sayısal görüntülerde dairelerin bulunması (a) örnek görüntü; (b) ABC algoritması uygulandığında elde edilen verilerin diyagramı; (c) kümeleme ile iki sınıfa ayrılmış diyagram; (d) bulunan en iyi çözümlerin görüntü üzerine yerleştirilmiş görüntü.

2.3. ABC Yöntemiyle Radargram Hiperbollerin Algılanması

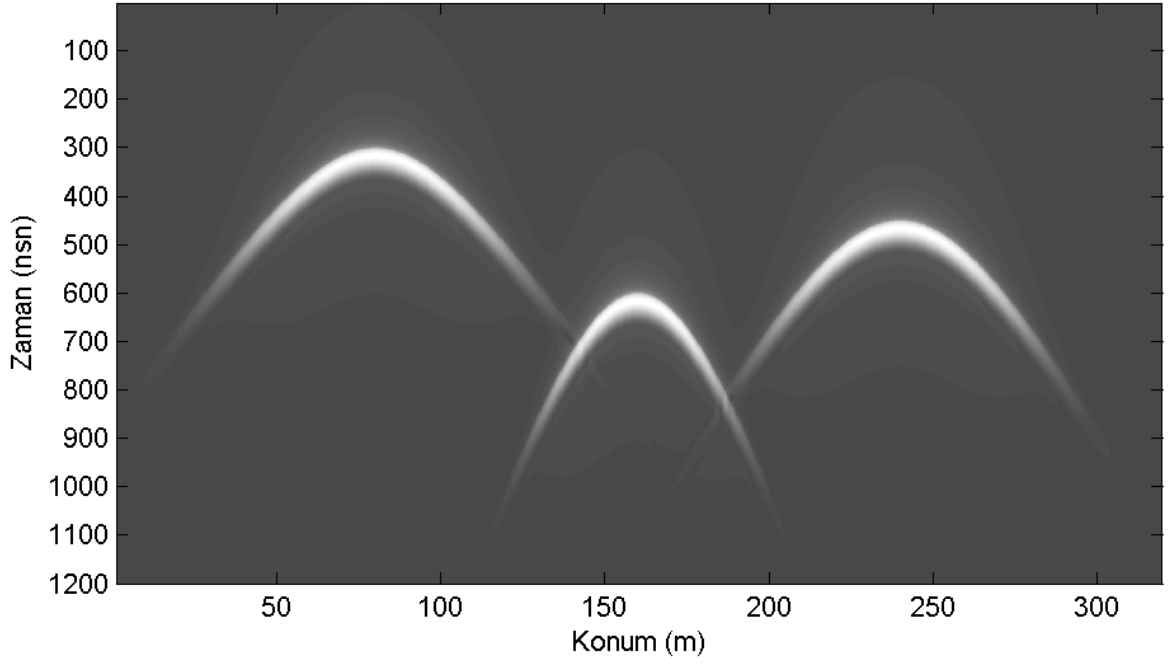
Radargram verilerindeki hiperboller, yer radarından elde edilen zaman-konum görüntülerini derinlik-konum görüntülerine çevirecek hız bilgisine saklamaktadır. Klasik Hough dönüşümü bu hiperbollerin konumlarının bulunması çok uzun zaman aldığından önerilen ABC algoritması kullanılmasının daha kısa sürede sonuç vermesi amaçlanmaktadır.

2.3.1. Yapay Radargramdaki Hiperbollerin Algılanması

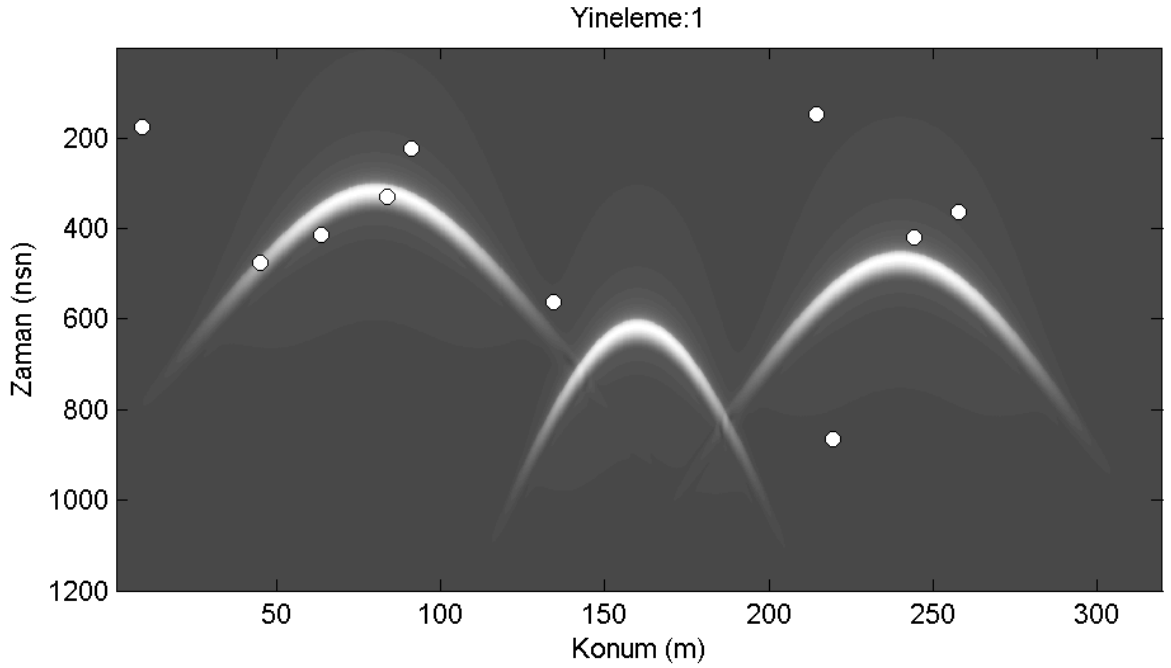
Öncelikle geliştirilen algoritmanın başarımını ölçmek için hızlarının tam olarak bilindiği yapay yer radarı verileri kullanılmıştır. Yapay verileri oluşturmak için SAMBAT yazılımı [1] kullanılmıştır. Bu yapay yer radarı görüntüsündeki hiperboller oluşturulan saçılmaların hızlar $v = \{0.1m/nsn, 0.05m/nsn, 0.08m/nsn\}$ olarak verilmiştir. Geliştirilen algoritma hiperboller için uygulanacak olursa, hiperbol bulma yöntemleri ile daire bulma yöntemleri arasında birçok farkın bulunduğu görülecektir. Bu kez uygunluk fonksiyonunda yine Hough dönüşümünden yararlanmak üzere, daire denklemi yerine hiperbol denklemi kullanılır. Örnek olarak Şekil 10'da verilen yer radarı görüntüsüne kenar algılama, bulanıklaştırma işlemleri yerine radar görüntüsünün iyi sonuç vermesi için Hilbert dönüşümü [1] uygulanmıştır (Şekil 11). Sonraki adım elde edilen bulanıklaştırılmış görüntünün yazılan algoritmada uygulanmasıdır. Bu işlem uygulanmasında 1. Çevrim sonucunda yiyecek bölgelerinin konumları Şekil 12'de verilmiştir. Şekil 13'de 34. çevrimin sonucu verilmiştir. Bu çevrimde limit değeri aşan yiyecek bölgelerinin verdiği sonuçlar görüntü üzerine işlenmiştir. Bu görüntüye göre iki hiperbol tam olarak çakışma sağlanırken ortadaki hiperbolün konumu yerel çözüme takılmıştır. Şekil 14'te 71. çevrimin sonucu görülmektedir. Bu sonuca göre hiperbollerin tam olarak çakışma gösterdiği görülmektedir. Tüm çevrimleri sonucunda da sonucun değişmediği görülmüştür.



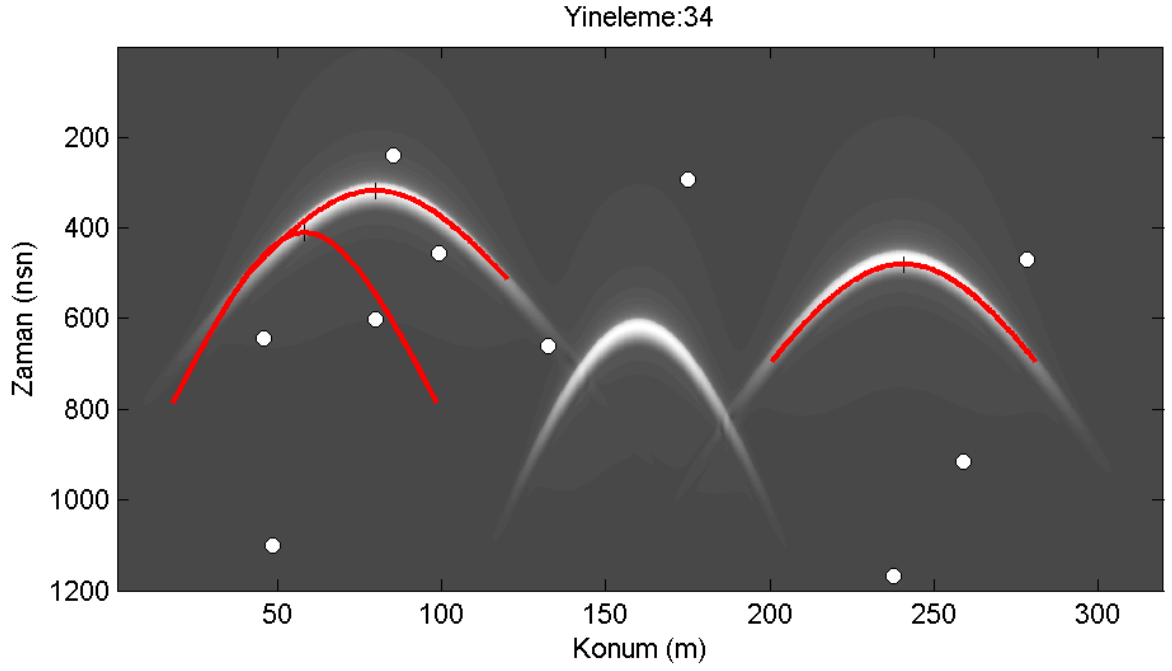
Şekil 10. Yapay yer radarı görüntüsü.



