

**GENETİK ALGORİTMA YARDIMI İLE YAPAY AÇIKLIKLI
RADARDA HEDEF KONUMUNUN BELİRLENMESİ**

DOKTORA TEZİ
Y. Müh. Serdar KARGIN
Enstitü No: 504932010

100821

100821

Tezin Enstitüye Verildiği Tarih : 21 Ağustos 2000
Tezin Savunulduğu Tarih : 13 Kasım 2000

Tez Danışmanı : Prof. Dr. Bingül YAZGAN

Diğer Jüri Üyeleri Prof.Dr. Ercan TOPUZ

Prof.Dr. A. Hamdi KAYRAN

Prof.Dr. Filiz GÜNEŞ (Y.T.Ü.)

Prof. Dr. Emin ANARIM (B.Ü.)

KASIM 2000

ÖNSÖZ

Bu tez çalışması sırasında sabrı ve özverili yardımlarıyla bana her zaman destek olan ve yol gösteren hocam sayın Prof. Dr. Bingöl YAZGAN'a en içten teşekkürlerimi sunarım.

Doktora eğitimimi yapmama olanak sağlayan Hava Kuvvetleri Komutanlığındaki değerli komutanlarıma teşekkürü bir borç bilirim.

Eylül 2000

Serdar KARGIN



İÇİNDEKİLER

KISALTMALAR	iv
TABLO LİSTESİ	v
ŞEKİL LİSTESİ	vi
ÖZET	viii
SUMMARY	xi
1. GİRİŞ	1
2. YAPAY AÇIKLIKLI RADAR VE OPTİMİZASYON YÖNTEMLERİ	
2.1.Yapay Açıklıklı Radar Kavramı, Geometrisi ve Ters Dönüşüm Bağıntısı	6
2.1.1. Yapay Açıklıklı Radar Prensibi	6
2.1.2. Yapay Açıklıklı Radar Görüntüleme Sistemi, Geometrisi ve Ters Dönüşüm Bağıntısı	11
2.1.3. Yapay Açıklıklı Radarda Yeniden Oluşturma Algoritması ve Akış Diyagramı	16
2.2. Optimizasyon Yöntemleri	19
2.2.1. Hooke Jeeves Optimizasyon Algoritması ve Akış Diyagramı	20
2.3.Global optimizasyon Yöntemi Genetik Algoritma	22
2.3.1. Genetik Algoritma Kavramı ve Diğer Geleneksel Yöntemler arasındaki Farklar	22
2.3.2. Basit Genetik Algoritma	28
2.3.3. Genetik Algoritmanın Teorisi	34
3. GENETİK ALGORİTMANIN YAPAY AÇIKLIKLI RADAR TERS DÖNÜŞÜM BAĞINTISINA UYGULANMASI	41
3.1.Uygulanan Yöntem	41
3.2.Genetik Algoritmanın Uygulama Adımları ve Akış Diyagramı	43
4. UYGULAMA ÖRNEKLERİ	47
4.1. Seçilen Uygulama Örnekleri	47
4.2. Yapay Açıklıklı Radar Ters Dönüşüm Algoritması ile Gerçekleştirilen Uygulama Örnekleri	49
4.3. Hooke Jeeves Algoritması ile Y.A. Radarda Gerçekleştirilen Uygulama Örnekleri	57
4.4. Genetik Algoritma ile Y.A. Radarda Gerçekleştirilen Uygulama Örnekleri	61
5. SONUÇLAR VE TARTIŞMA	76
KAYNAKLAR	80
ÖZGEÇMİŞ	83

KISALTMALAR

YAR :Yapay Açıklıklı Radar
GA :Genetik Algoritma



DOKÜMANLARIYI İNDİRİN

TABLO LİSTESİ

	<u>Sayfa No</u>
Tablo 2.1. Diziler , amaç fonksiyon değerleri ve kopyalama yüzdeleri.....	30
Tablo 2.2. Genetik algoritmalar.....	32
Tablo 2.3. İşlem havuzu ve çaprazlama işleminin uygulanması.....	33
Tablo 4.1. Tek Nokta dağılımı için algoritma parametrelerine göre hata değişimini gösteren şekiller.....	63
Tablo 4.2. Dört nokta dağılımı için algoritma parametrelerine göre hata değişimi gösteren şekiller.....	67
Tablo 4.3. Sekiz nokta dağılımı için algoritma parametrelerine göre hata değişimini gösteren şekil.....	72



ŞEKİL LİSTESİ

Sayfa No

Şekil 2.1	:Yandan bakış yapay açıklıklı radar prensibi.....	7
Şekil 2.2	:Yapay açıklıklı radar görüntüleme prensibi.....	8
Şekil 2.3	:Uzun doğrusal anten dizisi.....	9
Şekil 2.4	:Yapay açıklıklı radar görüntüleme sisteminin geometrisi.....	12
Şekil 2.5	:Yapay açıklıklı radarda yeniden oluşturma algoritmasının akış diyagramı.....	18
Şekil 2.6	:Hooke Jeeves optimizasyon algoritması akış diyagramı.....	21
Şekil 2.7	:Tek tepe noktalı uzay.....	23
Şekil 2.8	:Çok tepe noktalı uzay.....	24
Şekil 2.9	:Pratikte karşılaşılan bir amaç fonksiyonu örneği.....	25
Şekil 2.10	Arama yöntemlerinin etkinliği.....	26
Şekil 2.11	: $y=x$ amaç fonksiyonu.....	27
Şekil 2.12.	:Kapalı kutu örneği.....	27
Şekil 2.13	:Üreme operatörü için rulet çarkı.....	30
Şekil 2.14	:Çaprazlama işleminin şematik gösterilimi.....	31
Şekil 2.15	:Karar değişkenlerinin bir kromozom olarak kodlanması.....	36
Şekil 2.16	:İki Nokta, çok nokta ve düzgün çaprazlama.....	40
Şekil 3.1	:Gözönüne alınan YAR görüntüleme geometrisi.....	41
Şekil 3.2	:Kromozom örneği.....	43
Şekil 3.3	:Karar değişkenleri havuzu.....	44
Şekil 3.4	: Bit sayısı=5 ve çaprazlama bit konumu=13 için örnek çaprazlama işlemi.....	44
Şekil 3.5	:YAR'ın G.A.'ya uygulanmasını sağlayan programın akış diagramı.....	46
Şekil 4.1	:Tek nokta hedef için gözönüne alınan YAR görüntüleme geometrisi.....	47
Şekil 4.2	:Dört nokta hedef için gözönüne alınan YAR görüntüleme geometrisi.....	48
Şekil 4.3	:Sekiz nokta hedef için gözönüne alınan YAR görüntüleme geometrisi.....	48
Şekil 4.4	:Tek nokta hedef için $u \in [0,5]$ ve $nn = 10$ için elde edilen YAR görüntüsü.....	50
Şekil 4.5	:Tek nokta hedef için $u \in [0,5]$ ve $nn = 40$ için elde edilen YAR görüntüsü.....	50
Şekil 4.6	:Tek nokta hedef $u \in [0,150]$ ve $nn = 10$ için elde edilen YAR görüntüsü.....	51
Şekil 4.7	:Tek nokta hedef $u \in [0,150]$ ve $nn = 40$ için elde edilen YAR görüntüsü.....	51
Şekil 4.8	:Dört nokta hedef $u \in [0,5]$ ve $nn = 10$ için elde edilen YAR görüntüsü.....	52
Şekil 4.9	:Dört nokta hedef $u \in [0,5]$ ve $nn = 40$ için elde edilen YAR görüntüsü.....	53
Şekil 4.10	:Dört nokta hedef $u \in [0,150]$ ve $nn = 10$ için elde edilen YAR görüntüsü.....	54

Şekil 4.11	:Dört nokta hedef $u \in [0,150]$ ve $nn=40$ için elde edilen YAR görüntüsü.....	54
Şekil 4.12	:Sekiz nokta hedef $u \in [0,5]$ ve $nn=10$ için elde edilen YAR görüntüsü.....	55
Şekil 4.13	:Sekiz nokta hedef $u \in [0,5]$ ve $nn=10$ için elde edilen YAR görüntüsü.....	56
Şekil 4.14	:Sekiz nokta hedef $u \in [0,150]$ ve $nn=10$ için elde edilen YAR görüntüsü.....	56
Şekil 4.15	:Sekiz nokta hedef $u \in [0,150]$ ve $nn=40$ için elde edilen YAR görüntüsü.....	57
Şekil 4.16	:Tek nokta hedef için ilk değer 0.2 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı.....	58
Şekil 4.17	:Tek nokta hedef için ilk değer 0.4 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı.....	55
Şekil 4.18	:Dört nokta hedef için ilk değer 0.2 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı.....	59
Şekil 4.19	:Dört nokta hedef için ilk değer 0.4 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı.....	59
Şekil 4.20	:Sekiz nokta hedef için ilk değer 0.2 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı.....	60
Şekil 4.21	:Sekiz nokta hedef için ilk değer 0.4 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı.....	60
Şekil 4.22	:Tek nokta hedef için $nn=10$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı	65
Şekil 4.23	:Tek nokta hedef için $nn=20$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı.....	65
Şekil 4.24	:Tek nokta hedef için $nn=30$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı.....	66
Şekil 4.25	:Tek nokta hedef için $nn=40$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı.....	66
Şekil 4.26	:Tek nokta hedef için GA ile elde edilen hedef konumu (o), gerçek hedef konumu (+).....	67
Şekil 4.27	:Dört nokta hedef için $nn=10$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı.....	70
Şekil 4.28	:Dört nokta hedef için $nn=20$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı.....	70
Şekil 4.29	:Dört nokta hedef için $nn=30$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı.....	71
Şekil 4.30	:Dört nokta hedef için $nn=40$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı.....	71
Şekil 4.31	:Dört nokta hedef için GA ile elde edilen hedef konumu (o), gerçek hedef konumu (+).....	72
Şekil 4.32	:Sekiz nokta hedef için $nn=10$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı.....	74
Şekil 4.33	:Sekiz nokta hedef için $nn=20$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı.....	74
Şekil 4.34	:Sekiz nokta hedef için $nn=30$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı.....	75
Şekil 4.35	:Sekiz nokta hedef için $nn=40$ iken bit ve kromozom sayısına göre genetik algoritma ile elde edilen hata miktarı.....	75

GENETİK ALGORİTMA YARDIMI İLE YAPAY AÇIKLIKLI RADARDA HEDEF KONUMUNUN BELİRLENMESİ

ÖZET

Bu çalışmada bir Yapay Açıklıklı Radar (Y.A.R.) sistemi tarafından toplanan verilerden, optimizasyon yöntemi kullanılarak, görüntü elde edilmesi problemi üzerinde durulmuştur. Literatürde yapılan çalışmalarda menzil ve çapraz menzil çözünürlüklerinin iyileştirilmesi problemi sıkça karşımıza çıkmaktadır. Y.A.R. larda menzil ve çapraz menzil çözünürlükleri, yapay açıklık uzunlukları, örnek sayısı ve darbe süresi gibi parametrelere bağlıdır. Bu yöntemlerde radar alıcısında ölçülen işaretin frekans ve konuma bağlı değerleri elde edilerek Fourier uzayının ölçülen işaretlerle doldurulması sağlanır. Bu işlemin ardından iki boyutlu ters Fourier dönüşümü ile hedef uzayındaki hedef dağılımı bilgisinin oluşturulması amaçlanır. Elde edilen bu görüntülerdeki çözünürlüğün iyileştirilmesi Fourier uzayının yeterince sıklıkla ve geniş frekans bandı içinde doldurulmasına bağlıdır.

Bu çalışmada Fourier uzayının yeterince doldurulamadığı yani eksik bilgi alındığı durumlarda hedef üzerinde elde edilen önbilgiler kullanılarak, bilinmeyen hedef dağılımına ait parametreler, parametre kestirim yöntemine dayalı optimizasyon algoritmaları ile incelenmiştir.

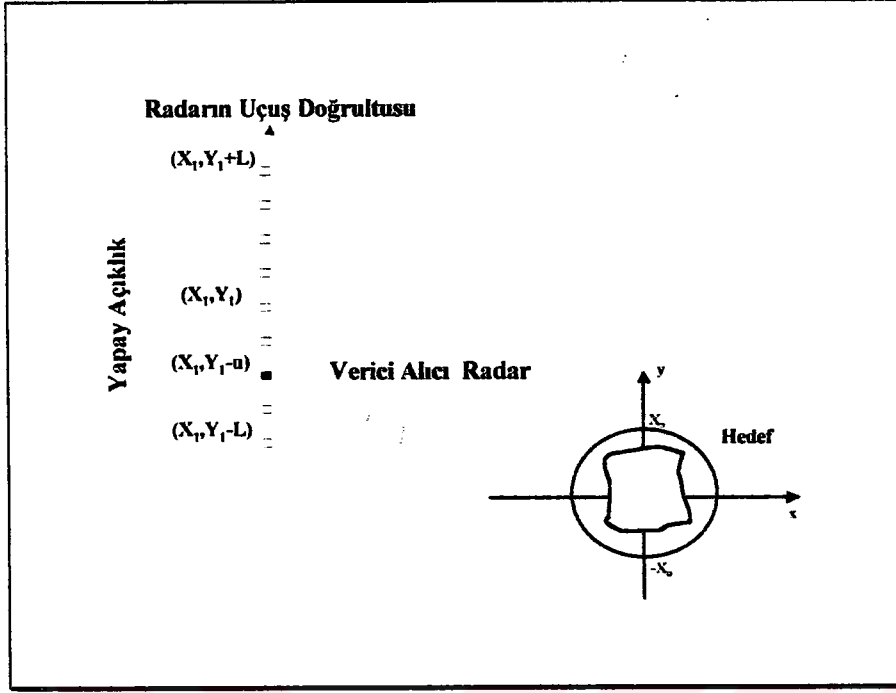
Bu incelemede, parametre kestiriminde kullanılan optimizasyon yöntemi için yeni gelişmekte olan Genetik Algoritma (G.A.), Y.A.R. görüntüleme problemine uygulanmıştır. Y.A.R. görüntüleme geometrisinde gözönüne alınan noktasal hedef dağılımında hedef konumları optimizasyon parametresi olarak gözönüne alınmıştır. Global optimizasyon yöntemi olan G.A. yanında lokal optimizasyon yöntemlerinden olan Hooke Jeeves yöntemi de hedef konumlarının minimum hata ile kestirimi probleminde incelenmiştir.

Bu çalışmada, incelenen Y.A.R. görüntüleme geometrisi Şekil 1' de verilmiştir. Bu geometride görüntülenecek hedef alanı (x,y) koordinat sisteminin merkezine yerleştirilmiş X_0 yarıçaplı bir diskler. Radarın ışıma diagramının küresel olduğu varsayılırsa, bu durumda (x,y) deki bir nokta hedef saçıcısı tarafından yansıtılan işaretin gidiş geliş yolu boyunca olan faz gecikmesi (round trip phase delay) $2k\sqrt{(X_1 - x)^2 + (Y_1 + u - y)^2}$ dir. Böylece kaydedilen toplam yansıyan işaret $u \in [0, L]$ yapay açıklığı boyunca (1) bağıntısıyla ifade edilir.

$$s(u, w) = \iint dx dy f(x, y) \exp[j2k\sqrt{(X_1 - x)^2 + (Y_1 + u - y)^2}] \quad (1)$$

Burada $f(x,y)$ cismin yansıma fonksiyonudur. Ayrıca (1) bağıntısının sağ tarafında görülen küresel dalganın, düzlemsel dalgaların ifadesiyle ayrıştırılabilirliği gözönüne alınırsa; Elde

edilecek bağıntının her iki tarafının u ya göre uzaysal Fourier transformu alınarak (2) bağıntısını elde ederiz.



Şekil 1. Yapay Açıklıklı Radar Görüntüleme Geometrisi.

$$F(k_x, k_y) = \exp[-j(k_x X_1 + k_y Y_1)] S(k_y, w) \quad (2)$$

Burada, $k_x = \sqrt{4k^2 - k_u^2}$, $k_y = k_u$ alınmıştır.

$s(u, w)$ nın u ya göre uzaysal Fourier transformu $S(k_u, w)$, $f(x, y)$ nin (x, y) ye göre uzaysal Fourier transformu $F(k_x, k_y)$ dir.

(2) bağıntısı ile elde edilen $F(k_x, k_y)$ nin iki boyutlu ters Fourier değerleri hedefin yansıtma fonksiyonu olan $f(x, y)$ değerlerini verecektir.

Bölüm 2' de Fourier uzayının yeterince doldurulmadığı yani eksik bilgi alındığı durumlarda, kullanılan optimizasyon yöntemlerinden önce Hooke Jeeves lokal optimizasyon algoritması ve ardından Genetik Algoritma yöntemi incelenmiştir. Bu algoritmaların parametre değerlerinin, yöntemin başarısı üzerindeki etkileride ayrıca incelenmiştir. Uygulanan Hooke Jeeves lokal optimizasyon algoritmasında ilk değer atama ve adım aralığı parametreleri, global optimizasyon yöntemi olan G.A. ise kromozom sayısı, bit sayısı ve iterasyon sayısı parametreleri gözönüne alınmıştır.

Bölüm 3' de G.A.' nın Y.A.R. ters dönüşüm bağıntısına uygulanması incelenmiştir. G.A.' da optimize edilecek parametre olarak nokta hedeflerin konumları seçilmiştir. Nokta hedeflerin sayısının ve merkezlerinin yerinin bilindiği kabulü altında optimizasyon işleminde minimize edilecek hata terimi (3) ile tanımlanmıştır.

$$e = \sum_{i=1}^m \{ \text{Re}[s_i^m(u, w)] - \text{Re}[s_i^c(u, w)] \}^2 + \sum_{i=1}^m \{ \text{Im}[s_i^m(u, w)] - \text{Im}[s_i^c(u, w)] \}^2 \quad (3)$$

Burada $s_i^m(u, w)$ ölçülen cisim yansıma fonksiyonu, $s_i^c(u, w)$ ise optimizasyon adımları sırasında belirlenen hedef konum kestirimleri için hesaplanan cisim yansıma fonksiyonudur.

Bölüm 4'te, seçilen örnek noktasal hedef dağılımları için Fourier domeni ters dönüşüm algoritması ile Hooke Jeeves ve G.A. optimizasyon yöntemlerinin uygulaması gerçekleştirilmiştir. Bunlara ilişkin elde edilen sonuçlar ve görüntüler bu bölümde verilmektedir. Sonuçlar incelendiğinde eksik bilgi durumunda Fourier domeni algoritmasının, sonuçları belirlemede yetersiz kaldığı, Hooke Jeeves lokal optimizasyon algoritmasının başarısının ilk değer seçimi ve arama adımı büyüklüğüne bağımlı olduğu, G.A. nın ise ilk değer seçiminden bağımsız olarak, yapay açıklığın büyük olmadığı durumlarda bile gerçek hedef dağılımına yakınsayan sonuçlar üretebildiği görülmüştür.



DETERMINATION OF THE TARGET POSITION BY THE SYNTHETIC APERTURE RADAR USING A GENETIC ALGORITHM

SUMMARY

In this study, the problem of image data acquisition gathered by the means of optimization method from the gathered by the S.A.R. (Synthetic Aperture Radar) system is presented We frequently meet the enhancement problem of range and cross range resolutions in the literature and the studies made. In S.A.R. systems the range and cross range resolutions depend upon the synthetic aperture length and pulse duration parameters. In these methods, the frequency and location values of the signal measured on the radar receiver are obtained and the Fourier space is filled with measured values. After this operation by means of two dimensional inverse Fourier transform, the target distribution information is formed in the target space. It is based on the filling of the Fourier space in a sufficiently wide frequency band to improve the resolution of these images. In this study, unknown target distribution are investigated by optimization algorithms depending on parameter estimation method in cases when the Fourier space is not sufficiently filled by using the pre knowledge on the target.

A global optimization method;Genetic Algorithm (G.A), which is used for parameter estimation is applied at S.A.R. imaging problem. The target locations which are considered in S.A.R. imaging geometry are regarded as optimization parameters.

In addition to G.A. (Global Optimization) method, the Hooke Jeeves method which is one of the local optimization methods is investigated in the problem of estimating the target locations with minimum error.

In this study, the S.A.R. imaging system being investigated is given in Figure 1. In this geometry, the target range (x,y) to be imaged is a disc with radius of X_0 in the center of the coordinate system. Considering the propagation diagram of the radar is spherical, the round trip phase delay, which is the phase delay of the signal reflected at a target distributer at (x,y) is $2k\sqrt{(X_1 - x)^2 + (Y_1 + u - y)^2}$. Therefore the recorded total signal along the synthetic aperture $u \in [0, L]$ is obtained by the equation (1).

$$s(u, w) = \iint dx dy f(x, y) \exp \left[j 2 k \sqrt{(X_1 - x)^2 + (Y_1 + u - y)^2} \right] \quad (1)$$

Here $f(x,y)$ is the reflection function of the object. Also considering the spherical wave seen at the right hand side of equation (1) can be decomposed of the plane wave equations, the equation (2) is obtained by taking the spatial Fourier transform of the both sides of determined equation.

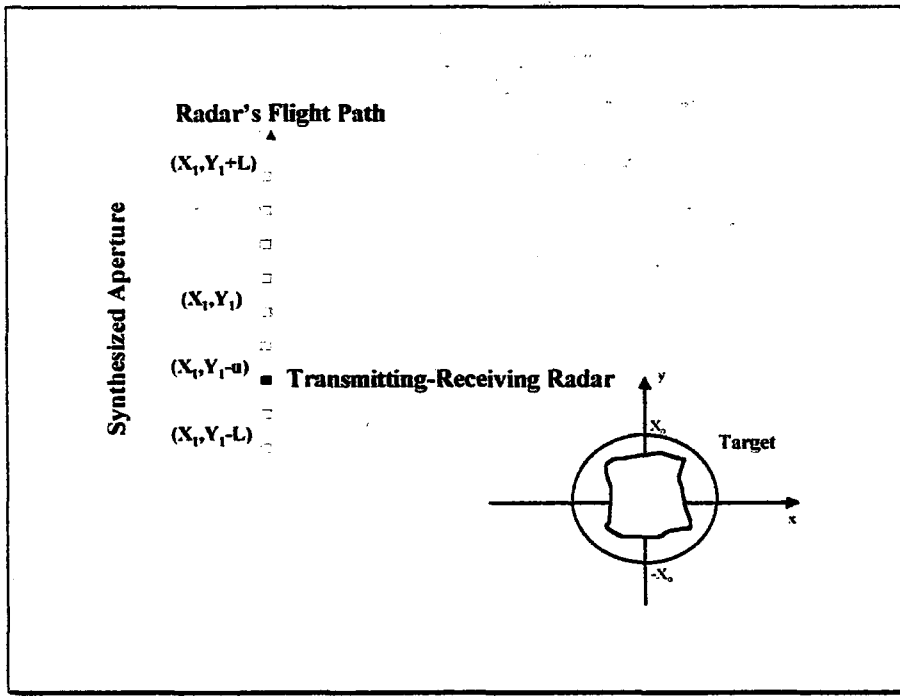


Figure 1. S.A.R. Imaging Geometry.

$$F(k_x, k_y) = \exp[-j(k_x X_t + k_y Y_t)] S(k_y, w) \quad (2)$$

$$\text{where } k_x = \sqrt{4k^2 - k_u^2}, k_y = k_u$$

The spatial Fourier transform of $s(u, w)$ with respect to u is $S(k_u, w)$ and, the spatial Fourier transform of $f(x, y)$ with respect to (x, y) is $F(k_x, k_y)$.

The two dimensional inverse Fourier values of $F(k_x, k_y)$ obtained by (2) will yield the $f(x, y)$ values which is the reflection function of the target.

In section 2, In cases in which the Fourier space is not sufficiently filled, in other words in the presence of missing information, the Hooke Jeeves local optimization and the Genetic Algorithm method is utilized. The influences of the parameter values of these algorithms on the success of the methods is also investigated.

For the application of Hooke Jeeves local optimization algorithm, first value assignment and step gap parameters and for the application of the G.A., chromozon number, bit number and number of iteration parameters are considered.

In part three, the application of genetic algorithm on the inverse transformation equation is handled. In the genetic algorithm and Hooke Jeeves, the positions of point targets are selected as the parameter to be optimized. Accepting that positions of the point targets is unknown, the error statement to be minimized is defined by (3).

$$e = \sum_{i=1}^m \left\{ \text{Re}[s_i^m(u, w)] - \text{Re}[s_i^c(u, w)] \right\}^2 + \sum_{i=1}^m \left\{ \text{Im}[s_i^m(u, w)] - \text{Im}[s_i^c(u, w)] \right\}^2 \quad (3)$$

where $s_i^m(u, w)$ is the measured reflection function of the object. $s_i^c(u, w)$ is the reflection function calculated for target location estimations defined during the optimization steps and nn is the total number of sample points.

In part four, the applications of Fourier inverse transform algorithm and Hooke Jeeves, G.A. optimization methods are implemented for the selected sample point scatterer target distributions. The relevant results and images are given in this section. When the results are considered, in the case of missing information, it is seen that the Fourier domain algorithm can not sufficiently obtain the results, the success of Hooke Jeeves local optimization algorithm depends upon the first value selection and step width. Being independent of the first value selection a G.A. can yield results that converge to the real target distribution even in cases where the synthetic aperture is not large.



1.GİRİŞ.

1.1 GİRİŞ VE TEZİN AMACI.

Görüntüleme radarları, hareketli bir sistem üzerine yerleştirilmiş anten ile incelenen hedef alanının belirlendiği Yapay Açıklıklı Radarlar (YAR) şeklinde olabildiği gibi , hareketli bir hedefi sabit bir radar anteni ile incelenmesi şeklinde de (ISAR) olabilir.

Radar çözünürlüğü (resolüsyon), menzil(range, horizontal) ve çapraz menzil (cross-range) doğrultusunda ifade edilir.Menzil çözünürlüğü gönderilen işaretin bant genişliği ile ters orantılıdır. Bant genişliği arttıkça menzil çözünürlüğü iyileşir.Çapraz menzil çözünürlüğü ise antenin çapraz menzil doğrultusundaki demet genişliği küçüldükçe artar. Demet genişliği ise anten açıklığı veya radar frekansı arttıkça küçülmektedir [1-4] . Radar anteninin hareketli bir platform üzerinde yer değiştirmesi ile her konumda elde edilen işaretlerin toplanması sonucu daha büyük anten genişliği sentezlemek mümkündür. Bu durum yapay açıklıklı radar olarak isimlendirilir. Bu şekilde çapraz menzil çözünürlüğünü arttırmak mümkündür.

Yapay açıklıklı radarlar için çeşitli çalışma modları vermek mümkündür [4,5].Spotlight modunda görüntülenecek hedef alanı sürekli olarak aydınlatılacak şekilde anten demeti yönlendirilir. Stripmap modunda, anten yönlendirilmesi uçuş hattı boyunca sabit olarak bırakılır. Hareket doğrultusuna paralel hedef şeriti boyunca hareketli anten ayak izi ile tarama yapılır. Bu şekilde görüntüleme geometrisi broadside veya antenin uçuş doğrultusunun normaline göre ilerde veya geride odaklandığı squinted görüntüleme geometrisi şeklinde iki farklı stripmap modu tanımlanabilir. Tarama modunda ise hareket doğrultusuna göre herhangi bir açıda olan hedef alan şeridi anten demeti ile taranır. Bu mod ek işlemler ve işleme kargaşası getirdiği için diğer iki mod en yaygın kullanılanlardır. Spotlight mod, bir veya daha fazla bölgesel alanda daha duyarlı çözünürlükte veri toplamak için, stripmap mod ise büyük bölgelerin kaba çözünürlükte incelenmesinde daha etkilidir. Hedefin hareketli, antenin ise sabit olduğu veri toplama yapısını ifade eden ters YAR

(ISAR) modu, hedef ile alıcı anten arasındaki göreceli konum ve hareket gözönüne alındığında spotlight moda benzerdir.

Bu tez çalışmasında YAR tarafından toplanan verilerden optimizasyon yöntemi kullanılarak görüntü elde edilmesi problemi üzerinde durulmuştur. Literatürde kullanılan çoğu yöntemde çapraz çözünürlüğün iyileştirilmesi için yapay açıklığın büyütülmesinin gerektiği görülür [6,7]. Tomografik yöntemlere dayanan YAR görüntüleme algoritmalarında genellikle Fourier domen dönüşüm yöntemleri uygulanmaktadır . Bu yöntemlerde radar alıcısında ölçülen işaretin frekans ve konuma bağlı değerleri elde edilerek Fourier uzayının ölçülen işaretlerle doldurulması sağlanır. Bu işlemin ardından iki boyutlu ters Fourier dönüşümü ile hedef uzayındaki hedef dağılımı bilgisinin oluşturulması amaçlanır. Elde edilen bu görüntülerdeki çözünürlüğün iyileştirilmesi Fourier uzayının yeterince sıklıkla ve geniş frekans bandı içinde doldurulmasına bağlıdır . Bu durum ise alınan frekans örnek sayısına ve yapay açıklık boyunun büyük tutulmasına bağlıdır [8,9,10].

Hedefe ait frekans aralığı ve yapay açıklık için yeterli ölçümlerin yapılamadığı durumlarda Fourier uzayı doldurulamıyacağından radar görüntüsünde iyi bir çözünürlük elde edilemeyecektir. Eksik bilginin olduğu böylesi durumlarda hedefe ait hedef merkezinin yeri ve nokta hedef sayısı gibi ön bilgilerin kullanılması gerekebilir. Bu bilgilerin elde edilebildiği durumlarda YAR görüntüleme problemi noktasal hedeflerin konumlarının belirlenmesi problemine indirgenecektir. Bu konum bilgileri bulunması istenilen parametreler olarak gözönüne alınırsa bu problem bir parametre kestirim problemi haline alacaktır. Bu durumda tanımlanan hatayı minimize edecek şekilde parametre kestirimi bir optimizasyon yöntemiyle verilebilir.

Bu çalışmada optimizasyon işleminde kullanılacak hata tanımı için optimizasyon adımları sırasında belirlenen hedef konumları için hesaplanan anten konumlarındaki yansıyan işaret değerleriyle gerçek ölçüm değerleri arasındaki fark kullanılmıştır. Bu hatayı minimize edecek şekilde lokal veya global optimizasyon algoritmaları kullanılabilir. Verilen hata tanımına ve seçilen ilk değerlerin doğruluğuna bağlı olarak görüntüleme algoritmasının başarısı için optimizasyon yönteminin seçimi önemlidir. İlk değer seçiminde yeterli doğruluğun sağlanamadığı durumlarda global optimizasyon yöntemlerinin kullanılması gerekecektir. Bu çalışmada böylesi

durumlarda YAR görüntülerini elde edebilmek için global optimizasyon algoritmalarından olan genetik algoritma (GA) , noktasal hedeflerin konumlarının belirlenmesi için kullanılmıştır. Lokal optimizasyon algoritmalarından Hooke Jeeves Algoritmasının bu durumdaki başarısı da ayrıca incelenmiştir.