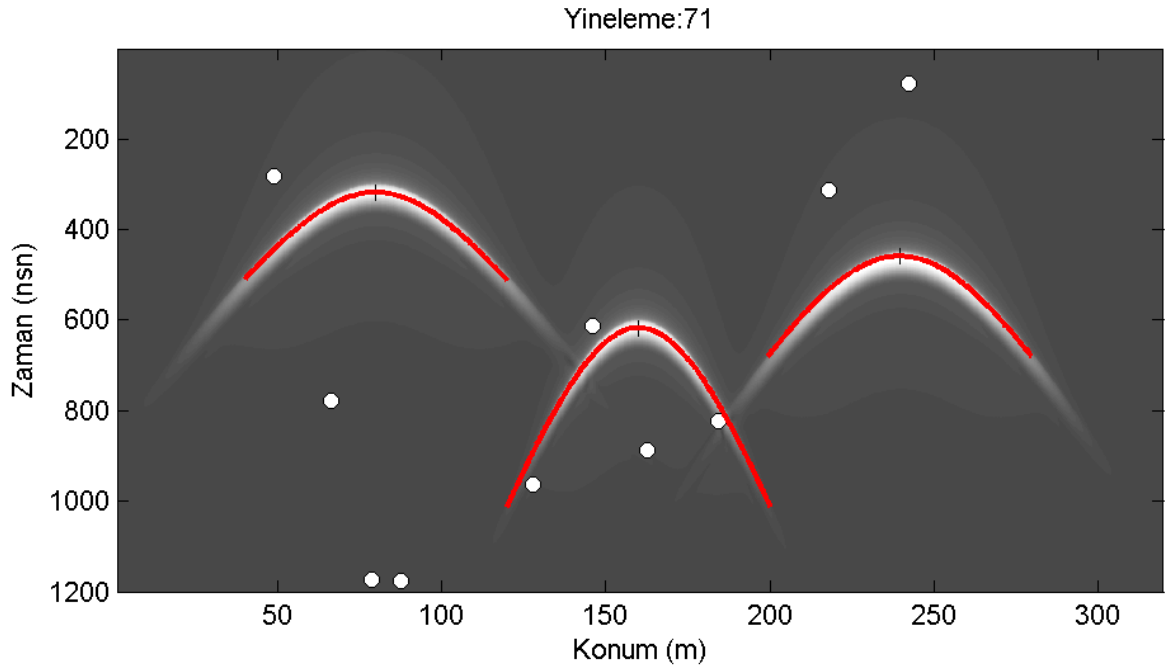
Şekil 11. Yapay yer radarı verilerine Hilbert dönüşümü uygulandıktan sonraki görüntüsü.



Şekil 12. Önerilen algoritmanın 1. çevrimdeki yiyecek bölgeleri yuvarlak noktalar biçiminde gösterimi.

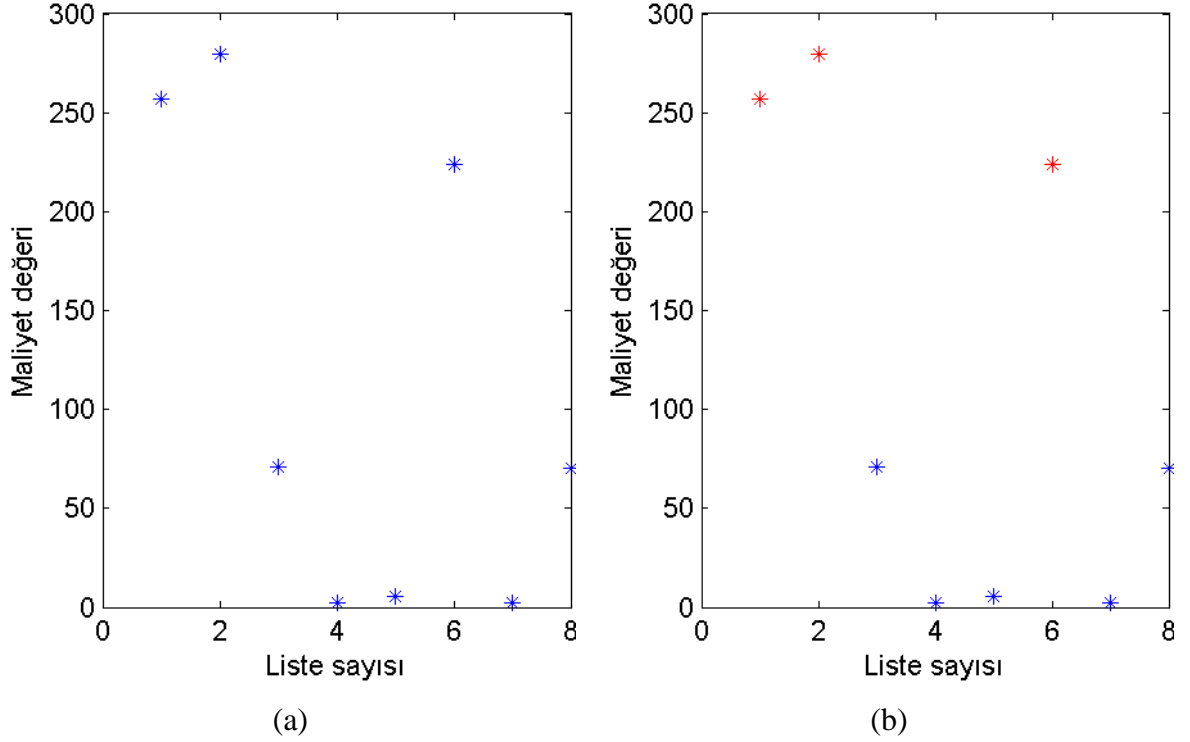


Şekil 13. Önerilen algoritmanın 34. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.



Şekil 14. Önerilen algoritmanın 71. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.

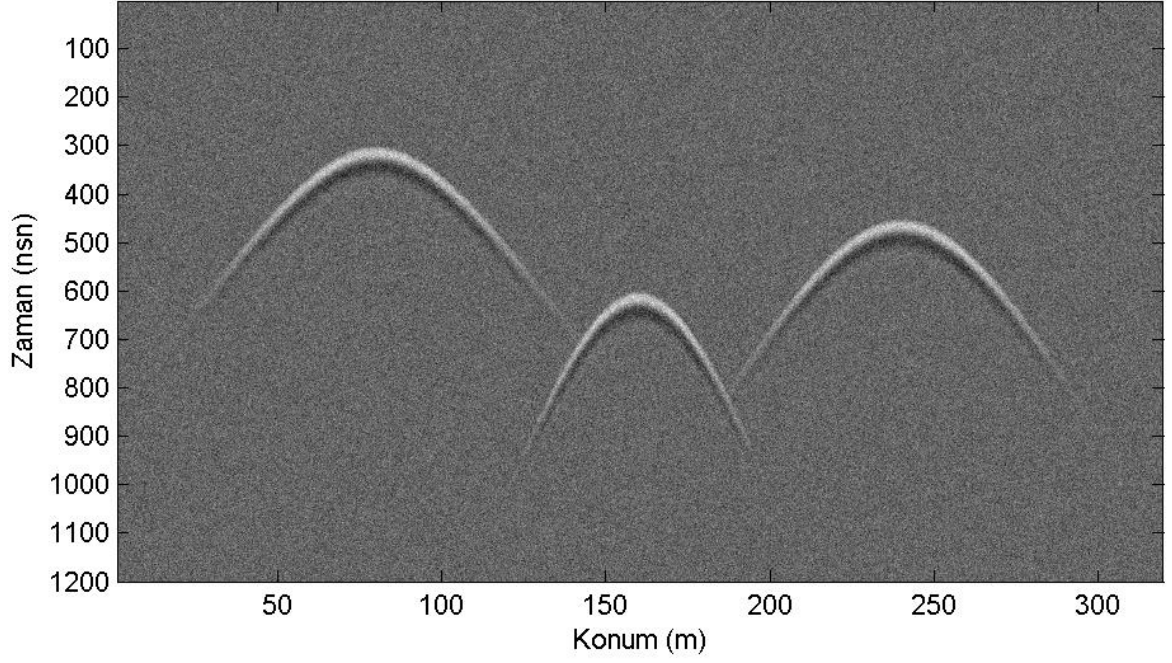
Tüm çevrimler bittiğinde elde edilen listenin verileri Şekil 15(a)'da verilmektedir. Bu liste elde edildikten sonra k-ortalama kümeleme yöntemiyle iki ayrı küme oluşturularak en iyi sonuçlar Şekil 15(b)'de kırmızı renkle gösterilmektedir.



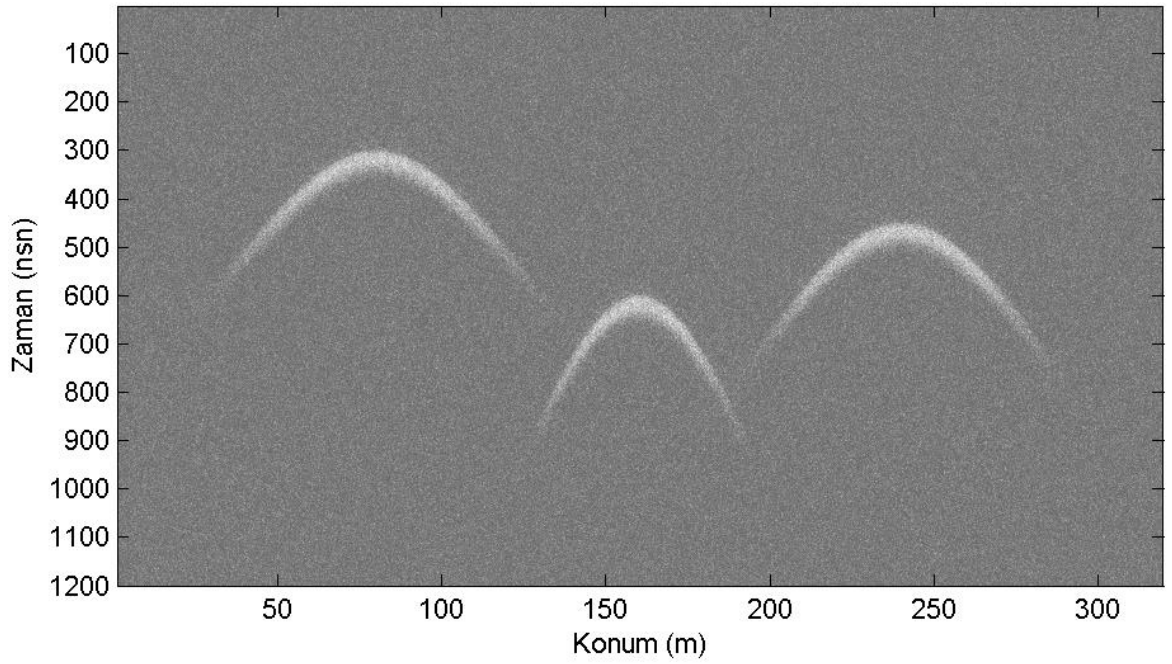
Şekil 15. Çözüm listesinin ayıklanması; (a) çözüm listesi diyagramı; (b) çözüm listesinin kümelenmiş diyagramı.

Algoritmayı gürültülü bir ortamda da denemek amacıyla Şekil 10'da verilen yapay radargram görüntüsüne ortalaması 0 olan bir gauss gürültüsü eklenmiştir (Şekil 16). Gürültülü radargram görüntüsüne Hilbert dönüşümü uyguladığında Şekil 17'deki görüntü elde edilir. Bu görüntüye önerilen algoritmanın uygulanmasında 1. Çevrimin görüntüsü Şekil 18'de verilmektedir. Burada beyaz daireler yiyecek bölgelerini göstermektedir. İlk sınır değerinin aşılması sonucunda elde edilen en iyi üç hiperbol Şekil 19'da verilmiştir. Şekil 20'de 25. çevrimin sonucuna göre bulunan çözümler ile hiperbollerden bir tanesinin tam çakışma gösterirken bir tanesi yarım çakışma diğeri ise kötü bir çakışma görülmektedir. Şekil 21'de iki hiperbol tam çakışma gösterirken bir tane hiperbol yarım bir çakışma göstermektedir. Şekil 22'de yarım çakışmada iyileşme gözükmesine rağmen hala yeterli bir çakışma söz konusu değildir. Şekil 23'te 118. çevrimin sonucu tüm

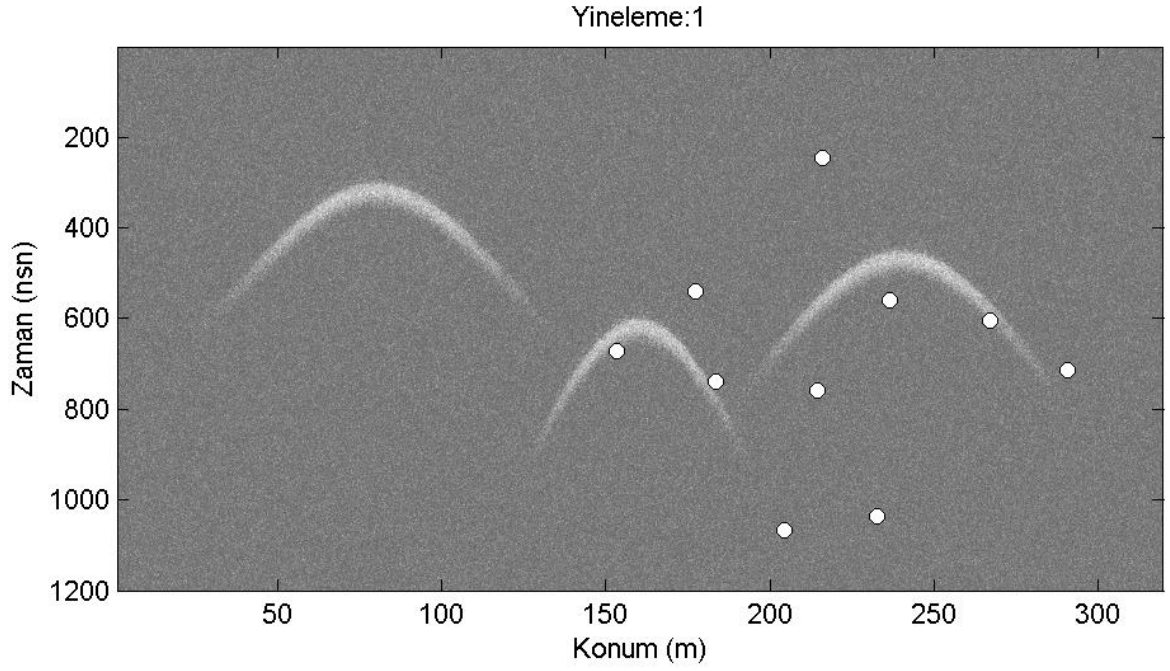
hiperbollerde tam bir çakışmanın olduğu görülmektedir. Sonraki çevrimlerde ise sonuç değişmemektedir.



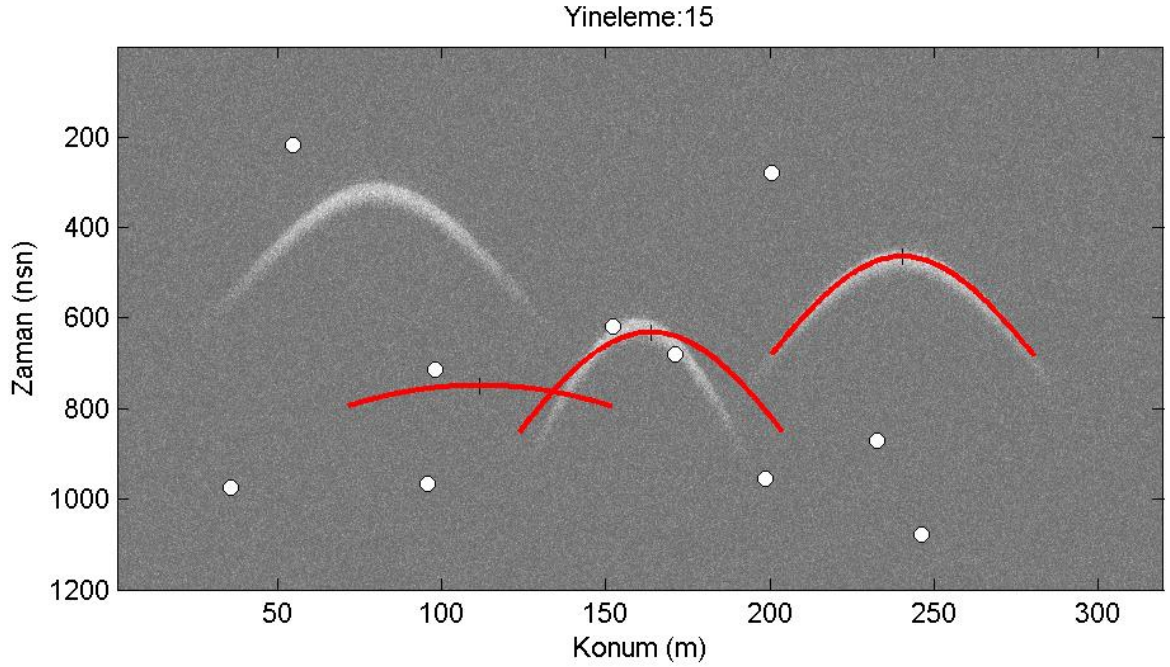
Şekil 16. Gürültülü yapay radargram görüntüsü.