Optimizasyon yöntemleri, kullanılan hata fonksiyonlarına ait bilgilere göre üç guruba ayrılabilir:

- (1) Doğrudan yöntemler: Türev bilgisi gerektirmeyen yöntemlerdir. Başlıcaları, rastlantısal arama (random search), döner koordinatlar (rotating coordinates), eşlenik yönler (conjugate directions), Hooke Jeeves, simpleks veya polytope olarak isimlendirilmiştir.
- (2) Birinci mertebe yöntemler: Birinci mertebeden türev hesabını gerektiren yöntemlerdir. Başlıcaları en dik iniş (steepest descend), eşlenik gradyanlar (conjugate gradients), quasi newton yöntemleridir.
- (3) İkinci mertebe yöntemler: Birinci ve ikinci mertebe türev bilgisi gerektirmektedir. Newton yöntemleri olarak isimlendirilirler.

Yukarıdaki yöntemlerin tümü literatürde geniş biçimde incelenmiştir.[11] Türeve dayalı yöntemler optimizasyon problemlerinde zaman zaman başarıyla uygulanmalarına rağmen, bu tür yöntemlerin doğrudan uygulanması hesaplama bakımından ağır bir yük getirmektedir. Bu tür yöntemlerin, değişik biçimleri önerilmiş olmakla beraber, bölgesel (lokal) minimum problemi ortaya çıkmaktadır. Buna çözüm olarak birden fazla başlangıç çözümlerinin denenmesi yolu kullanılmaktadır. Teorik olarak, bir optimizasyon probleminin iyi tanımlı amaç fonksiyonu için en iyi çözümü, sadece tüm çözümlerin oluşturulması ve değerlendirilmesi ile mümkün olabilir. Bununla beraber, çözüm uzayının büyüklüğü, en iyi çözümün uygun bir sürede elde edilmesini güçleştirmektedir.

Bir optimizasyon işleminin başlangıcında ilk parametre değerleri oluşturulur. Her adımda parametre değerleri az miktarda değiştirilerek, amaç fonksiyonunun değeri hesaplanır ve sadece amaç fonksiyonunun değerini iyileştiren parametre değerleri kabul edilir. Bu tür yaklaşımın olumsuz tarafı, uygun başlangıç değerleri

kullanılmadığında en iyi çözümün elde edilememesi ve çözümün bölgesel olmasıdır. Çözümün en iyi olması için global optimizasyon yöntemleri kullanılır.

Pratikte çok sayıda mühendislik uygulamaları global optimizasyon problemi olarak formüle edilebilir. Global optimizasyon probleminde ilgilenilen bölgede çok sayıda bölgesel minimum vardır. Sistem tasarımında olası çözümler bir alt bölgede sınırlandırılmıştır. Amaç fonksiyonunun bu alt bölgede çok sayıda minimumu olabilir. Global iyileştirmenin amacı, bu alt bölgede amaç fonksiyonunun en küçük değerini verecek çözümü bulmaktır. Global optimizasyon, bu alt bölgede bir bölgesel minimum yerine, en küçük bölgesel minimumu (global) bulmayı amaçlar [12,13].

1975' te Holland yukarıdakilerden çok farklı olan bir diğer optimizasyon yöntemi tanıttı. Bu doğal evrimde gözlenen sürece benzeyen ve GA olarak bilinen bir mekanizmadır [14]. Bu optimizasyon tekniğine benzer: kılavuzlanmış rastgele teknikler olarak sınıflandırılan yumuşatılmış simulasyon, evrimsel stratejiler ve evrimsel programlama teknikleri bulunmaktadır [15,16,17,18]. GA tamamen farklı bir optimizasyon yöntemi olarak çalışır ve işaret işleme için daha fazla esneklik sağlayan bir yöntemdir. Uygulamadaki basitlik nedeni ile, GA yapay zeka sistemlerinin tasarlanmasında da bir optimizasyon aracı olarak kullanılmaktadır [19,20]. GA'nın kullanışlılığına ve bu konuda geniş literatüre rağmen, GA' nın işaret işlemede kullanımı sayıca azdır, fakat bir uygulama alanı olarak GA'nın işaret işlemede potansiyel kullanımı geniştir.

Genetik Algoritmalar modelleme, işaret karakterizasyonu için önemli bir tekniktir. Geçmişte çözülemez olduğu düşünülen pekçok problem, hesaplama gücünün artmasıyla, şimdi kolaylıkla çözülebilir duruma gelmiştir. Bu alandaki başarılı uygulamalar arasında sonar ve radar sistemleri için sonlu birim basamak cevabı olarak modellenen zaman gecikmeli kestirim [21]; doğrusal kestirimli kodlama ile ses kodlama [22]; ses ve görüntü kodlama ve tanıma için dalgacıklar (wavelet) [23]; görüntü sıkıştırma ve tanıma sistemleri için frakteller [24]; aktif gürültü kontrolü için gecikmeli X filtreler [25,26]; mikrodalga devre tasarımı ve elektromanyetik alanda uygulamaları sayılabilir [27,28,29,30,31].

İşaret işleme için etkin bir model geliştirmek kolay değildir. Modelin doğruluğu genellikle arama bölgesinde optimize edilen bir dizi değişken veya parametre

dizisiyle kontrol edilir. Bazen optimizasyon algoritmasının kendi amacına ulaşmakta yetersiz kaldığı sınırlı bir bölgede çalışmak zorunda kalınılabılır. Tipik bir örnek, kararlılık için sonsuz impuls tepkili bir filtrenin katsayıları üzerindeki sınırlamalardır. Amaç fonksiyonu doğrusal yada değil, eşitlik yada eşitsizlik, düzgün yada değil, amaç fonksiyonu genellikle probleme dönük olduğundan çözüm tek değildir.

Bu tez çalışmasında, klasik YAR algoritması seçilen sabit noktasal hedef dağılımları için uygulanmış ve buna karşılık düşen görüntüler elde edilmiştir. Elde edilen görüntülerde, yapay açıklık boyu ve örnek sayısı (yapay açıklık boyunca gerçek anten konum sayısı) bilgilerinin sınırlı (eksik) kullanıldığı durumlarda, nokta hedeflerin konumlarının yeterli doğrulukta belirlenemediği görülmüştür. Bu durumda, YAR sisteminde hedef konumunun (nokta hedeflerin konumları) belirlenmesi için optimizasyon algoritmasına dayanan yeni bir yaklaşım sunulmuştur. Önerilen yöntemde, Fourier domeninde klasik YAR algoritması ile konumları belirlenemeyen noktasal hedeflerin belirlenebilmesi için seçilen optimizasyon algoritmaları (GA ve Hooke Jeeves) YAR görüntüleme problemine uygulanmıştır. YAR görüntüleme geometrisinde gözönüne alınan noktasal hedef dağılımında, hedef konumları (noktasal hedeflerin açısı ve menzili) optimizasyon parametresi olarak alınarak, global optimizasyon yöntemi olan GA ve bunun yanında performanslarının karşılaştırılması amacıyla lokal optimizasyon yöntemi olan Hooke Jeeves algoritması uygulanarak hedef konumlarının minimum hatayla kestirimine çalışılmıştır. Minimize edilecek hata fonksiyonu, radar anteninde ölçülen işaret ile optimize edilen hedef dağılımı için anten ölçüm noktalarında hesaplanan işaret arasındaki farkın karesi ile belirlenmektedir. Kullanılan iki farklı optimizasyon algoritmasından biri hata fonksiyonunun yerel minimumlarını bulan Hooke Jeeves optimizasyon algoritması ve diğeri belirlenen hata fonksiyonunun global minimumunu bulan GA'dır. Sonuçlar üzerinde, optimizasyon parametrelerinin değişimlerinin etkileride incelenmiştir. Bu parametreler, Hooke Jeeves algoritmasında adım büyüklüğü, GA'da iterasyon sayısı, bit sayısı ve kromozom sayısıdır. Ayrıca, elde edilen sonuçlar klasik YAR algoritmasının, eksik bilgi nedeniyle yetersiz kaldığı durumlarda, GA ile hedef konumlarının minimum hata ile gerçek hedef konumlarına çok yakın olarak elde edildiği gösterilmiştir.

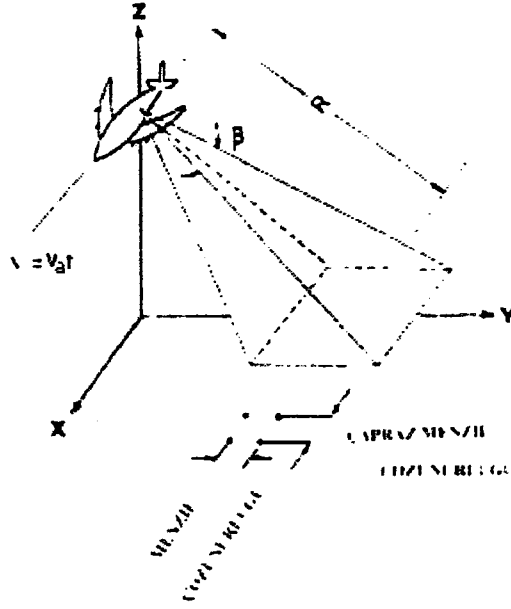
2. YAPAY AÇIKLIKLI RADAR VE OPTİMİZASYON YÖNTEMLERİ

2.1 YAPAY AÇIKLIKLI RADAR KAVRAMI, GEOMETRİSİ VE TERS DÖNÜŞÜM BAĞINTISI

2.1.1 Yapay Açıklıklı Radar Prensibi

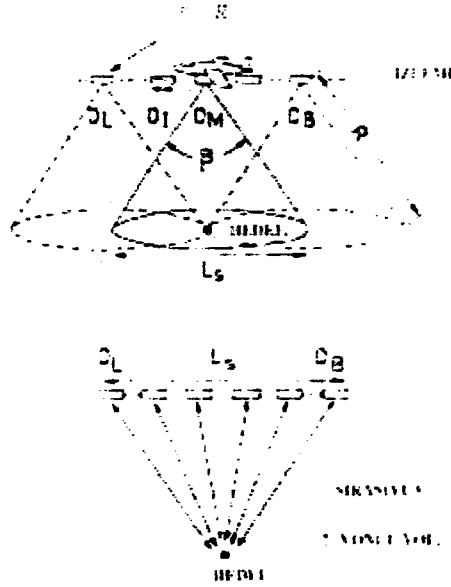
Yapay açıklıklı radar (Synthetic Aperture Radar) mikrodalga bölgesinde kullanılan bir uzaktan algılama ve görüntüleme sistemidir. İlk deneysel YAR uygulaması 1953 yılında İllionis Üniversitesinden bir grup tarafından Florida'da gerçekleştirilmiş ve 3 cm dalga boyunda çalışılmıştır [6]. Bu uygulamada radar, yeryüzünde belirli bir şeridi süpürmüş ve bu arada gözlenen noktadan anten demeti içerisinde kalındığı süre boyunca yansıyan işaret toplanarak işlenmiştir. Bu uyumlu integrasyon süresince radar uçuş menzili, yapay açıklık boyunu belirlemektedir. Radar bir uçakta olabileceği gibi uydu üzerinde de olabilir. Uzaydaki ilk uygulama 1978 yılında SEASAT uydusuyla gerçekleştirilmiştir. Bugün YAR uygulamaları çok amaçlıdır. Geniş yüzey taramalarında gerçek açıklıklı radar sistemi kullanılmaktadır. Özel hedeflere yönelindiğinde ise yüksek çözünürlüklü YAR sistemi büyük önem kazanmaktadır.

Yüksek çözünürlüklü görüntüleme radarlarında yüksek menzil (across-track) ve yüksek çapraz menzil (along-track) çözünürlüklerinin her ikisinde işlenmesi için iki boyutlu görüntü üreten YAR sistemine ihtiyaç vardır. Böyle bir sistem, yandan bakış YAR (side-looking SAR) görüntüleme sistemi Şekil 2.1'de verilmiştir. Yandan bakış YAR sistemi incelendiğinde, radarın ışın demeti yönündeki çözünürlüğe menzil çözünürlüğü, radarın uçuş doğrultusuna paralel olan çözünürlüğe çapraz menzil çözünürlüğü denir.



Şekil 2.1 Yandan bakış yapay açıklıklı radar prensibi

Yapay açıklıklı radar, yapay açıklıklı anten dizisi prensibine dayanır. Yapay açıklıklı anten dizisinin fonksiyonunu anlamak için Şekil 2.1’de gösterilen yandan bakış YAR görüntüleme prensiplerini ve geometrisini incelemek gerekir. Radar, uçuş doğrultusuna paralel taralı alanı aydınlatmak için yatay doğrultulu bir anten sayesinde ışıma yaymaktadır. Çapraz menzil boyutunda genellikle dar olan antenin ışın demeti döndürülmekte ve uçağın ileri doğru hareketiyle birlikte aydınlatılan bölgenin ayrıntılı bir biçimde incelenmesi sağlanmaktadır. Radar sistemi, aydınlatılan bölgeden saçılan yansımanın iki boyutlu görüntüsünü üretir. Bu görüntünün bir boyutu görüş menzili pozisyonuyla yani menzil çözünürlüğü, ikinci boyutu ise izleme koordinatı pozisyonuyla yani çapraz menzil çözünürlüğüyle ilişkilidir. Yapay açıklıklı radarın sabit bir hızla yatay doğrultuda hareket ettiğini varsayalım. Belirlenecek hedefe yandan bakış doğrultusunda radar işareti gönderilir. Şekil 2.2’de yapay açıklıklı radarın izleme hareketi gösterilmiştir. Radar yatay uçuş doğrultusu boyunca hareket ettiğinde düzenli aralıklarla darbe işareti yayar ve belirlenecek hedeften yansıyan işareti kaydeder. Böylece alınan işaretlerin hem genliği hem fazı kaydedilir. Kaydedilen bu bilgi uygun bir biçimde işlenir.



Şekil 2.2 Yapay açıklıklı radar görüntüleme prensibi

Şekil 2.3’de gösterildiği gibi uzun doğrusal bir gerçek anten dizisinin karakteristiklerini referans olarak kullandığımızı farzedelim. Böyle bir anten dizisi, uzun doğru bir hat boyunca uygun yerlere yerleştirilmiş belirli bir şekle sahip N adet anten elemanından oluşturulmuştur. Anten dizisi verici durumunda ise dizinin her bir elemanı aynı anda beslenir. Benzer şekilde anten dizisi alıcı olarak kullanıldığında, yansıyan işaret her bir elemana aynı anda ulaşır. Dizinin ışıma diyagramı dizideki bir elemanın ışıma diyagramının dizi faktörüyle çarpımına eşittir. Dizi faktörünün yarım güç ışın demeti genişliği (half power beam width);

$$\beta = \frac{\lambda}{L} \text{ (Radyan)} \quad (2.1)$$

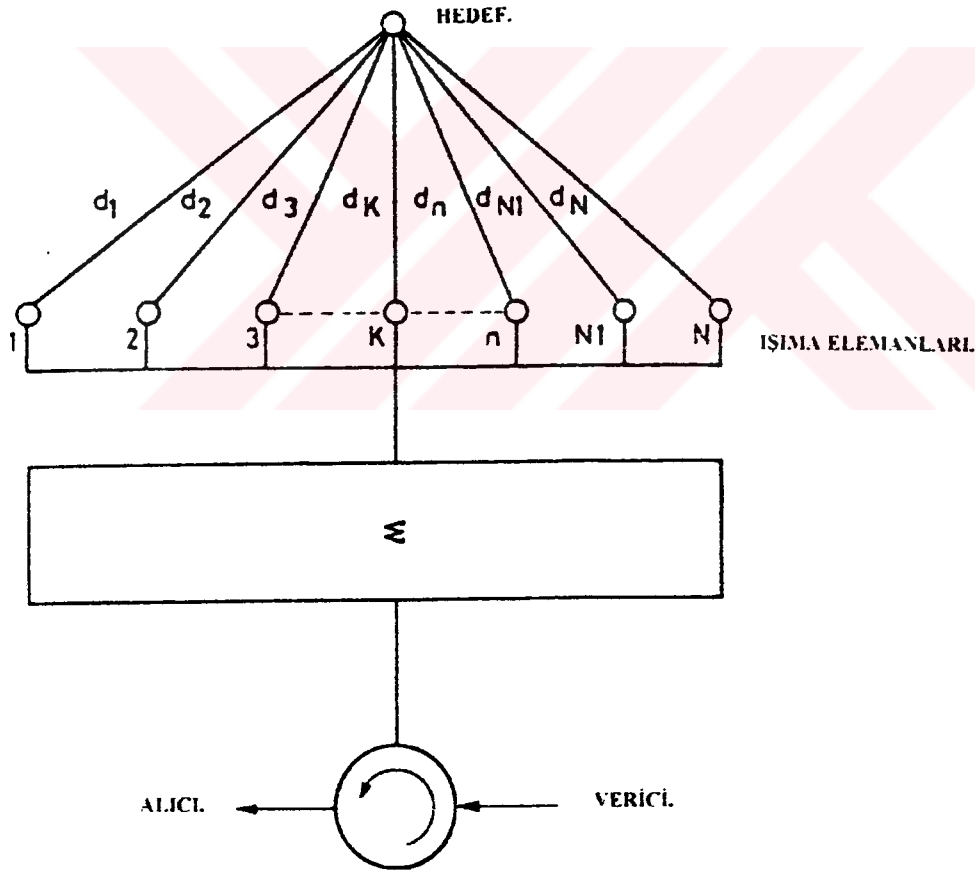
bağıntısıyla ifade edilir. Bu bağıntıda L gerçek anten dizisinin uzunluğu, λ ise dalga boyudur. Gerçek açıklıklı radarın çapraz menzil çözünürlüğü ise;

$$\rho_a = \beta R = \frac{\lambda R}{L} \quad (2.2)$$

bağıntısıyla ifade edilir. Burada R hedefin uzaklığıdır. Örneğin; $R=200$ km., $\lambda=2$ cm. (x-band), $\rho_a=10$ m. ise anten açıklığı $L=400$ m. olacaktır. Örnekten de görülebileceği gibi gerçek açıklıklı radar kullanıldığında yüzlerce hatta binlerce metre uzunlukta antene ihtiyaç

duyulur. Kesinlikle böyle bir sistem pratik kullanımda uygun değildir. YAR sistemi bu kısıtlamaları ortadan kaldıracak bir çözüm sağlar.

Yapay açıklıklı anten ile gerçek anten dizisi arasında bazı benzerlikler ve farklar vardır. En önemli fark yapay açıklıklı anten dizisinde tek bir verici elemanı vardır. Radarın hedefe göre hareketiyle bu verici elemanın konumu D_B pozisyonundan D_L pozisyonuna değiştirilir. Şekil 2.2’de gösterildiği gibi tek bir eleman her bir konumda hedefe işaret gönderir ve yansıyan işareti alır. Verici eleman L_s uzunluğunca hareket ettirildikten sonra gerçek doğrusal dizide olduğu gibi kaydedilen işaretler birleştirilir. Sonuç olarak, gerçek doğrusal diziye benzer biçimde etkin bir anten dizisi oluşturulur. Bu yolla oluşturulan antene yapay açıklıklı anten, böyle bir anten prensibini kullanan radara da yapay açıklıklı radar denir.



Şekil 2.3 Uzun doğrusal anten dizisi

Yapay açıklıklı anten dizisinin açisal çözünürlüğünün, gerçek açıklıklı anten dizisinin çözünürlüğüne göre iki kat daha iyi olması her iki anten dizisi arasındaki en önemli farktır [7]. Bunun nedeni doğrusal dizinin seçileceği işaretin alınış işlemi esnasında sağlanır. İşlem esnasında, doğrusal dizinin her bir elemanı tarafından alınan işaretin fazındaki farklar, antenin ışıma diyagramını verir. Yapay açıklıklı anten dizisinde tek bir eleman işareti hem iletip hem de almaktadır. Bu sebeple, yansıyan işarete iki yönlü faz kayması ortaya çıkar. Tek yönlü durum için yarım güç ışın demeti genişliği $\beta = \lambda / L$ iken, ikiyeönlü durum için yarım güç ışın demeti genişliği;

$$\beta_s = \frac{\lambda}{2L_s} \quad (2.3)$$

bağıntısıyla ifade edilir. Burada, β_s yapay açıklıklı antenin etkin ışın demeti genişliği, L_s yapay açıklıklı antenin uzunluğudur. Şekil 2.2'den görüldüğü gibi,

$$L_s = \frac{\lambda}{D} R \quad (2.4)$$

bağıntısı yazılabilir. D antenin yatay açıklığıdır, R hedefin uzaklığıdır. R görüş mesafesindeki yapay açıklıklı radarın çapraz menzil çözünürlüğü ise;

$$\rho_a = R \beta_s = R \frac{\lambda}{2L_s} \quad (2.5)$$

bağıntısıyla ifade edilir. Yukarıdaki (2.4) ve (2.5) bağıntılarının birleştirilmesiyle çapraz menzil çözünürlüğü;

$$\rho_a = \frac{R \lambda}{2} \frac{D}{\lambda R} = \frac{D}{2} \quad (2.6)$$

ifadesi elde edilir. Yapay açıklıklı antenin çapraz menzil çözünürlüğü dalga boyu ve hedefin uzaklığından bağımsız olup, antenin yapay açıklığı ile doğrusal olarak değişir.

2.1.2 Yapay Açıklıklı Radar Görüntüleme Sistemi, Geometrisi ve Ters Dönüşüm Bağıntısı

Radar ve hedefin aynı düzlem üzerinde bulunduğu iki boyutlu YAR görüntüleme sistemini gözönüne alalım. Bu durumda sırasıyla uzaysal ve uzaysal frekans domenleri diye tanımlanan (x,y) ve (k_x,k_y) domenleri kullanırız. Uzaysal domendeki işareti $f(x,y)$, bunun uzaysal fourier transformunu $F(k_x,k_y)$ ile göstereyim. Radar tarafından aydınlatılan cisim alanı uzaysal domenin merkezindeki X_0 yarıçaplı bir diskin içine yerleştirilmiştir. x koordinatı menzili tanımlamak için kullanılır ve y koordinatı ise dikey eksenini belirtir. Ortam genelde homojen hava veya sudur. Bu ortamdaki propagasyon hızı c dir. Bu homojen ortamda zamana ait w frekansıyla yürüyen bir dalganın dalga sayısı; $k=w/c$ dir.

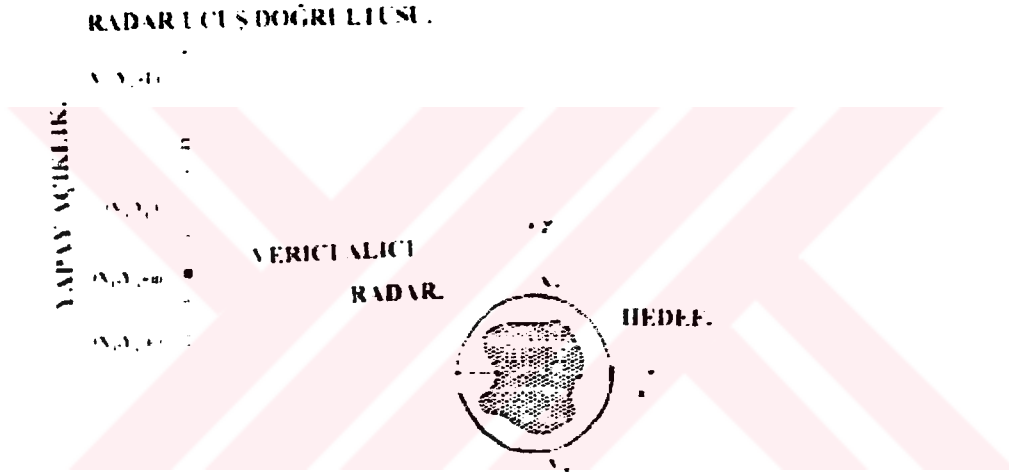
Radar işareti darbeleri işaretin herhangi bir tipi olabilir. Örneğin, radar işareti bir sinüzoidal taşıyıcıyla modüle edilmiş bir dikdörtgen darbeyse, radar işaretinin Fourier transformu taşıyıcı etrafına yerleştirilmiş bir sinc fonksiyonudur. Böylece, radar işareti bir frekans bandına sahip olmaktadır. Klasik YAR görüntüleme yöntemi bu frekans bandındaki incelenen frekans bileşenine sahip yansıyan işaretin uyumlu işleme yöntemiyle seçilip, bu işaretin ölçülen işaret olarak kullanılmasına dayanır. Hedef, radarı taşıyan uçağın anteni tarafından aydınlatılır. Bunun neticesinde hedeften yansıyan işaret uçağın anteni tarafından toplanır ve uygun bir dedektörden geçirilir. Daha sonra analog/sayısal konvertörler sayesinde aynı fazlı ve 90° faz farklı zamanda işaret örnekleri elde edilir. Uygun dedektörün iki kanalında örneklenmiş bilgidan zamandaki kompleks işaret oluşturulur. Bu işaretin gerçek ve sanal kısımları sırasıyla aynı fazlı ve 90° faz farklı kanallardaki örneklerdir. Daha sonra, örneklenmiş bilginin ayrık Fourier transformu hesaplanır ve bu Fourier işleminde radar işareti faz ve genliğine göre normalize edilir.

YAR görüntüleme sisteminde (x,y) düzleminde $x=X_1$ hattı boyunca radarın hareket ettiğini varsayalım. Radar (x,y) düzleminde $u \in [-L, +L]$ açıklıkta (X_1, Y_1+u) koordinatında transmisyon yapar ve yine aynı koordinatta işareti alır. (X_1, Y_1) bilinen sabitlerdir. Y_1 bakış modu dikey koordinatıdır. $Y_1=0$ ise, broadside moduna karşılık düşer. Kartezyen koordinatları (X_1, Y_1) , (θ_p, R_p) polar koordinatlarıyla ifade edilebilir. Görüntülenecek hedef alanı koordinat sisteminin merkezine yerleştirilmiş X_0 yarıçaplı bir disktr. (Şekil 2.4)

Radarın ışıma diyagramının küresel olduğu varsayılırsa, bu durumda (x,y) deki bir nokta saçıcısı tarafından yansıtılan işaretin gidiş geliş yolu boyunca olan faz gecikmesi $2k\sqrt{(X_1 - x)^2 + (Y_1 + u - y)^2}$ dir. Radar alıcısında kaydedilen toplam yansıyan işaret.

$$s(u, w) = \iint dx dy f(x, y) \exp \left[j2k\sqrt{(X_1 - x)^2 + (Y_1 + u - y)^2} \right] \quad (2.7)$$

Bu toplam yansıyan işaretin uzaysal domendeki genlik fonksiyonu bastırılmıştır. Mesafenin cismin boyutundan ve yapay açıklıktan önemli derecede büyük olduğu durumlardaki yapay açıklıklı radar görüntülemelerinde uzaysal ve uzaysal frekans domeni genlik fonksiyonları önemli değildir [32]. Kaydedilen toplam yansıyan işaretin yalnızca fazı ile ilgilenilecektir.



Şekil 2.4 Yapay açıklıklı radar görüntüleme sisteminin geometrisi

Burada $f(x,y)$ cismin yansıma fonksiyonudur. Ayrıca (2.7) bağıntısının sağ tarafında görülen küresel dalga, düzlemsel dalgaların ifadesiyle ayrıştırılabilirliği göz önüne alınırsa [33]; bu ayrıştırma (2.8) bağıntısıyla ifade edilebilir.

$$\exp \left[j2k\sqrt{(X_1 - x)^2 + (Y_1 + u - y)^2} \right] = \int dk_x \exp \left[j\sqrt{4k^2 - k_x^2} (X_1 - x) + jk_x(Y_1 + u - y) \right] \quad (2.8)$$

[32].(Genlik fonksiyonu $\frac{1}{\sqrt{4k^2 - k_u^2}}$ bastırılmıştır.)

YAR görüntülemesinde kullanılan, görüntüleme modlarından ikisi spotlight ve stripmap modlarıdır. Bu YAR modları için mevcut ters dönüşüm yöntemlerinde Fresnel veya düzlemsel dalga yaklaşımları vardır [34]. (2.8) bağıntısının sağ tarafındaki bu ayrıştırma işlemi gerçekte (x,y,u) 'nun lineer olmayan faz fonksiyonunu, (x,y,u) 'nun lineer faz fonksiyonuna dönüştürür. Bu ifadeyi (2.7) bağıntısındaki küresel dalga fonksiyonunun yerine yerleştirilir ve gerekli düzenlemeler yapılırsa aşağıdaki ifadeler elde edilir [8].

$$\begin{aligned}
 s(u, w) &= \int_{-2k}^{2k} \exp(j\sqrt{4k^2 - k_u^2} X_1 + jk_u Y_1) \exp(jk_u u) \\
 &\times \left[\iint f(x, y) \exp[-j(\sqrt{4k^2 - k_u^2} x + k_u y)] dx dy \right] dk_u \\
 &= \int_{-2k}^{2k} \exp(j\sqrt{4k^2 - k_u^2} X_1 + jk_u Y_1) F(\sqrt{4k^2 - k_u^2}, k_u) \exp(jk_u u) dk_u. \quad (2.10)
 \end{aligned}$$

$s(u, w)$ 'nin u 'ya göre uzaysal Fourier transformu $S(k_u, w)$ ile gösterilir. (2.10) bağıntısının her iki tarafının u 'ya göre uzaysal Fourier transformunu alırsak (2.11) bağıntısını elde ederiz.

$$S(k_u, w) = \exp(j\sqrt{4k^2 - k_u^2} X_1 + jk_u Y_1) F(\sqrt{4k^2 - k_u^2}, k_u). \quad (2.11)$$

Sonuç olarak, (2.11) bağıntısından aşağıdaki dönüşümü yazabiliriz.

$$F(\sqrt{4k^2 - k_u^2}, k_u) = \exp\left[-j(\sqrt{4k^2 - k_u^2} X_1 + k_u Y_1)\right] S(k_u, w) \quad (2.12)$$

burada, $k_x = \sqrt{4k^2 - k_u^2}$, $k_y = k_u$ alınmıştır.

(2.12) eşitliğinde görülen $F(k_x, k_y)$ fonksiyonu $f(x, y)$ fonksiyonunun iki boyutlu Fourier dönüşümü olduğundan, hedef yansıma fonksiyonu olan $f(x, y)$ 'yi, $F(k_x, k_y)$ 'nin iki boyutlu ters Fourier dönüşümü alınarak belirlenir.