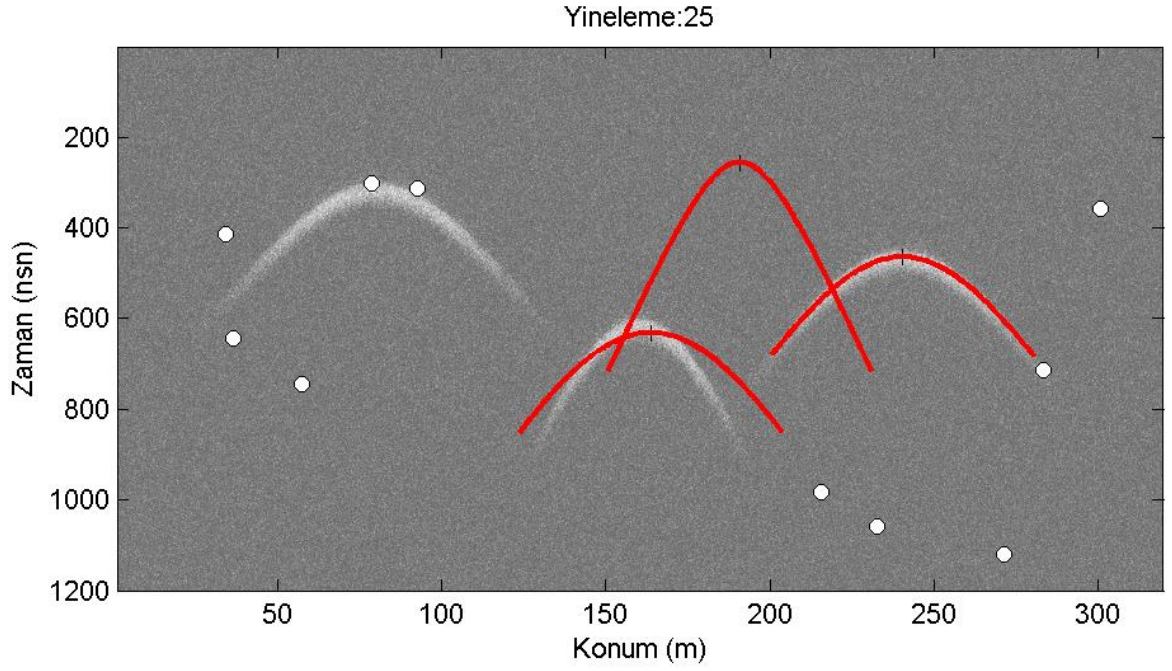
Şekil 17. Gürültülü yapay radargram görüntüsüne Hilbert dönüşümü uygulaması.



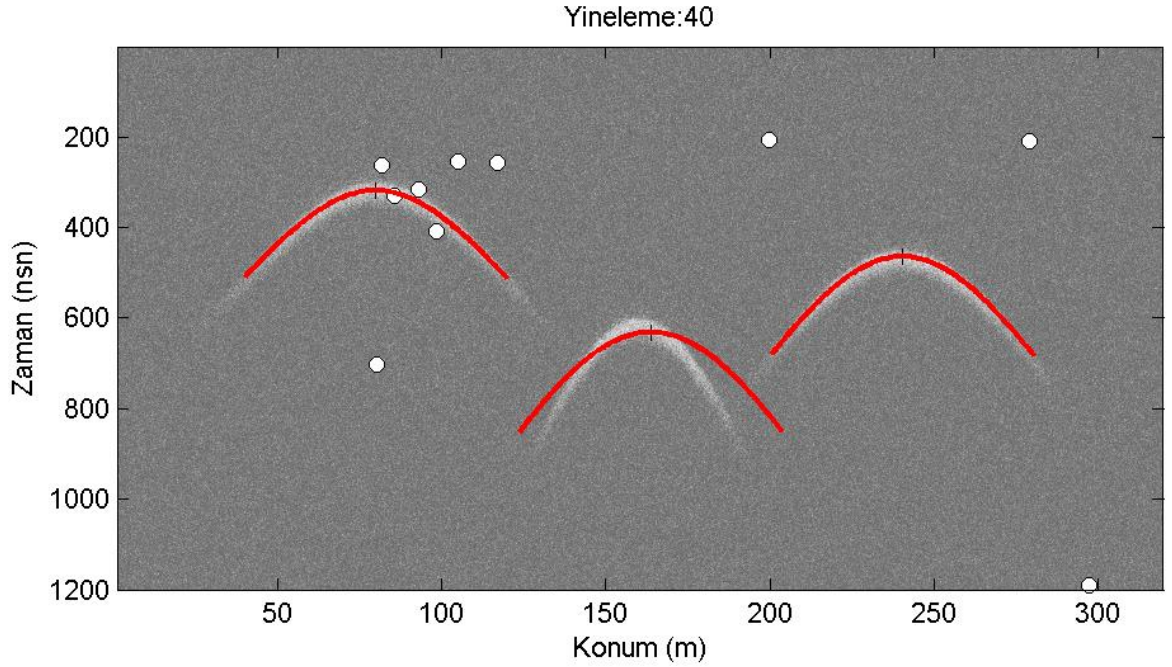
Şekil 18. Önerilen algoritmanın 1. çevrimdeki yiyecek bölgeleri yuvarlak noktalar biçiminde gösterimi.



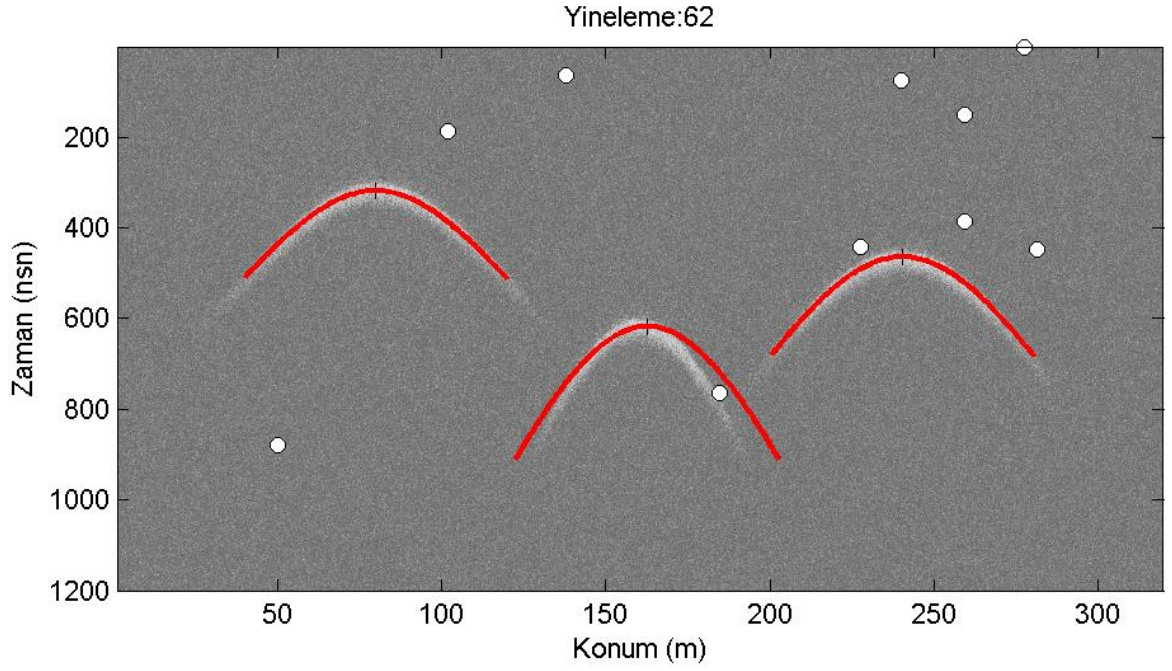
Şekil 19. Önerilen algoritmanın 15. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.



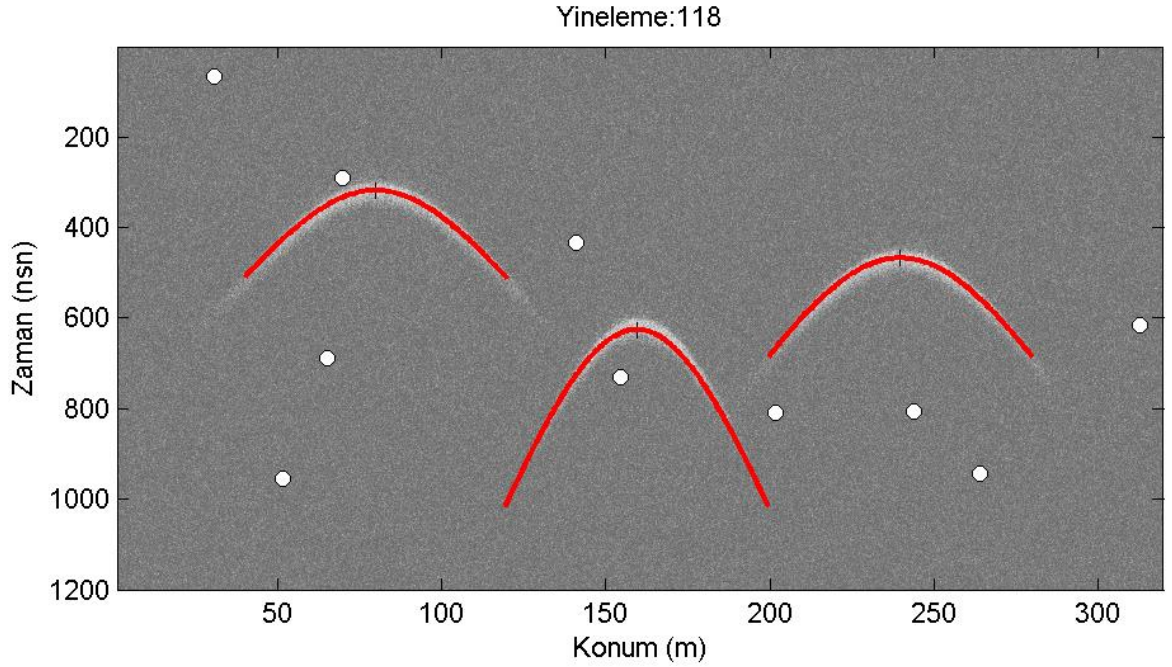
Şekil 20. Önerilen algoritmanın 25. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.