(2.9) eşitliği aşağıdaki gibi yazılabilir [9].

$$s(u, w) = \iint f(x, y) IFT_{(k_u)} \left[\exp\left[j\sqrt{4k^2 - k_u^2} (X_1 - x) + jk_u (Y_1 - y)\right] \right] dx dy \quad (2.13)$$

Burada $IFT_{(k_u)}$, k_u 'ya göre uzaysal ters Fourier dönüşümünü gösterir. Daha sonra Doppler analizi (u 'ya göre Fourier dönüşümü) $s(u, w)$ 'da sonlu yapay açıklık boyu ($L < \infty$) gözönüne alınarak gerçekleştirilir. $u \in [-L, L]$ açıklığı boyunca (2.13) bağıntısının her iki tarafının u 'ya göre Fourier dönüşümü alındığında (2.14) bağıntısını verir.

$$S(k_u, w) = \iint f(x, y) \left[\exp\left[j\sqrt{4k^2 - k_u^2} (X_1 - x) + jk_u (Y_1 - y)\right] * 2L \frac{\sin(Lk_u)}{Lk_u} \right] dx dy \quad (2.14)$$

Burada $*$ işareti k_u domeninde konvolüsyon (katlama) işlemini gösterir. Dikkat edilirse açıklık transfer fonksiyonu $2L \text{sinc}(Lk_u)$ olan (yani $2L[\sin(Lk_u)/Lk_u]$) bir alçak geçiren filtredir. Böylece, faz modüleli fonksiyonun $\exp\left[j\sqrt{4k^2 - k_u^2} (X_1 - x) + jk_u (Y_1 - y)\right]$, $u \in [-L, L]$ aralığı dışına düştüğü anlık frekans değerlerinde, yukarda bahsedilen alçak geçiren filtrenin katlama işlemiyle faz modüleli dalganın bileşenleri filtrelenir.

(x, y) 'de nokta hedefler için, (2.15) bağıntısı sayesinde k_u domeninde faz modülasyonlu fonksiyonun anlık frekans değerleri bulunabilir.

$$\frac{\partial}{\partial k_u} \left[\sqrt{4k^2 - k_u^2} (X_1 - x) + k_u (Y_1 - y) \right] = \frac{-k_u}{\sqrt{4k^2 - k_u^2}} (X_1 - x) + (Y_1 - y) \quad (2.15)$$

Böylece, aşağıdaki (2.16) eşitsizliği kullanılarak faz modülasyonlu dalganın filtrelenmediği

durumdaki k_u değerlerinin aralığı bulunur.

$$-L \leq \frac{-k_u}{\sqrt{4k^2 - k_u^2}}(X_1 - x) + (Y_1 - y) \leq L \quad (2.16)$$

(2.16) eşitsizliğinden bu aralığın çözümlenmesiyle k_u domeninde $(x, y, L \ll X_1)$, (2.17) bağıntısı ile ifade edilen band elde edilir.

$$\Omega(y) \equiv \left[2k \sin \theta_p - 2k \frac{L-y}{R_p} \cos^2 \theta_p \times 2k \sin \theta_p + 2k \frac{L+y}{R_p} \cos^2 \theta_p \right] \quad (2.17)$$

Bu band nokta hedeflerin çapraz menzil (cross range) fonksiyonudur.

Hedef alanındaki bütün nokta hedeflerden yansıyan işaretlerin destek bandlarının birleşimi, k_u domeninde $s(u, w)$ YAR işaretinin destek bantıdır. Bu band Ω_s ile gösterilir.

$$\begin{aligned} \Omega_s &= \bigcup_{|y| \leq X_0} \Omega(y) = \bigcup_{|y| \leq X_0} \left[2k \sin \theta_p - 2k \frac{L-y}{R_p} \cos^2 \theta_p \times 2k \sin \theta_p + 2k \frac{L+y}{R_p} \cos^2 \theta_p \right] \\ &= \left[2k \sin \theta_p - 2k \frac{L-X_0}{R_p} \cos^2 \theta_p \times 2k \sin \theta_p + 2k \frac{L+X_0}{R_p} \cos^2 \theta_p \right] \end{aligned} \quad (2.18)$$

Bu bağıntıdan görüldüğü gibi YAR işareti bir band geçiren işaret olup uzaysal merkez frekansı $k_u = 2k \sin \theta_p$ dir. Band genişliği ise $4k(L + X_0 / R_p) \cos^2 \theta_p$ dir. Böylece, örtüşme olmaksızın band geçiren işaretin işlenmesi ve örneklenmesi için; u domenindeki kullanılan örnekleme aralığı (2.19) eşitsizliğini sağlamalıdır. Ayrıca band geçiren YAR işaretinin örnekleri ayrık Fourier dönüşümleri alınmadan önce $\exp(-j2k \sin \theta_p u)$ ifadesi ile çarpılarak band geçiren işaret alçak geçiren işarete dönüştürülmesi gerekir.

$$\Delta_u \leq 2\pi \frac{R_p}{4k(L + X_0) \cos^2 \theta_p} \quad (2.19)$$

2.1.3 Yapay Açıklıklı Radarda Yeniden Oluşturma Algoritması ve Akış Diyagramı

Bu bölümde YAR görüntüleme algoritmasının işlem adımları verilmiştir. Algoritmada radar işaretinin verilen bir frekansı için aşağıdaki adımlar uygulanır.

Adım 1: u domeninde toplanan bilgi (R_1, θ_p) polar koordinat sistemi için $\exp(-j2k \sin \theta_p u)$ terimiyle çarpılır [6]. Bu işlem band geçiren bilgiyi alçak geçiren hale dönüştürür. (Temel band işareti elde edilir.) Bu adım örtüşmeden sakınmak için ikinci adımdaki Ayırık Fourier Transformu işleminden önce gereklidir.

Adım 2: u domeninde alçak geçiren duruma dönüştürülen bilginin Ayırık Fourier Transformu alınır.

Adım 3: (2.12) bağıntısının sağ tarafında tanımlanan Doppler bilgisinin ters dönüşümü hesaplanır.

Adım 4: Demodülasyon öncesi band geçiren bölgenin merkezi yaklaşık olarak, k_c merkez frekansındaki dalga sayısı gözönüne alınarak;

$$(k_{x_0}, k_{y_0}) = (2k_c \cos \theta_1, 2k_c \sin \theta_p) \text{ dir.}$$

Sonuç bilgisi $(k_x - k_{x_0}, k_y - k_{y_0})$ için elde edilir. Bu işlem (2.12) bağıntısının ters dönüşüm işlemini oluşturur ve (k_{x_0}, k_{y_0}) 'nın çıkarılması sayesinde frekans bilgisini alçak geçiren uzaysal frekans domenine kaydırır. Bu kaydırma işlemi (x, y) domeninde bir doğrusal faz fonksiyonu üretir. Kaymanın bütün zamandaki frekans değerleri için aynı olması gerekir. Aksi halde yeniden oluşturulan uzaysal domen bilgisi istenmeyen sıkıştırma ve genleşmelere sebep olur.

Adım 5: Düzensiz olarak yerleştirilen bilginin interpolasyonu alınarak (aradeğerleme) düzgün ızgaraya (üniform grid) yerleştirilir.

Adım 6: Radar işaretinin mevcut frekansları için 1 den 5'e kadar adımlar tekrar edilir.

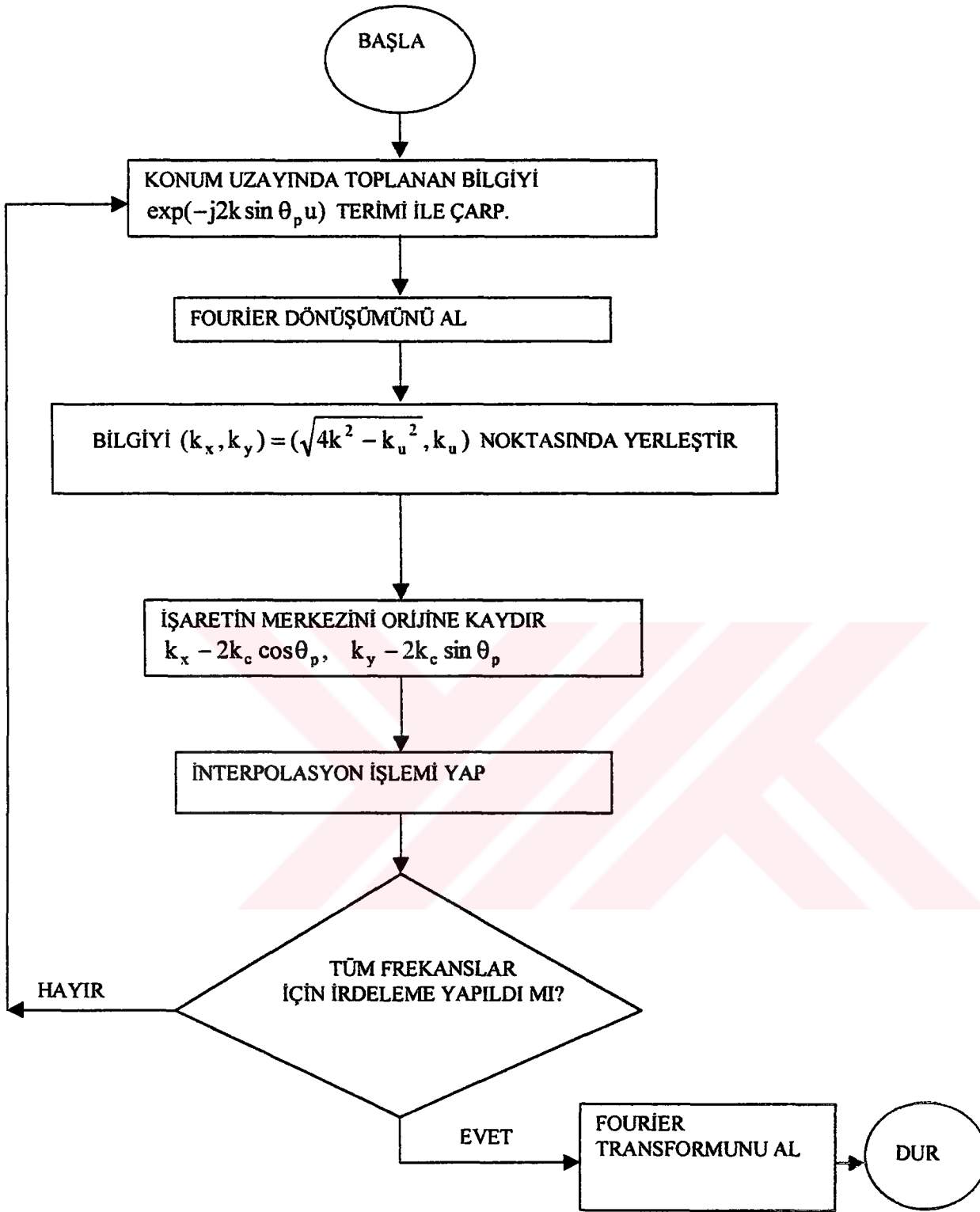
Adım 7: Düzgün ızgaradaki bilginin iki boyutlu Ayırık Fourier Transformunun tersi alınır. Sonuç bilgisinin genliği elde edilir ve görüntülenir.

Bu adımlara ilişkin uygulanacak akış diyagramı Şekil 2.5'de verilmiştir.

(2.18) ve (2.19) ifadelerinden görüldüğü gibi işlenebilen işaret band genişliği yapay açıklık boyuna bağlıdır. Buna bağlı olarak, yapay açıklık içindeki örnekleme aralığı içinde bir üst sınır söz konusudur. Bu nedenle klasik YAR yönteminde elde edilen görüntünün iyiliği seçilen yapay açıklık boyuna ve örnekleme aralığına bağlı olacaktır. Yapay açıklık boyunun ve örnekleme sayısının büyük tutulamadığı durumlarda görüntü kalitesinin düşeceği (çözünürlük hücrelerinin büyüyeceği) gözönünde bulundurulmalıdır.

Bu çalışmada, yapay açıklık boyunun küçük olduğu ve örnekleme sayısının yetersiz olduğu, yani eksik bilgi eldesi durumunda klasik YAR yönteminin yeterince iyi işleyemediği, ölçüm verilerini daha iyi değerlendirip hedef dağılımının kestirimindeki hatayı düşürmek için optimizasyon yöntemleri önerilmiştir. Bu yöntemler tanımlanan hatayı minimize edecek şekilde bir ters dönüşüm (hedef dağılımı kestirimi) gerçekleyerek hedef dağılımının daha iyi belirlenmesini amaçlamaktadır.





Şekil 2.5 Yapay açıklıklı radarda yeniden oluşturma algoritmasının akış diyagramı

2.2 OPTİMİZASYON YÖNTEMLERİ

Optimizasyon problemi genel anlamda bir fonksiyonun en küçük değerini belirleme işlemi şeklinde değerlendirilebilir. Bu fonksiyon bir hata fonksiyonu olarak verildiğinde tanımlanan hatayı en aza indirme işlemi gerçekleştirilecektir. Tanımlanan hata fonksiyonu bağlı olduğu parametre değerlerinin değiştirilmesi ile optimizasyon işlemi gerçekleştirilebilir. Bu parametrelerin nasıl değiştirileceği seçilen optimizasyon yöntemine göre farklılıklar gösterir. Optimizasyon yönteminin seçiminde hata fonksiyonunun tanımı önemlidir. Bu fonksiyona bağlı olarak belirlenecek minimum hata değeri optimizasyon yöntemine göre lokal veya global minimum olabilmektedir. Literatürde çok sayıda optimizasyon yöntemi incelenmiştir [35].

Bu tez çalışmasında klasik YAR görüntüleme algoritmasında eksik bilgi durumunda elde edilen görüntülerdeki hatanın minimize edilerek, noktasal hedef dağılımının düzeltilmesi için optimizasyon yöntemleri önerilmiştir. Hata minimizasyon probleminin doğasına uygun olarak literatürde çeşitli optimizasyon yöntemleri önerilmektedir. Hedef dağılımındaki her noktasal hedefe ait menzil ve açı değerleri optimizasyon algoritmasının belirlenecek parametrelerini oluşturmaktadır. Bu nedenle optimizasyon algoritmasında belirlenecek parametre sayısı doğrudan nokta hedef sayısına bağlı kalmaktadır. Nokta hedef sayısının yüksek olduğu durumlarda özellikle deterministik optimizasyon yöntemlerindeki başarı düşmektedir. Literatürde Hooke Jeeves gibi deterministik, Monte Carlo veya GA gibi deterministik olmayan optimizasyon yöntemleri bulunmaktadır.

Monte Carlo yöntemi, karmaşık ve çok elemanlı sistemlerin özelliklerinin hesaplanmasında güçlü bir yöntem olmakla birlikte, deterministik olmayan işlemlerde kullanılır. Monte Carlo yöntemi, deterministik sayısal bir yöntem değildir. Yöntemde, rastgele sayılar üretilerek doğal rastgele süreçler canlandırılmaya çalışılır. Bu yöntem, hesaplamada kullanılan programın pek çok kez çalıştırılması ve bir çok denemenin sonucunda ulaşılan ortalama çözümleri verir. Bu nedenle çözümler, beklenen değer hakkında yapılan tahminlerin dalgalanmalarını ve de yükselip alçalmalarını da içerir. Olasılıksal bir yaklaşım yapıldığından, tam olarak kesin bir sonuç elde edilmesi imkansızdır. Bu yöntemde, gerçek çözümlere ulaşabilmek için kullandığımız örneklerin sayısı ne kadar yüksekse çözüm için elde ettiğimiz kestirim değerleri, aradığımız gerçek çözüm değerlerine o kadar yakındır. Monte Carlo yönteminde hata boyuttan bağımsızdır ve $1/\sqrt{N}$ ile orantılı olarak azalır[36]. Örneğin, hatayı 1/10 azaltabilmek için örnek sayımız $100 \cdot N$ yani 100 kat artırılmalıdır. Diğer yöntemlerde hata

$\frac{1}{N^{2/d}}$, (d boyut) ile orantılıdır. Bu durumda Monte Carlo yöntemi $d < 4$ için diğer yöntemler kadar verimli olmamaktadır. Ayrıca, Monte Carlo yöntemi örnek uzayında belirlenen örnek aralığı içindeki tüm örneklerin denenmesini gerektirir. Oysaki, sonraki bölümlerde görüleceği gibi yine bir deterministik yöntem olmayan GA seçilen kromozom sayısının sınırladığı örnek değerlerin kullanıldığı rastgele bir süreçtir ve boyuta doğrudan bir bağımlılık söz konusu değildir.

2.2.1. Hooke Jeeves Optimizasyon Algoritması ve Akış Diagramı

Bu yöntem 60 lı yıllarda önerilmesine karşın oldukça etkin bir yöntemdir [37]. Araştırma bir temel nokta etrafında araştırma adımları (exploratory moves) ve arkasından örnek adımları (pattern moves) adı verilen işlemlerden oluşmaktadır. Algoritma şu şekildedir :

Adım 1 b_1 Temel başlangıç noktası ve her x_j , $j= 1,2 , \dots n$ değişken için adım uzunluğu h_j seçilir.

Adım 2 b_1 civarında bir araştırma yapılır. Bunun için şu yol izlenir.

(i) $f(b_1)$ değeri hesaplanır.

(ii) Her bir değişkeni adım uzunluğu ekleyerek sırasıyla değiştirilir. Bunun için önce e_1 , x_1 eksenini yönündeki birim vektör olmak üzere $f(b_1-h_1e_1)$ değeri hesaplanır. Eğer sonuçta bir azalma oluyorsa b_1 yerine $b_1-h_1e_1$ değeri alınır. Azalma olmuyorsa b_1 değiştirilmeden bırakılır ve x_2 deki değişim incelenir. Bu inceleme tüm değişkenler için yapıldıktan sonra yeni temel nokta b_2 elde edilir.

(iii) Eğer $b_2=b_1$ ise araştırma işlemi azaltılmış adım uzunluğu için b_1 civarında tekrar edilir.

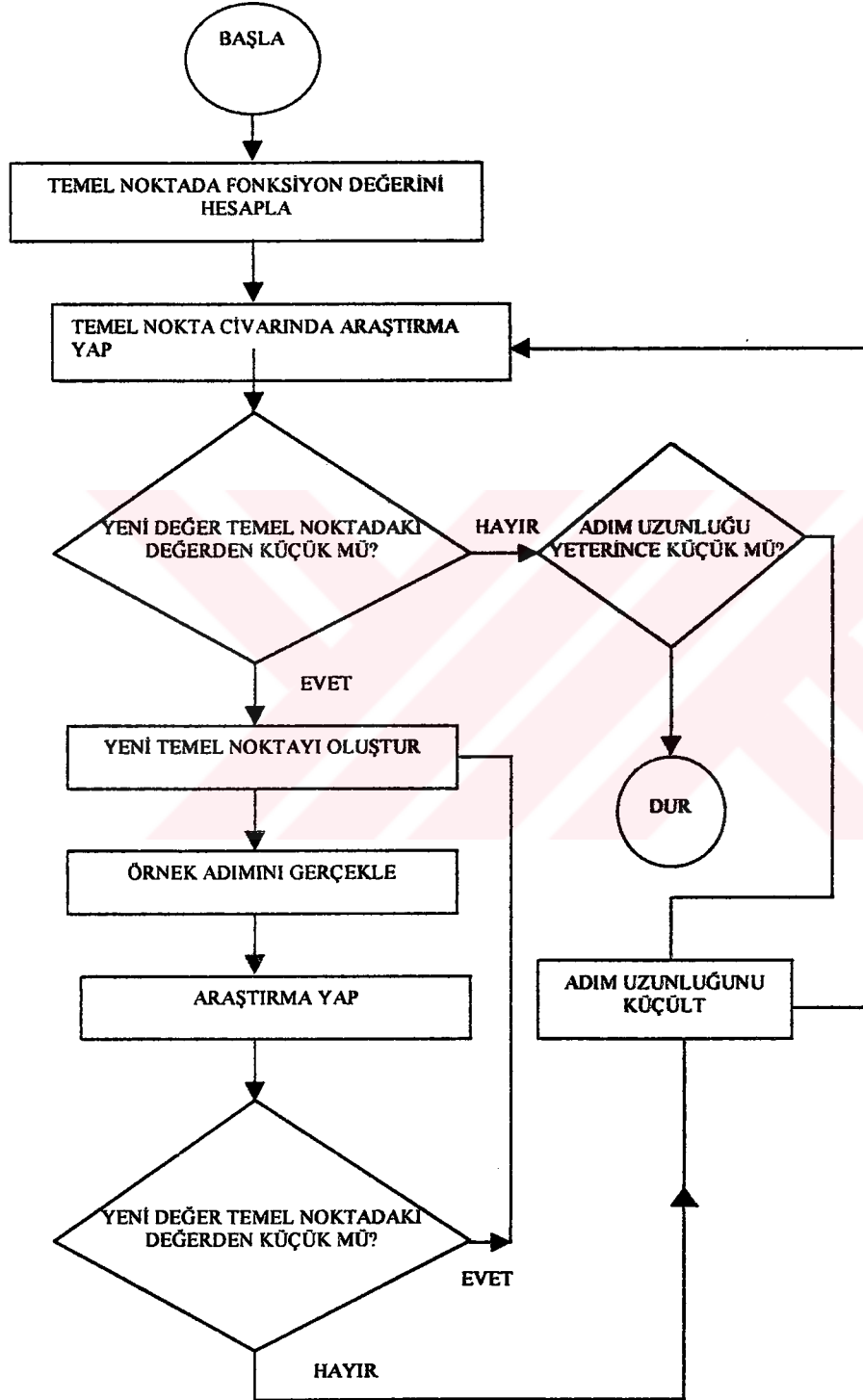
(iv) Eğer $b_2 \neq b_1$ ise örnek adımı yapılır.

Adım 3 Örnek adımında, araştırma adımından elde edilen bilgi kullanılır. Bunun için şu yol izlenir.

(i) $P_1=b_1+2(b_2-b_1)$ (genel olarak $P_i=b_i+2(b_{i+1}-b_i)$) noktasındaki fonksiyon değeri bulunur.

(ii) Keşif hareketlerine P_1 (P_i) civarında devam edilir.

(iii) Eğer (ii) adımıdaki en küçük değer b_2 (b_{i+1}) değerinden daha küçük ise yeni temel nokta b_3 (b_{i+2}) elde edilir. Bu durumda (i) tekrar edilir. Diğer durumda b_2 ($b_i + 1$) civarında araştırma adımına devam edilir.



Şekil 2.6 Hooke Jeeves optimizasyon algoritması akış diyagramı

Adım 4 Adım uzunlukları önceden belirlenen küçük bir değere ulaştığında işlem sona erdirilir.

Algoritmanın akış diagramı Şekil 2.6'da verilmiştir.

2.3 GLOBAL OPTİMAZASYON YÖNTEMİ GENETİK ALGORİTMA

2.3.1. Genetik Algoritma Kavramı ve Diğer Geleneksel Yöntemler Arasındaki Farklar

Bu bölümde, genetik algoritmanın ne olduğu, nereden geldiği gösterilecek, diğer en iyi arama yöntemleri ile karşılaştırılması verilecektir. Algoritmanın işleyişi bir örnek üzerinde açıklanacaktır [38].

Genetik algoritmalar, doğal seçme ve doğal genetik kurallara dayanır. Doğal seçme, doğa koşullarına en fazla uyum sağlamış olan canlının neslini devam ettirmesi, uyum sağlamamış olan türlerin ise elenmesidir. Canlılar nesilden nesile genlerini başka genlerle eşleştirir, değiştirir ve yeni genleri oluştururlar. Genetik algoritmalarda tabiattaki bu iki oluşumu birleştirerek en iyi (optimal noktayı) arar. Bir önceki neslin en uygun elemanlarının parçaları ve “bit” leri kullanılarak yeni neslin üyeleri oluşturulur. Bu üyelerin uygunlukları test edilerek, en iyilerden hareketle yeniden başka elemanlar türetilir. Genetik algoritmalarda olasılığın rolü olmasına karşın, tamamiyle rastgele bir arama metodu değildir. Etkinliği doğadaki canlıların evriminden de görülebilir.

Genetik algoritmalar Michigan Üniversitesinden John Holland ve arkadaşları ile öğrencileri tarafından geliştirilmiştir. Araştırmalarının iki amacı vardı. Birincisi, canlılardaki adaptasyonun yapısını anlamak ve kabaca açıklamak. İkincisi, bu yapıdan hareketle, insan yapısı sistemlere uygulanabilecek algoritma bulmak. Bu yaklaşım daha sonra önemli gelişmelere ışık tutmuştur.

Genetik algoritmalarındaki araştırmaların temel ilgi noktası değişen ortam şartlarına uyum sağlayabilen etkin yapıyı oluşturabilmektir. İleri derecede adaptif (uyumlu) sistemler oluşturulabilirse, pahalıya mal olan yeniden tasarım problemleri azaltılabilir, hatta ortadan kaldırılabilir. Eldeki mevcut sistemler adaptif bir kontrol ile değişen şartlarda yeniden kullanılabilir ve görevlerini daha iyi ve daha uzun zaman yerine getirebilirler.

Genetik algoritmaların esasını temsil eden biyolojik sistemler, dikkatli bir inceleme ile bize yeni ufuklar açabilecek sırlara sahiptir. Bugün için, en iyi adaptasyon doğada var olan adaptasyondur. Buna dayanan genetik algoritmalar, Holland'ın 1975'te yayınladığı "Doğal ve Yapay Sistemlerde Adaptasyon" isimli çalışmasından bu yana başka pek çok makalede ele alınmış, ve etkinliği gösterilmiştir.

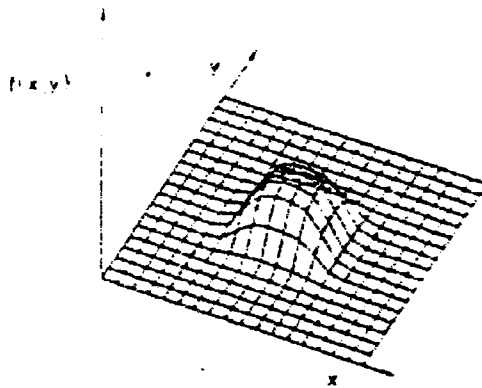
Genetik algoritmalar, basit hesaplamalar gerektirir. Ancak işlemlerin basitliği etkinliğini azaltmaz. Diğer pek çok en iyiyi arama yönteminde görülen, süreklilik, türev bilgisi, tek minimum yada maksimumun varlığı gibi kabuller bu yöntem için gerekli değildir.

Geleneksel en iyiyi arama yöntemlerini üç ana başlık altında toplanabilir:

Bunlar türev hesabına dayanan, nümerik ve rastlantısal arama yöntemleridir. Hesaba dayanan yöntemler bugüne kadar literatürde yoğun bir biçimde incelenmişlerdir. Bunlarda kendi aralarında, dolaylı ve doğrudan olmak üzere ikiye ayrılırlar :

Dolaylı yöntem, amaç fonksiyonunun gradyanının sıfıra eşit alınması sonucu çıkan nonlineer denklem takımının çözülmesi esasına dayanır. Bu Şekil 2.7'de gösterilen yüzey üzerindeki yerel maksimumun n boyutlu yüzeylere genellenmesidir. Aramada da, her yöndeki türevi sıfır olan bu tepe ve çukur olması muhtemel noktaları civarında yapılır.

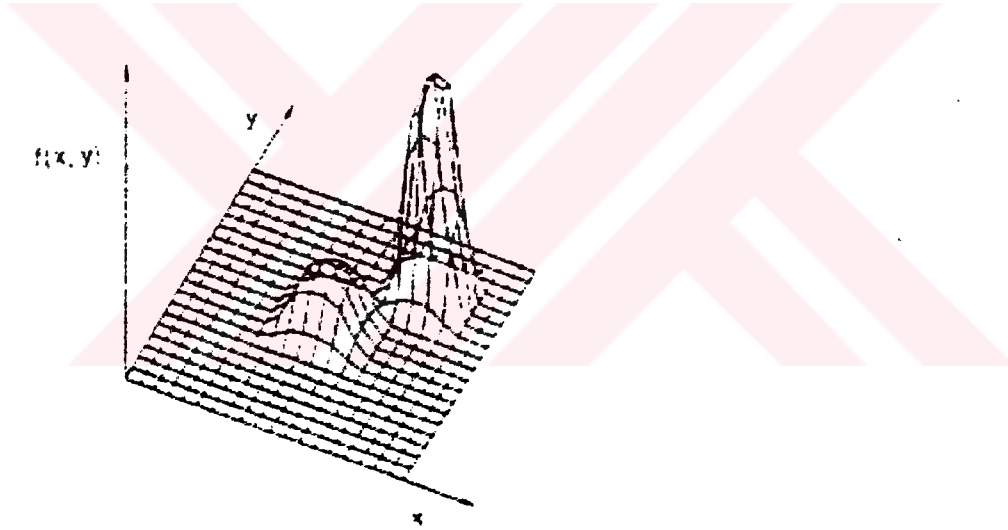
Diğer taraftan, doğrudan arama yönteminde maksimum noktasının yeri hakkında bir tahminde bulunur ve buna yakın herhangi bir noktanın gradyanı, veya tersi yönünde hareket edilerek arama sürdürülür. Buna literatürde (hill climbing), tepe tırmanma adı verilir. Yerel en iyi noktayı bulmak için, mümkün olan en dik eğimli yönde yürünür. Her ne



Şekil 2.7 Tek tepe noktalı uzay.

kadar bu iki arama yöntemi pek çok çalışmada ele alınmışsa da , birkaç basit örnek, bunların zayıf yönlerini göstermeye yeter.

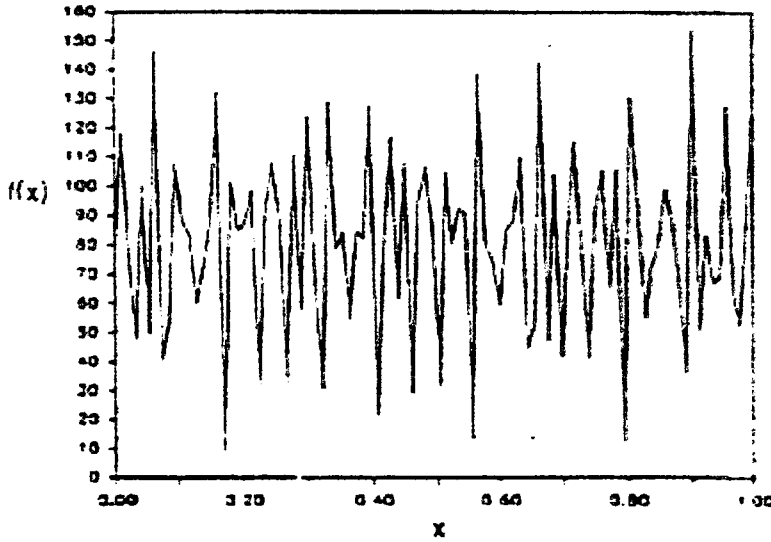
Bu yöntemlerin en önemli eksikliği yerel bakış açısıyla yürütölmeleridir. Aradıkları en iyi nokta, hareket noktasının yakınındadır. Örneğin Şekil 2.7'de gösterilen, uzayın sınırlı bir bölgesi olsun. Bu uzayın daha geniş bir bölgesi de Şekil 2.8'de verilmiş olsun. Bu ikinci şekilden de görölebileceği üzere, alçak tepenin yakınından aramaya başlanırsa, en iyi nokta olarak alçak tepenin üzerinde kalınacaktır. Oysa, bulunacak bu noktadan daha iyi bir nokta vardır ki oda daha yüksek olan ikinci tepe noktasıdır. Alçak tepenin yakınındaki bir noktada başlamak bile esas en iyi noktayı kaybettirecektir. Buna bir çözüm olarak, arama işlemine rastgele seçilmiş başka bir noktadan tekrar başlanabilir. Şekil 2.7'deki gibi tek tepe noktalı fonksiyonlar hesaba dayalı arama yöntemleri için kolay olmasına rağmen Şekil 2.8 deki gibi fonksiyonlarda hangi tepeye tırmanılacağı ayrı bir problemdir.