Şekil 21. Önerilen algoritmanın 40. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.

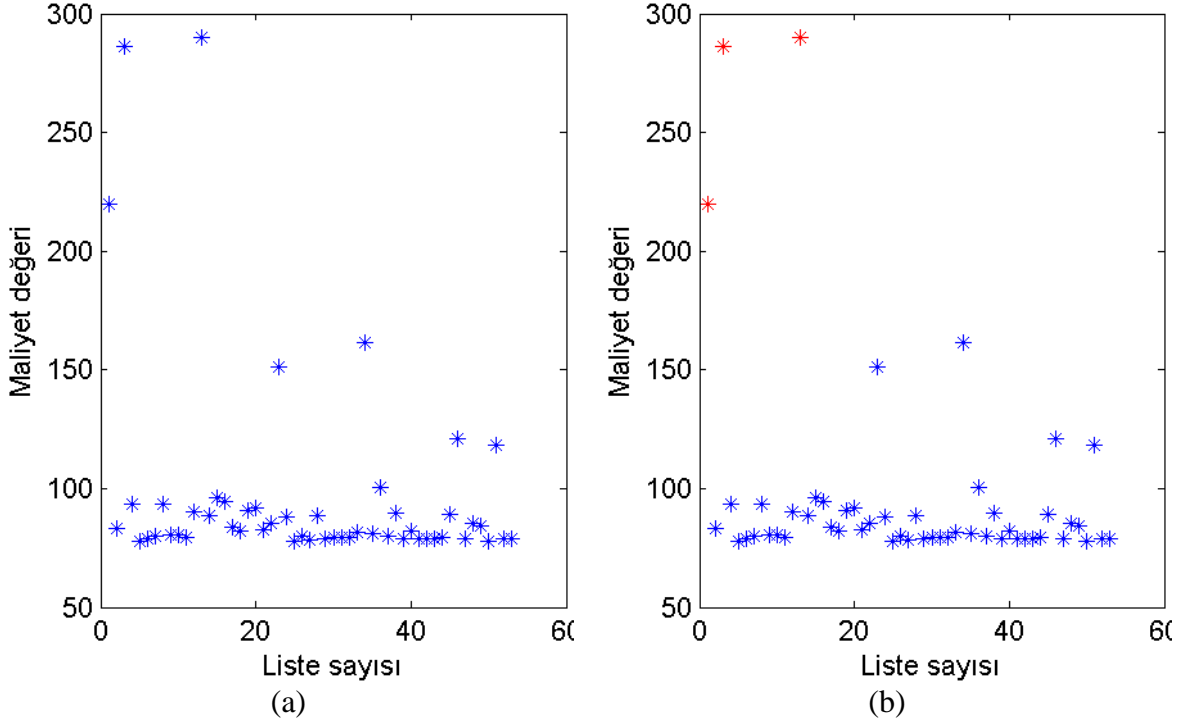


Şekil 22. Önerilen algoritmanın 62. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.



Şekil 23. Önerilen algoritmanın 118. çevrim sonucunda çözüm listesine kayıt edilen en iyi üç hiperbol gösterimi.

Tüm çevrimler bittikten sonra çözüm listesindeki noktaların diyagramı Şekil 24(a)'da verilmiştir. Bu liste Şekil 24(b)'deki iki ayrı kümeye yarılarak en iyi hiperbollerin parametreleri bulunmuştur. Bu örnekte listedeki değer sayısı gürültüden dolayı artmış olmakla birlikte hiperbollerin maliyet değerleri daha açık bir şekilde ayıklanmıştır.



Şekil 24. Çözüm listesinin ayıklanması; (a) çözüm listesi diyagramı; (b) çözüm listesinin kümelenmiş diyagramı.

2.3.2. Gerçek Radargramdaki Hiperbollerin Algılanması

Önerilen algoritmanın yapay veriler üzerindeki başarımı önceki bölümlerde ayrıntılı bir şekilde verilmiştir. Bu bölümde ise gerçek veriler üzerinde yöntemin başarımı test edilecektir. Yapay hiperboller üzerinde yapılan işlemlerin sonuçlarına bakılırsa, algoritmanın kusursuz çalıştığı görülmektedir. Ancak önemli olan gerçek görüntülerde de yapay görüntülerdeki gibi bu kadar iyi çalışmasıdır.

L. Capineri ve arkadaşları [33] Hough dönüşümünden yararlanarak hiperbollerini bulmaya çalışmışlar ancak daha önce bahsedildiği gibi Hough dönüşümünün uzun bir hesaplama zamanı ve büyük bir belleğe ihtiyaç duyduğu bilmektedir. Yaptıkları iş iyi sonuçlara vardığına da hesaplama zamanı ve büyük bir belleğe ihtiyacı bu yöntemin

büyük bir dezavantajdır. Daha önce anlattığımız gibi Hough dönüşümüyle önerilen yöntem arasında yaklaşık %0,02 gözetik hesaplama oranı bulunmaktadır. Bu oran görüntünün boyutu büyüdükçe daha düşmektedir. Bu çalışmada, L. Capineri ve arkadaşlarının [33] yaptığı gibi Hough yöntemini kullanılmıştır. Ancak Hough dönüşümü doğrudan değil geliştirilen algoritmadaki maliyet fonksiyonun bulunmasında kullanılmıştır.

Örnek olarak Şekil 25'te verilen yer radarı görüntüsü [34] hiperbollerini tam olarak görülemeyen bir görüntü olarak seçilmiştir. Seçilen gerçek radargram görüntüsüne önerilen yöntemin uygulanmasında elde edilen 1. çevrimdeki görüntü Şekil 25'de verilmiştir. Tüm işlemlerde gerçek radargram görüntüsünün Hilbert dönüşümü kullanılmasına rağmen anlaşılabilirliği yüksek olması için gösterimler gerçek görüntü üzerinde yapılmıştır.

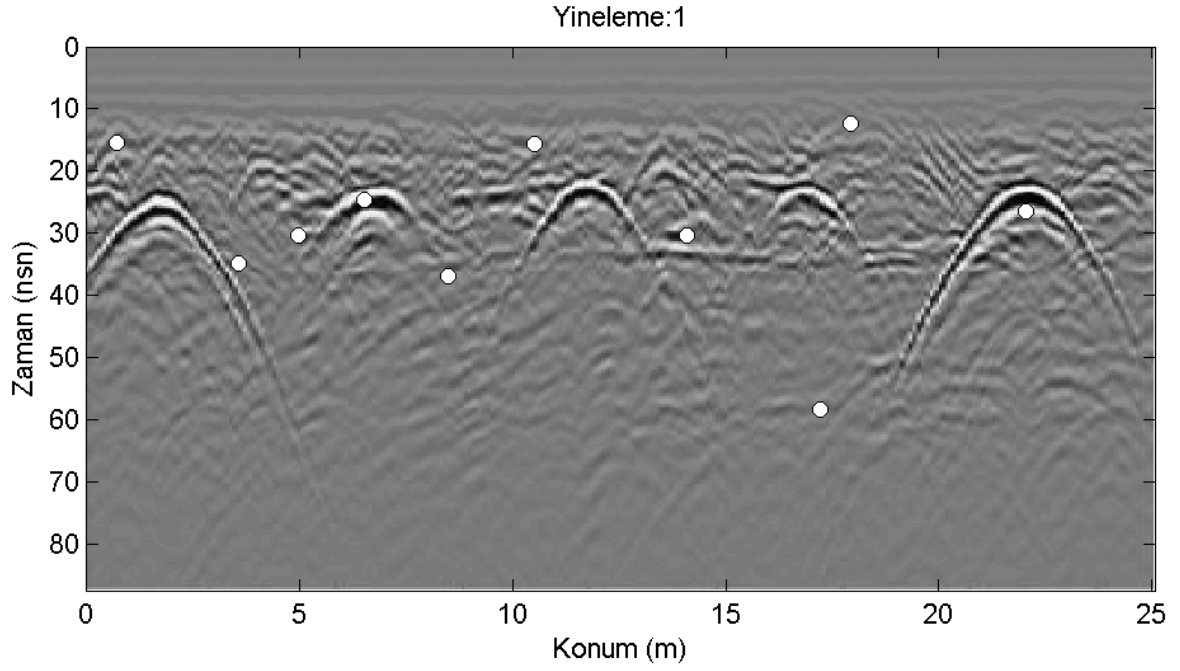
Şekil 26'da gerçek radargram görüntüsünün 17. çevrimde ilk defa limit değeri aşılarak en iyi hiperbollerin gösterimi yapılmıştır. Burada dört hiperbol konumu yaklaşık bir çakışma verirken bir tanesi farklı bir konuma yerleşmiştir. Öte yandan sağdaki hiperbolün dışındaki tüm hiperbollerin gerçek hızlarında bir çakışma olmadığı görülmektedir.

Şekil 27'de tüm hiperboller gerçek konumlarına yerleşirken en sağdaki hiperbol dışındaki diğerlerinin hız çakışmasının sağlanmadığı görülmektedir.