Şekil 2.8 Çok tepe noktalı uzay.

İkinci önemli eksiklik ise, amaç fonksiyonunun türevlerinin var olmasının gerekliliğidir. Teorik olarak hesabı kolay kuadratik amaç fonksiyonlarının bulunabilmesine karşın, gerçek hayatta karşılaşılanlar teoriden uzaktır. Parça parça süreksizlikler içerebileceği gibi gürültülü ortamlardan elde edildikleri için Şekil 2.9'daki gibi pekçok tepe noktalı olabilir. Bundan da

anlaşılabileceği gibi hesaba dayalı yöntemler ancak sınırlı sayıda uygulamalar için kullanışlıdır, bütün uygulamalar için değildir.



Şekil 2.9 Pratikte karşılaşılan bir amaç fonksiyonu örneği.

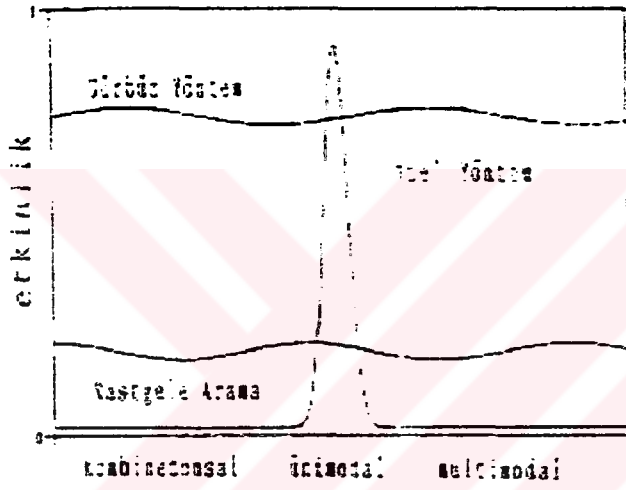
Nümerik yöntemlerin de üzerinde pekçok çalışmalar yapılmıştır. Bu tip arama yöntemlerinde temel düşünce, sınırlı bir arama uzayındaki noktaların amaç fonksiyon değerlerine bakıp ona göre yeni noktaların bulunmasıdır. Ancak bu tip arama yöntemlerinde pratikte arama uzayının geniş olması ve bundan dolayı noktaların birer birer ele alınmasının mümkün olamaması dezavantajlardır.

Rastlantısal arama yöntemleri, diğer arama yöntemlerinin bu zayıf yönlerinden dolayı git gide artan bir ilgiyle incelenmektedir. Bununla beraber, rastlantısal bir yönde tepe tırmanma veya rastgele noktalar seçerek bunların amaç fonksiyonlarını bulma gibi düşünceler rastgele arama yöntemlerinden dışarıda tutulmalıdır. Çünkü etkin değildirler. Genetik algoritmalar, bu gibi rastgele arama yöntemlerinden farklı olarak olasılığı, kodlanmış parametre uzayında işlem yapmada bir araç olarak kullanırlar. Bir arama algoritmasında olasılığı kullanmak ilk önce şaşırtıcı gelebilir ancak doğada bulunan sayısız örneği mevcuttur.

Genetik algoritmalar ile global optimizasyon yöntemlerinden CRS1 ve CRS2 (Controlled Random Search – Kontrollu Rastlantısal Arama) [39] ile karşılaştırsak, bu yöntemlerde parametrelerin kodlanmadığını görebiliriz. Yakın zamanlarda ilgi çeken bir başka yöntem ise Isıl İşlem Benzetimi (Simulated Annealing) olarak geçen yöntemdir. Davis'in 1987 de çıkan

bir kitabında bu yöntemle genetik algoritmalar arasındaki ilişki incelenmektedir [40]. Bu arada önemli bir saptama yapmak gerekirse, rastlantısallığı kullanmak, belli bir mantıktan yoksun arama yapmak demek değildir. Olasılığı kullanmayan global arama yöntemleri de vardır. (Örneğin ızgara arama) Ancak bunlarda da işlem sayısının fazlalığı n boyutlu uzaylarda çözümü bulmayı zorlaştırmaktadır.

Şekil 2.10'da görülebileceği gibi, pekçok geleneksel arama yöntemi küçük bölgelerde iyi sonuç vermektedir. Nümerik veya rastlantısal arama yöntemleri bütün spektrum için yeterli etkinlikte değildir. Gürbüz yöntemler bütün spektrum için yeterli etkinliktedir.



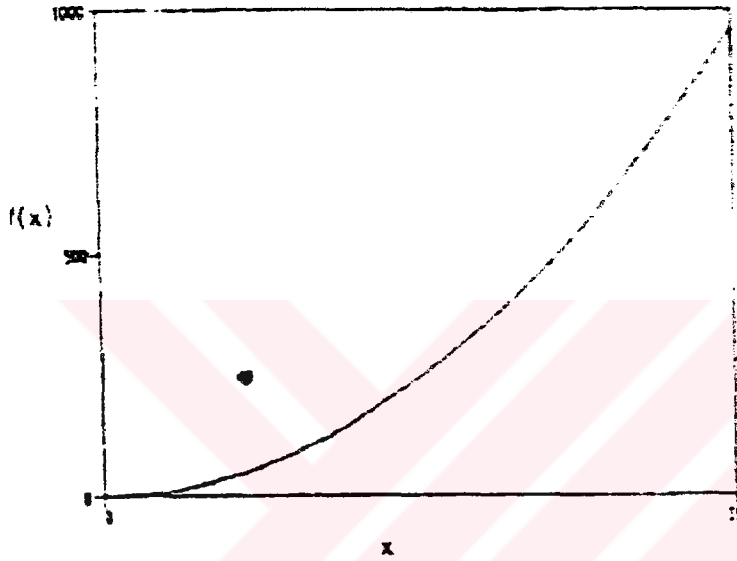
Şekil 2.10 Arama yöntemlerinin etkinliği.

Genetik algoritmaların (G.A) temel özellikleri dört maddede toplanabilir.

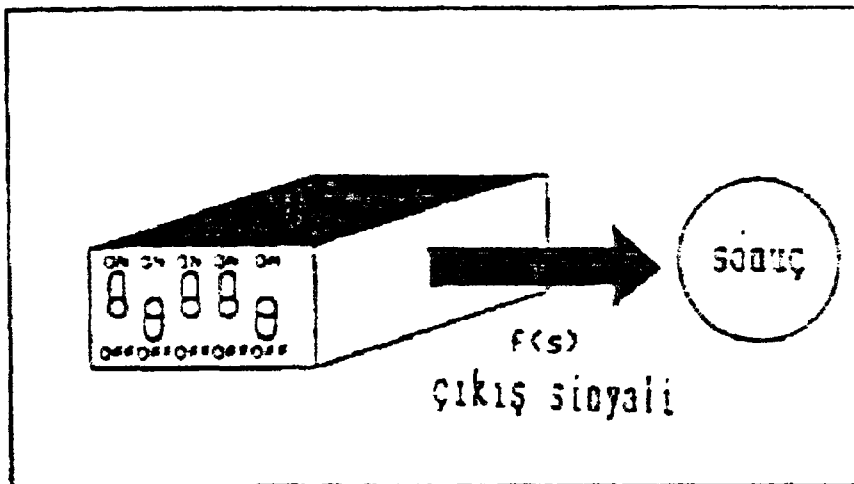
1. G.A.parametrelerin kodlanmış halleriyle çalışır, parametrelerin kendileri ile değil.
2. G.A. tek bir nokta yerine noktalar topluluğundan hareket eder.
3. G.A. türev ve daha farklı bilgileri değil sadece amaç fonksiyon değerlerini kullanır.
4. G.A: deterministik değil, olası kurallara dayanır.

Genetik algoritmalarda parametreler sonlu uzunlukta diziler ile gösterilirler. Örnek olarak Şekil 2.11'de gösterilen (0.31) aralığındaki optimizasyon problemi ele alınsın. İlk olarak x parametresi açık kapalı biçiminde iki farklı değer alan bir değişkenle kodlanır. Daha sonra açık kapalı yerine “bir” ve “sıfır” lar kullanılır.

Sistem Şekil 2.12'de gösterildiği gibi bir kapalı kutu olarak düşünülebilir.



Şekil 2.11. $y=x$ amaç fonksiyonu.



Şekil 2.12. Kapalı kutu örneği.

Kapalı kutu örneğinde beş adet giriş düğmesi ve bir adet f çıkış işareti bulunmaktadır. Amaç fonksiyonu matematiksel olarak, s belli bir düğme kombinasyonu olmak üzere $f=f(s)$ şeklinde yazılabilir. Problemin amacı, bu beş anahtarın mümkün olan en yüksek f değerini elde edecek uygun kombinezonunu bulmaktır. Diğer arama metodları düğmelerin durumları ve bir durumdan diğerine nasıl geçileceğini araştırırken, genetik algoritmalarda önce düğmelerin durumları kodlanır. Bu kodlama beş bitlik sayı dizileriyle yapılabilir.

Tek bir nokta ele alıp, bu noktadan diğer bir noktaya belli bir kural kullanarak geçmek yanlış bir yerel en iyi noktaya yerleşmeye sebep olabilir. GA, tek bir nokta yerine, bir noktalar topluluğundan yararlandığı için, yapılacak ikinci adım bu noktalar topluluğunu belirlemektir. Böylece global en iyi noktaya yakınsama şansı artacaktır. G.A.'da nesil adı verilen bu noktalar topluluğundan , yeni bir nesil, yeni bir noktalar topluluğu elde edilir. Örnekte birey sayısı $n=4$ alınmıştır ki bu sayı GA standartları için küçük bir sayıdır.

01101
11000
01000
10011

Bu dört noktadan hareketle GA kullanarak yeni nesiller bulunur. Bir nokta topluluğu üzerinde paralel çalışmanın, olumlu etkilerinin bulunacağı açıktır.

2.3.2 Basit Genetik Algoritma

Genetik algoritmaların mantığı, bit dizilerini, kopyalamaya ve iki bit dizisinin bazı parçalarının yerlerini değiştirmeye dayanır. Uygulamasının kolaylığı ve etkinliği genetik algoritmalarla çözüm yaklaşımının en önemli ilk üstünlüğüdür.

Bir önceki bölümde genetik algoritmaların, bir dizi topluluğu üzerinde nasıl işlem yaptığı kısaca anlatılmıştı. Kapalı kutu örneğini tekrar ele alırsak, ilk nesilde dört dizi bulunmaktaydı,

01101
11000
01000
10011

Bu dört dizi de, rastgele olarak(0/1 kodlu), bir parçanın 20 kere atılması sonucu oluşturulmuş olsun. Bu aşamadan sonra, ilk nesillerden yeni nesilleri türetmede kullanılacak genetik operatörlerin tanımlanması gerekir.

Pek çok uygulamada iyi sonuçlar veren basit genetik algoritmada üç operatör bulunur. [38,41]. Bunlar:

- 1 Üreme
- 2 Çaprazlama
- 3 Mutasyon

operatörleridir.

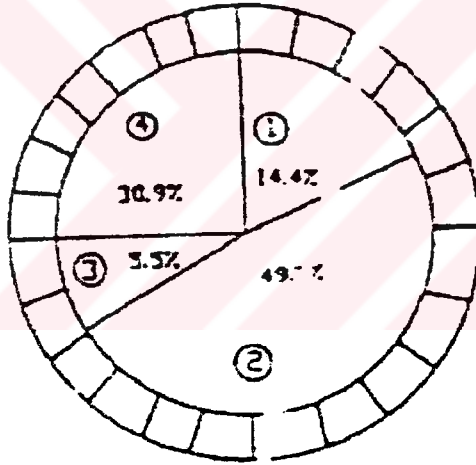
Üreme işlemi sırasında, sayıları (noktaları) temsil eden bit dizileri, amaç fonksiyon değerlerine göre kopyalanırlar. Amaç fonksiyon değerleri ise, maksimize (veya minimize) edilecek fonksiyonun o noktadaki değeridir. Bir diziyi amaç fonksiyonuna göre kopyalamak demek maksimizasyon yapılırken, yüksek uygunluk değerli, amaç fonksiyonunda yüksek bir değer alan diziyi sonraki nesillere daha yüksek bir olasılıkla kopyalamak demektir. Doğada bu operatör gerçek anlamda bulunmamakla beraber, canlıların ortama uyum sağlayabilmeleri üreme ve neslini devam ettirme başarımları üreme operatörüne denk olaylardır. Ortama uyum sağlayan (amaç fonksiyon değeri yüksek olan nokta) neslini devam ettirir. (Bir sonraki dizi topluluğuna daha yüksek bir olasılıkla kopyalanır.)

Bu operatör genetik algoritmalarda pekçok değişik biçimde kullanılabilir. En basiti, bir rulet çarkı almak ve bunun üzerinde, noktalara uygunluk değerleri kadar bir dilim ayırmak olabilir. Yukarıdaki dizinin elemanlarının bu kapalı kutudan çıkan uygunluk değerleri Tablo 2.1 ile gösterilmiştir.

Tablo 2.1. Diziler , amaç fonksiyon deęerleri ve kopyalama yzdeleri.

No.	Dizi	Fonk.deęeri(uygunluęu)	Yzdesi
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Toplam		1170	100.0

Tablo 2.1 den hareketle rulet arkı oluřturulur. Bu arkta her bir diziye amaç fonksiyon yzdesiyle orantılı bir dilim ayrılır. rnek arkın grnř řekil 2.13'de verilmiřtir.



řekil 2.13 reme operatr iin rulet arkı.

reme iřlemini gerekleřtirmek iin rulet arkını drtkere dndrmek ve her seferinde okun karřısına gelen diziye yeni nesilin bir ye adayı olarak yazmak gerekir. Bu arada nesillerdeki ye veya eleman sayısının sabit kaldığını kabul ediyoruz. Bu řekilde yeni nesilde, rneęin birinci dizi 0.144, ikinci dizi 0.492 olasılıkla yer alacaklardır.

Bu řekilde seilen drt aday, iřlem havuzunu oluřturur. İřlem havuzunda bu dizilere aprazlama ve/veya mutasyon iřlemleri uygulanır.