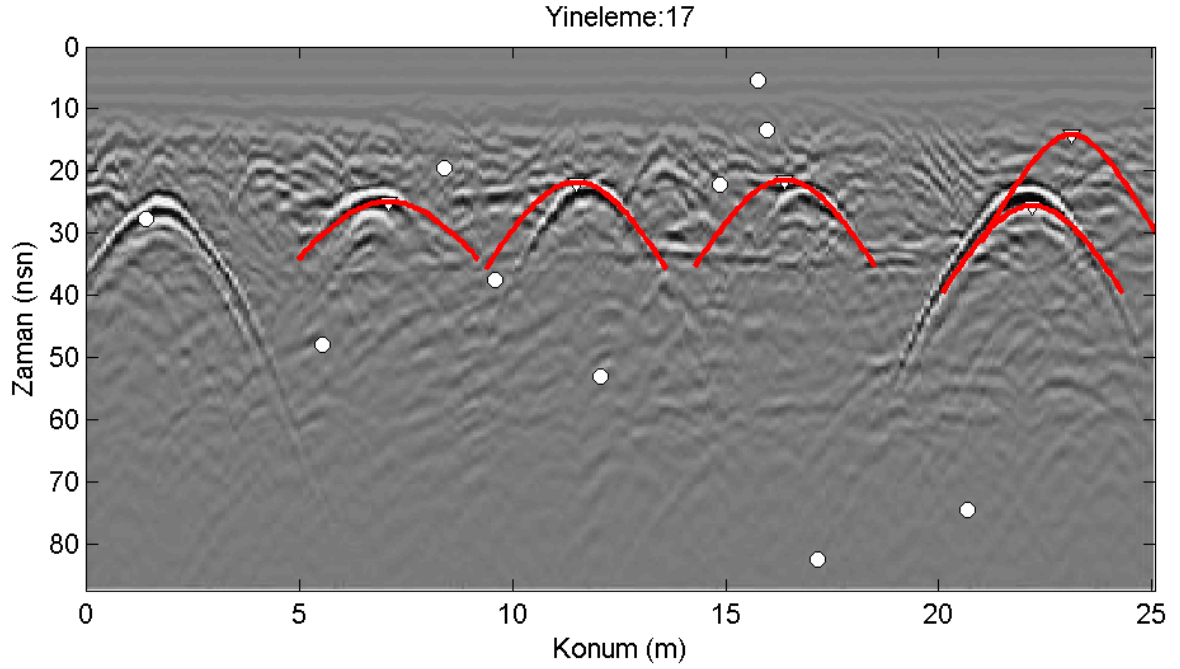
Şekil 28'de 75. çevrim sonucu soldan 3. hiperbolün hız çakışması en iyi şekilde gerçekleşmiştir.

Şekil 29'da 128. çevrim sonucunda soldan 1. hiperbolün konumunda bir iyileşme görülürken Şekil 30'da ise hız bilgisinde iyileşme görülmüştür.

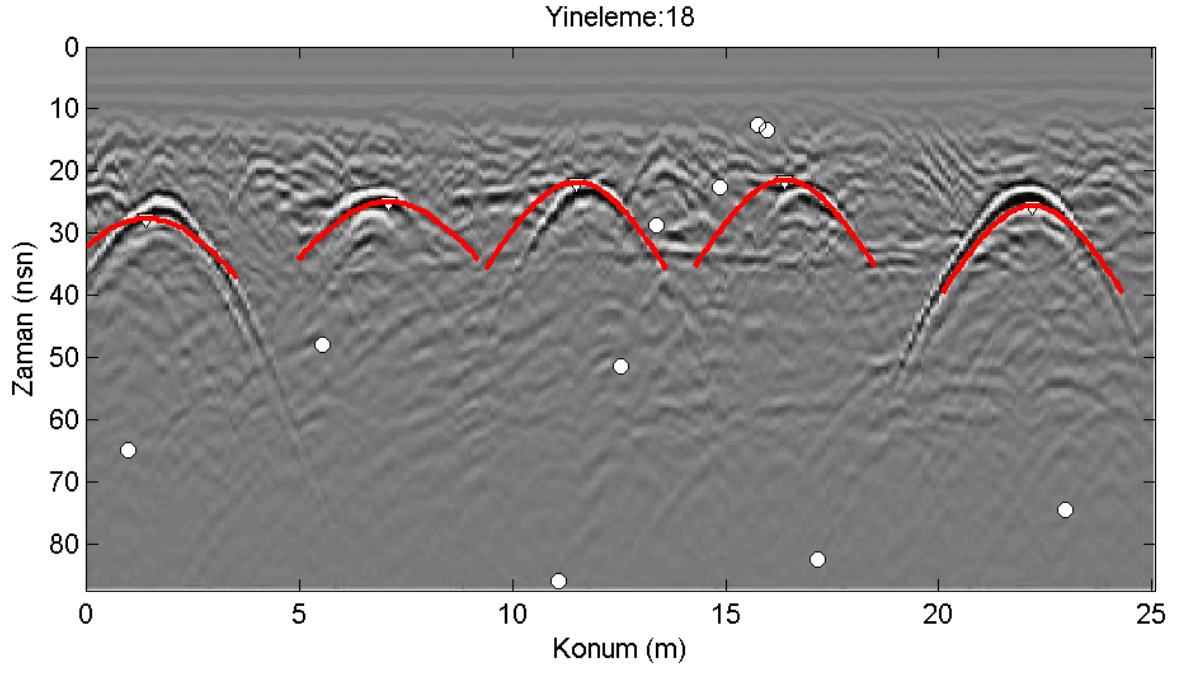
Tüm çevrimler sonucunda başka bir iyileşme görülmediğinden Şekil 31(a)'da çözüm listesinin diyagramı gösterilmiştir. Bu listenin kümelenmesi bazı hiperbollerin net olmamasından dolayı iyi gerçekleştirilmemiştir. Ancak hiperbol sayısı önceden verildiğinde ve önceden bir eşik değeri belirlenirse hiperboller parametreleri net bir biçimde bulunabilir.



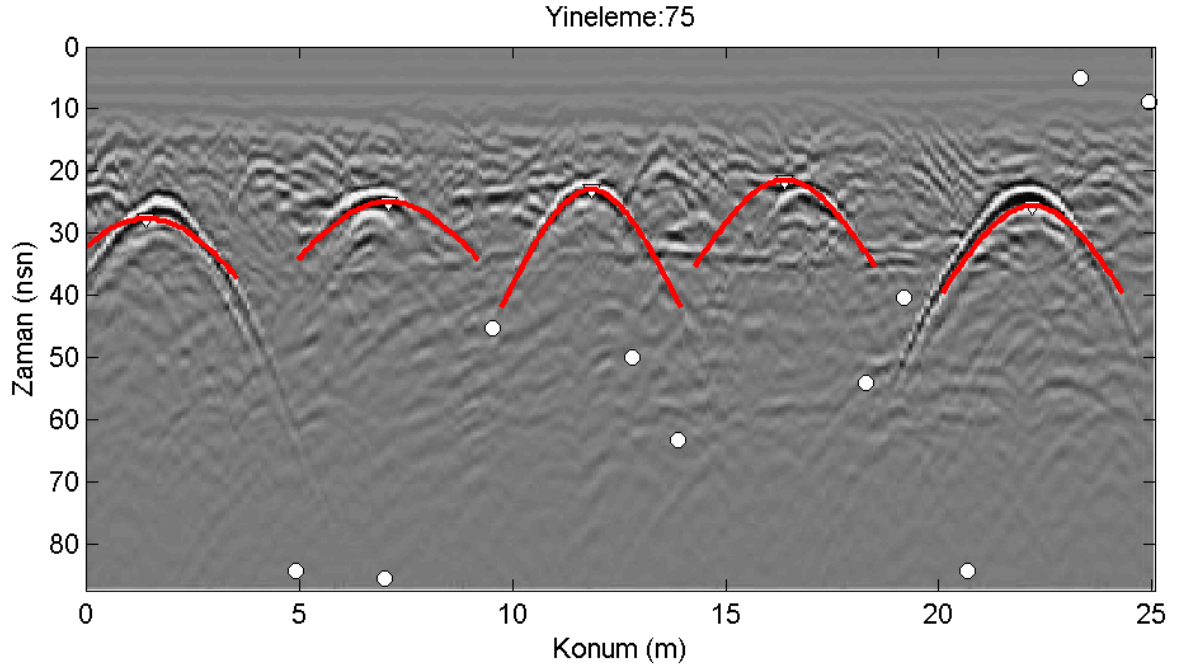
Şekil 25. Gerçek radargram görüntüsünün 1. çevrimdeki yiyecek noktalarının gösterimi



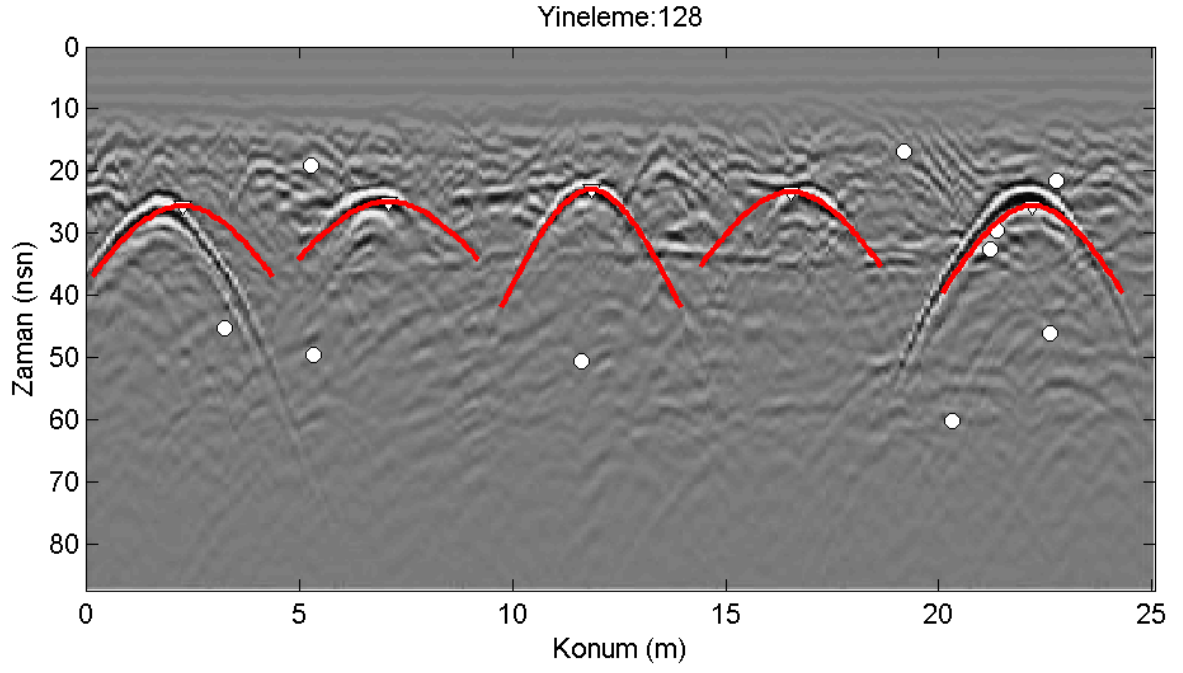
Şekil 26. Gerçek radargram görüntüsünün 17. çevrimde ilk defa limit değeri aşılarak en iyi hiperbollerin gösterimi.



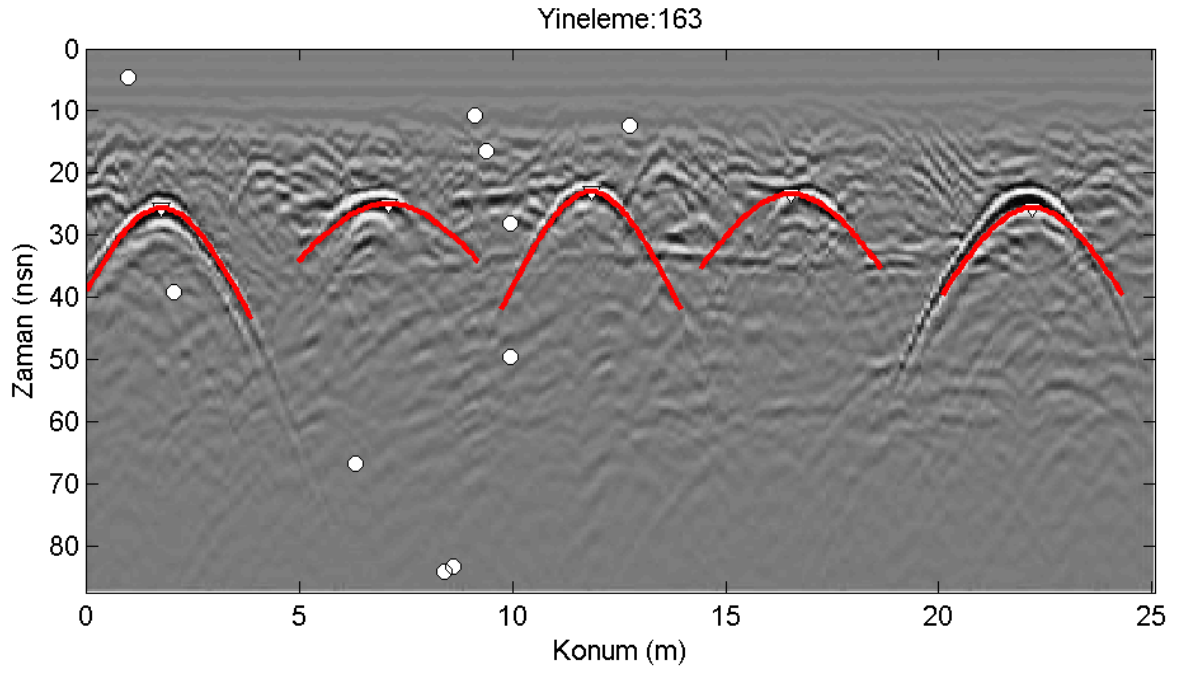
Şekil 27. Gerçek radargram görüntüsünün 18. çevrimdeki en iyi hiperbollerin gösterimi.



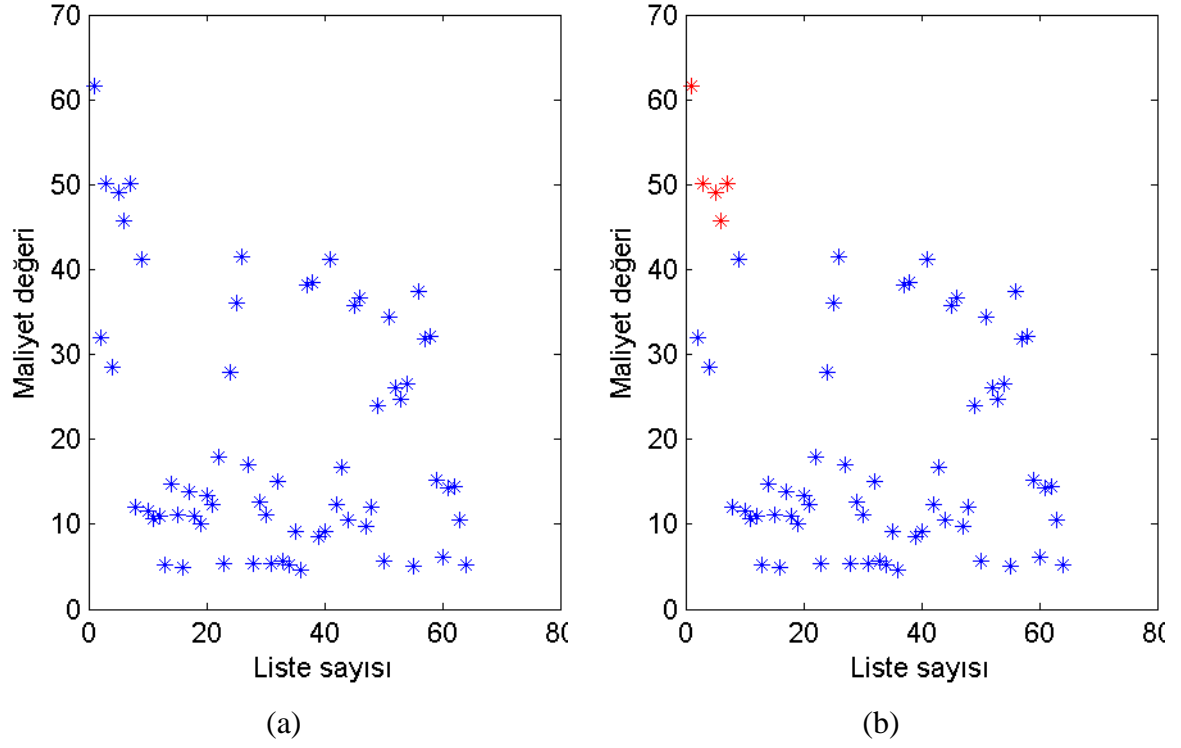
Şekil 28. Gerçek radargram görüntüsünde 75. çevrimdeki en iyi hiperbollerin gösterimi.



Şekil 29. Gerçek radargram görüntüsünde 128. çevrimdeki en iyi hiperbollerin gösterimi.



Şekil 30. Gerçek radargram görüntüsünün 163. çevrimdeki en iyi hiperbollerin gösterimi.



Şekil 31. Çözüm listesinin ayıklanması; (a) çözüm listesi diyagramı; (b) çözüm listesinin kümelenmiş diyagramı.

3. BULGULAR VE SONUÇLAR

Bu çalışmada, çok tepeli yapay arı koloni algoritmasının literatürden [18] farklı olarak herhangi bir dış parametre kullanmadan gerçekleştirilmesi için bir algoritma geliştirilmiştir. [18]'de verilen algoritma tepelerin birbirine yakın olması durumunda istenen sonucu vermeyebilmektedir. Ayrıca kapsama yarıçapı yanlış seçilmesi durumunda birçok çözümün dışarıda kalmasına neden olur.

Geliştirilen yöntem, örüntü tanıma sistemlerinde en çok kullanılan çoklu daire algılamada etkili sonuçlar verdiği görülmüştür. Öte yandan Hough dönüşümü yardımıyla birçok geometrik şeklin algılanması oldukça uzun zamanlarda gerçekleştirilirken bu yöntemle logaritmik olarak daha kısa sürede gerçekleşmektedir. Görüntülere gürültü eklendiğinde ayrımlılığın artmasından dolayı algılanan kenarlar daha keskin kenarlara sahip olduğu için çözümler daha iyi sonuçlar vermiştir.

Yöntemin gerçek yaşam problemlerinden olan yer radarı görüntülerindeki hiperbollerin algılanması işleminde uygulanması için öncelikle yapay radargram görüntülerinde uygulanmış ve sonuçlar tam olarak verdiği görülmüştür. Gürültülü yapay radargramlarda ise çözümün anlaşılabilirliği azalmıştır. Sayısal görüntülerde gürültü eklemek iyi sonuç verirken radargramlarda Hilbert dönüşümü yardımıyla çözümün iyileştirilmesi yöntemi gürültülerden kötü yönde etkilenmektedir.

Aynı işlemleri gerçek radargram verilerinde uygulanmasıyla da tatmin edici sonuçlara ulaşılmıştır. Radargram görüntülerindeki hiperboller zaman-konum bilgisinden derinlik bilgisini elde etmek için önemli bir parametredir. Öte yandan hiperbollerin hızlı ve otomatik bulunması gerçek zamanlı ölçümlerde hayati öneme sahip olabilmektedir. Özellikle plastik mayınların taranırken bulunması ve derinliğinin kestirilmesi, çevre kirliliğine neden olan gömülerin derinliğini ve konumunun belirlenmesi, arkeolojik aramalarda anında sonuç vermesi gibi birçok gerçek zamanlı talebe cevap vermektedir.

4. ÖNERİLER

Geliştirilen yöntemin başarımını artırmak için görüntüleme ön işlemlerin etkisi ileride araştırılabilecek konulardan olabilir. Hough dönüşümü uygulanan bir çok geometrik şekil yine akıllı optimizasyon yöntemleri kullanılarak daha kısa sürede algılanabilir. Geometrik şekillerin algılanmasında uygulanacak optimizasyon yönteminde bulanık geçişli kenar algılama yöntemi geliştirilerek özellikle net olmayan görüntülerdeki yapılar ortaya çıkarılabilir.

Radargram görüntülerinde hiperbollerin bulunması için geliştirilen yöntemin başarımını artırmak için hiperbolün maliyet fonksiyonundaki Hough dönüşümü hesaplanmasında düşey yönde bir aralık kullanılarak hesaplanabilir. Malum radargramdaki hiperboller çizgiler şeklinde değil şeritler şeklinde yer almaktadır.

Özellikle gürültülü radargram görüntülerinde birçok görüntü hiperbollerin verdiği maliyet değerine yakın değerler verebilir. Bu durumda hiperbol seçiminde sıkıntılar oluşmaktadır. Bu sorunu gidermek için bir birine yakın hiperbollerin aynı hızları verdiği kabul edilirse, maliyet değeri düşük olan ancak maliyet değeri yüksek olan bir hiperbole yakın olması o hiperbolün elenmesini sağlayacaktır. Böylelikle elenen hiperboller çıkarıldıktan sonra listedeki hiperbollere kümeleme algoritması uygulanarak daha net bir sınıflandırma yapılabilir.

5. KAYNAKLAR

1. Kesemen, O., Yer Radarı (GPR) Verilerinin İşlenmesinde Yeni Yaklaşımlar, Doktora Tezi, Karadeniz Teknik Üniversitesi, Fen Bilimleri Enstitüsü, 2007.
2. Daniels, D.J., Gunton, D.J. and Scott, H.F., Introduction to subsurface radar. IEE Proceedings, 135 (1988) 278–320.
3. Daniels, D.J., Surface-Penetrating Radar, IEE Radar, Sonar, Navigation and Avionics Series 6, London, 1996.
4. Davis, J. L. and Annan, A. P., Ground penetrating radar for high resolution mapping of soil and rock stratigraphy, Geophysical Prospecting, 37 (1989) 531–551.
5. Kruk, J. V. D., Three-dimensional imaging of multi-component ground penetrating radar data, PhD thesis, Delft University of Technology, <http://borneo.mp.tudelft.nl/~jvdkruk>, Delft, 24.04.2013.
6. Capineri, L., Grande1, P. and Temple, J. A. G., Advanced image-processing technique for real-time interpretation of ground-penetrating radar images, International Journal of Imaging Systems and Technology, 9, 1 (1998) 51–59.
7. Carlson C. F., Lecture 10: Hough circle transform. Rochester Institute of Technology: Lecture Notes, October 11, 2005.
8. Karaboğa, D. Yapay Zeka Optimizasyon Algoritmaları, Nobel Yayınevi, Ankara, 2011.
9. Mehdi, S. M., Evolutionary algorithms, Iran University of Science & Technology - IUST, Esfahlan, 2008.
10. Haupt, R. L. and Haupt, S. E., Practical Genetic Algorithms, 2nd Edition, John Wiley & Sons Inc, 2004.
11. Pedregal, P., Introduction to Optimization, Springer, New York Inc, 2004.
12. Karaboga, D, and Akay, B., Artificial Bee Colony Algorithm on Training Artificial Neural Networks, Signal Processing and Communications Applications, SIU 2007, IEEE 15th. (2007) 1 – 4.
13. Bonabeau, E., Dorigo, M. and Theraulaz, G., Swarm Intelligence: From Natural to Artificial Systems. New York, Oxford University Press, 1999.
14. Yang, X., Engineering Optimizations via Nature-Inspired Virtual Bee Algorithms, Lecture Notes in Computer Science, 3562 (2005) 317-325

15. Tereshko, V., Reaction-diffusion model of a honeybee colony's foraging behavior, Lecture Notes in Computer Science, 1917(2000) 807-816.
16. D. Karaboga, An Idea Based On Honey Bee Swarm For Numerical Optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department 2005.
17. Storn, R. and Price, K., Differential Evolution -- A Simple And Efficient Heuristic For Global Optimization Over Continuous Spaces, Journal of Global Optimization, 11, 1 (1997) 341-359.
18. Cuevas, E., Sencio'n-Echauri, F., Zaldivar, D. and Pe'rez-Cisneros, M., Multi-Circle Detection On Images Using Artificial Bee Colony (ABC) Optimization, Soft Computing, 16 (2012) 281-296
19. Li, X., Adaptively Choosing Neighbourhood Bests Using Species In A Particle Swarm Optimizer For Multimodal Function Optimization, Lecture Notes in Computer Science, 3102, 1 (2004) 105-116.
20. Barrera, J. and Coello, C. A., A Particle Swarm Optimization Method for Multimodal Optimization Based on Electrostatic Interaction, Lecture Notes in Computer Science, 5845,1 (2009) 622-632.
21. Iwamatsu, M., Locating All The Global Minima Using Multi-Species Particle Swarm, Proceedings of the IEEE Congress on Evolutionary Computation, (2006) 816-822.
22. Iwamatsu, M., Multi-Species Particle Swarm Optimizer For Multimodal Function Optimization, Transactions on Information and Systems, (2006) 1181-1187.
23. Li, J.-P., Balazs, M.E., Parks, G.T. and Clarkson, P.J, A Species Conserving Genetic Algorithm For Multimodal Function Optimization, Evolutionary Computation, (2002) 207-234 .
24. Goldberg, D.E. and Richardson, J., Genetic Algorithm with Sharing For Multimodal Function Optimization, Proceedings of the Second International Conference on Genetic Algorithms, (1987) 41-49.
25. Brits, R., Engelbrecht, A.P. and Bergh, F.V.D., A Niching Particle Swarm Optimizer. Proceedings of the Conference on Simulated Evolution and Learning, (2002) 692-696.
26. Bird, S. and Li, X. Adaptively Choosing Niching Parameters in a PSO, Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, (2006) 3-10.
27. Passaro, A. and Starita, A., Particle Swarm Optimization For Multimodal Functions, Journal Artificial Evolution and Applications, (2008) 1-15.

28. Kennedy, J., Stereotyping: Improving Particle Swarm Performance with Cluster Analysis, Proceedings of the 2000 Congress on Evolutionary Computation, 2, (2000) 1507–1512.
29. Pelleg, D. and Moore, A., X-Means: Extending K-Means With Efficient Estimation Of The Number Of Clusters, Proceedings of the 17th International Conference on Machine Learning, (2000) 727–734.
30. Li, X., A Multimodal Particle Swarm Optimizer Based On Fitness Euclidean-Distance Ratio, Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, (2007) 78–85.
31. Peram, T., Veeramachaneni, K. and Mohan, C.K., Fitness-Distance-Ratio Based Particle Swarm Optimization, Proceedings of the 2003 IEEE Swarm Intelligence Symposium, (2003) 174–181.
32. MacQueen, J., Some Methods For Classification And Analysis Of Multivariate Observations. Proc. 5th Berkeley Symposium, (1967) 281–297.
33. Capineri, L., Grande, P. and Temple, J. A. G., Advanced Image-Processing Technique for Real-time Interpretation of Ground-Penetrating Radar Images, International Journal of Imaging Systems and Technology, 9,1 (1988) 51–59.
34. <http://iecfl.com/products-overview/columns-2#!/home>, Key Term: Innovative Engineering, 18.04.2013.

ÖZGEÇMİŞ

Taymaz RAHKAR FARSHİ,28 Aralık 1985 tarihinde Tabriz (İran)'da doğdu. Ön lisansını İslamic Azad Üniversitesinde 2005-2006 öğretim yılında tamamladı. 2006-2007 öğretim yılında Nabi Akram Üniversitesi, Fen Bilimler Fakültesi, Bilgisayar Mühendisliği bölümüne yerleşti. 2009 yılında bu üniversiteden mezun oldu. 2010 yılında Karadeniz Teknik Üniversitesi, Fen Bilimler Enstitüsü, İstatistik ve Bilgisayar Anabilim dalında tezli yüksek lisans programına başladı.