Üreme işleminden sonra çaprazlama işlemi uygulanır. Bu işlem iki kademede gerçekleştirilir. İlk olarak işlem havuzundaki diziler rastgele olarak ikiye ikiye eşleştirilir. Sonra, çiftlerde rastgele seçilmiş bir yerden sonraki bitler değiş tokuş edilir. Örneğin, A1 ve A2 dizileri ele alınsın.

$$A_1 = 01110 - 1$$

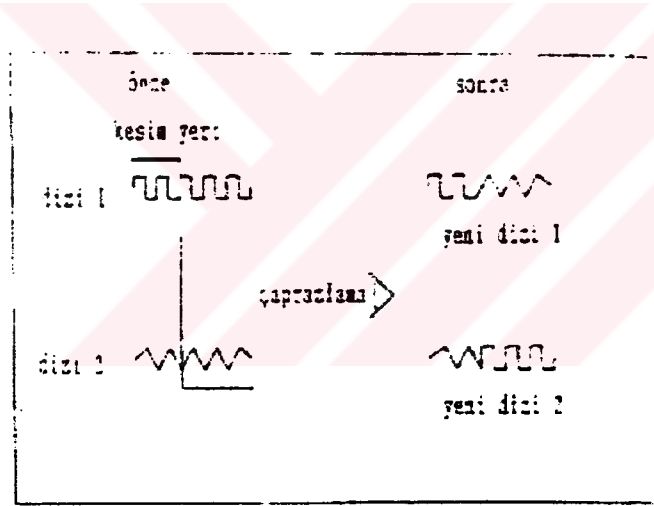
$$A_2 = 1100 - 0$$

Bu dizilerde “ - ” işareti kesim yerini belirtsin. Kesim işlemi, bu işaretten sonraki bir bit için uygulanacaktır. Bu işlemden sonra elde edilecek yeni diziler,

$$A_1' = 01100$$

$$A_2' = 11001$$

olacaktır. Çaprazlama işleminin şematik gösterilimi Şekil 2.14’de verilmiştir.



Şekil 2.14 Çaprazlama işleminin şematik gösterilimi.

Üreme , çaprazlama işlemleri, dizi kopyalaması, rastlantısal sayı üretimi ve bazı alt dizilerin değiş tokuşu gibi basit ancak etkili ve genetik algoritmalara gücünü veren işlemlerden oluşur. İyi olan dizinin daha yüksek olasılıkla kopyalanması, ondan yeni parçalarının başka dizilerin alt parçalarıyla değişimi ise daha iyiyi aramanın temel mantığıdır.

Çaprazlama operatörü başka şekillerde de uygulanabilir. Örneğin, tek bir kesim yeri belirleyeceğine iki yer belirlenir ve sadece bu iki yerin arasında kalan bitler değiş tokuş edilir. Bu uygulamanın iyiliği çaprazlama operatörünün bütün bitler üzerinde homojen olarak uygulanabilmesidir. Eğer diziler, optimize edilmek istenen , parametrelerden oluşturulmuşsa,

örneğin bir PID (Proportional Integral Derivative – Oransal İntegral Türev Alıcı) kontrolünün K_p , K_i ve K_d parametrelerini oluşturan sayılar yan yana yazılarak bir dizi yaratılmışsa, o zaman parametreler arası çaprazlama uygulanabilir. Bit bit çaprazlama yerine, parametreler sekiz bit ile ifade ediliyorsa, byte byte çaprazlama uygulanabilir. Bu tarz çaprazlama işlemi robotkol yörünge optimizasyonunda kullanılmaktadır. Ancak yinede bit bit kesişimin hassasiyeti , seçiciliği daha yüksektir.

Mutasyon operatörü ise, dizilerin rastgele seçilmiş bitlerindeki “1”leri “0” , “0” ları “1” yapar. Genetik algoritmaların rastlantısallık faktörüne en fazla bağlı işlemidir. Eğer mutasyon olasılığı yüksek tutulursa, arama rastlantısal bir şekil alır ki istenen bu değildir. Mutasyon uygulanmaz ise, o zaman sonuç bir yerel en iyiye takılabilir. Örneğin, en iyi netice için bir dizinin en son bitinin “1” olması gerekir ve o andaki elde olan dizilerin hiç birinde son bit “1” olmaz ise en iyi sonuca ulaşmak mümkün olmaz. İşte bu son biti “0” dan “1” yapacak olan operatör mutasyondur. Bu nedenle, her ne kadar aramaya rastlantısallık kazandırsa da mutasyon gereklidir. Ancak çaprazlama operatörünün olasılığına göre çok daha küçük bir olasılık verilir. Bu operatörün aynısı doğada da mevcuttur.

Basit bir genetik algoritmanın adım adım bir örnek üzerinde uygulamasını görmek için amaç fonksiyonu olarak örneğin Şekil 2.11’de gösterilmiş olan $f(x)=x^2$ fonksiyonu ele alınsın. Burada x değişkeni 0 ve 31 arasında değer alsın ve ikilik düzende 5 bit ile ifade edilsin. Sayılar işaretli ve tamsayı olsun. Bu durumda $(00000)_2=(0)_{10}$ ve $(11111)_2=(31)_{10}$ olacaktır. Başlangıç olarak rastgele oluşturulmuş bir dizi topluluğu ele alınsın. Bunun için daha önceki kapalı kutu örneğinde ele alınmış topluluk düşünölsün. Bu topluluğun elemanları Tablo 2.2’de gösterilmiştir. Tabloda soldan sağa olarak x in onluk tabandaki değeri, amaç fonksiyon değeri, seçilme olasılığı verilmiştir.

Tablo 2.2. Genetik algoritmalar.

Dizi No.	İlk Nesil	x	$f(x)=x^2$	$\frac{f_i}{\sum f}$	$\frac{f_i}{<f>}$
1	01101	13	169	0.14	0.58
2	11000	24	576	0.49	1.97
3	01000	8	64	0.06	0.22
4	10011	19	361	0.31	1.23

İlk nesildeki dizilerin ortalama uygunluk değeri 293 ve en iyi uygunluk değeri ise 576 dır.

Bir sonraki adım ve dizilerden işlem havuzunun oluşturulması ve çaprazlama işleminin uygulanmasıdır. Bunlarda Tablo 2.3’de gösterilmiştir.

Tablo 2.3. İşlem havuzu ve çaprazlama işleminin uygulanması.

İşlem Ha.	Eşi	Yer	Yeni Nesil	x	f(x)
0110-1	2	4	01100	12	144
1100-0	1	4	11001	25	625
11-000	4	2	11011	27	729
10-011	3	2	10000	16	256
Toplam					1754
Ortalama					439
En iyi					729

Tablo 2.3'de üreme ve çaprazlama işlemleri uygulanmıştır. Mutasyon olasılığı 0.001 gibi küçük bir değer olarak seçildiğinden,yukarıdaki dizilerin 5x4=20 bitinde görülmemiştir. Sonuçta hiçbir bit, mutasyon sonucu değişmemiştir.

Tablo 2.2 ve Tablo 2.3'deki diziler (ve dolayısıyla temsil ettikleri onluk düzendeki sayılar) karşılaştırıldığında en iyi sonuca yaklaşıldığı görülmektedir. Ortalama ve en yüksek uygunluk değeri yükselmiştir. Örneğin ortalama 293 ten 439 a yükselmiştir. Ancak genetik algoritmalar olasılığa da dayandığı için bir nesilden diğerine geçerken iyileşme bu kadar hızlı olmayabilir. Hatta yeni nesilin ortalaması bir öncekinden daha düşük olabilir. Fakat uzun vadede bulunacak sayılar mutlaka ilk nesili oluşturanlardan daha iyi amaç fonksiyon değerleri vereceklerdir.

Genetik algoritmalarındaki en son adım, bulunacak yeni nesildeki elemanları, ilk nesil gibi ele alarak işlemleri bunların üzerinde yeniden uygulamaktır. Bu işlemler tekrar ve tekrar belli bir

durma kriteri sađlanana kadar yrtlr. En sonra bulunan sayı tekrar onluk dzene evrilerek kullanılır.

2.3.3. Genetik Algoritmanın Teorisi

Optimizasyon problemlerinin zm iin, son zamanlarda genetik algoritmalar incelenmiř, ve adaptif bir řekilde byk ve kompleks bir uzayın arařtırılmasında, etkin oldukları gsterilmiřtir. Bu arařtırma esnasında denk olan evrim mekanizmalarından yeniden reme, aprazlama ve mutasyon rehber olarak alınmıřtır.

Bir optimizasyon probleminin zm iin, bir genetik algoritmada beř iřlem gereklidir.

1. Kodlama (Encoding). Bu metod optimizasyon probleminin karar deđiřkenlerinin kodlanması yntemidir ve bu kodlama 1 ve 0 lardan oluřan ikili bit dizisiyle yapılır. Eđer bir optimizasyon probleminde m karar deđiřkeni varsa ve her karar deđiřkeni n basamaklı ikili bir sayı olarak kodlanırsa, bu durumda kromozom $n \times m$ ikili basamakta oluřan bir dizidir.
2. Ama fonksiyonunun incelenmesi. Bu fonksiyon verilen karar deđiřkenlerinin deđerlendirilmesi ve bir sonu deđerı vermesi iin kullanılır. Kromozom karar fonksiyonunun deđerı, o kromozomun uygunluđudur. Uygunluk deđerı, bu kromozomun yeni kromozomlar oluřturmak iin ebeveyn kromozom olarak seilip seilmeyeceđi olasılıđının belirlenmesi iin kullanılır.
3. Kromozomların oluřturulması. Kromozom poplasyonunun bařlatılması iin bir metoda ihtiya vardır. Genellikle kromozom poplasyonu rastgele (random) bařlatılır.
4. Ardıřık Kromozom oluřturulmaları sırasında, genetik algoritmalar, ebeveyn seim teknikleri kullanırlar ve bunlar dođal seleksiyon iřlemini taklit ederler. Bu iřleme gre, en uygun elemanlar en sık rerler. Ebeveyn seiminden sonra, aprazlama iřlemi uygulanacak iki kromozom yeniden birleřtirilerek iki yeni kromozom oluřur.
5. Genetik algoritmanın ynlendirilmesi iin bir grup parametre nceden tanımlanmıřtır. Bunlara rnek olarak, ikili dizi olarak kodlanan her karar deđiřkeninin uzunluđu, retilmesi ve kullanılması gereken kromozomların sayısı, aprazlama hızı, mutasyon hızı, ve iřlemin durma kriteri verilebilir. Iřlemin durma kriteri, iřlem sayısı veya ama fonksiyonu iin bir tolerans deđerı olarak nceden belirlenir.

Adı geçen beş işlem sırasıyla, P1–P5 olarak adlandırılırsa, bir genetik algoritma şu şekilde tanımlanabilir.

1. Karar değişkenlerini bir kromozom olarak kodla
2. Şu anki jenerasyon olarak bir kromozom oluşumunu başlat. (oluşumu başlatma metodu kullanarak) (P3)
3. Bir işlem durdurma kriteri gerçekleşene kadar aşağıdaki iterasyonları gerçekleştir.
 - (i) P2’de anlatılan o anki jenerasyon oluşumunun amaç fonksiyon değerlerini değerlendir.
 - (ii) Yeni bir kromozom jenerasyonu oluşturmak için, daha yüksek uygunluk değerlerine sahip bazı kromozomları ebeveyn kromozom olarak seç. (P4’deki ebeveyn seçimi)
 - (iii) Çaprazlama ve mutasyon (P5) operatörlerini, bir önceki aşamada seçilen ebeveyn kromozomlara uygula. Ebeveynlere uygulanan operatörlerin seçimi, ebeveynle ilgili rastgele değerlerin yanısıra, P5’te tanımlanan önceden belirlenmiş çaprazlama ve mutasyon hızlarına göre yapılır.
 - (iv) Şu anki jenerasyon olarak, bütün kromozom oluşumları yeni oluşturulan kromozomlarla değiştir.

Hoffmeister ve Böck (1991), genetik algoritmayı sekiz parametrelilik bir algoritma olarak tanımlanmıştır [42]. Bu çalışmada, daha önce anlatılan genetik algoritmanın beş parçası genişletilmiş ve sekiz parametrelilik bir algoritma olarak tanımlanmıştır.

$$G.A.=(p^0, I, \lambda, L, f, s, c, m, T) \quad (2.20)$$

Burada

$$p^0 = (a_1^0, \dots, a_\lambda^0) = \left[\begin{bmatrix} a_{1,1} \\ \vdots \\ a_{1,L} \end{bmatrix} \dots \begin{bmatrix} a_{\lambda,1} \\ \vdots \\ a_{\lambda,L} \end{bmatrix} \right] \in I^\lambda \quad \text{Kromozom oluşumları}$$

$$I = \{0,1\}^L \quad \text{Kromozomların kodlanması}$$

$\lambda \in \mathbb{N}$

Kromozomların sayısı

 $L \in \mathbb{N}$

Kromozomların uzunluğu

 $f: I \rightarrow \mathbb{R}$

Fonksiyon değerlerinin uygunluğu

 $s: I^\lambda \rightarrow I$

Kromozomların seçilme işlemi

 $c: I^2 \rightarrow I^2$

Çaprazlama işlemi

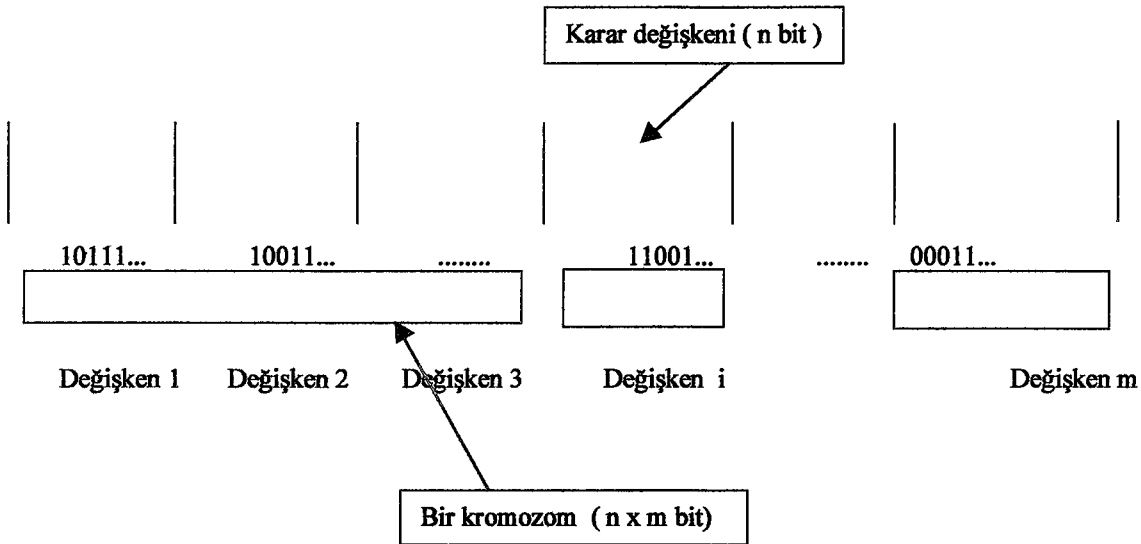
 $m: I \rightarrow I$

Mutasyon işlemi

 $T: I^\lambda \rightarrow \{0,1\}$

Sonlandırma kriteri

Kromozom sayısı λ tanedir. Kromozomların başlangıç popülasyonu (kromozom oluşumu), p^0 rastgele oluşturulur. α_k^t ifadesinin anlamı, t nci jenerasyon popülasyonunun $(p^t)^k$ k nci kromozomu demektir. Bir kromozom, I, ikili basamak dizisi olarak (1'ler ve 0'lar) kodlanır. Eğer bir optimizasyon probleminde m karar değişkeni varsa ve her karar değişkeni n basamaklı ikili bir sayı olarak kodlanmışsa, bu durumda bir kromozom $m \times n$ lik bir dizidir. Şekil 2.15 de bir sütun vektörü $[a_{k,1}, \dots, a_{k,L}]^T$ olarak gösterilir. $g: X \rightarrow Y$ terimi, $x \in X$ ve $y \in Y$ durumda x'den y'ye tanımlı bir fonksiyondur. N ve R değişkenleri, tamsayı ve reel sayı kümeleridir. Genetik algoritmanın evrim işlemi ($T=0$), işlem durma kriterlerinden biri karşılanana kadar ($T=1$) devam eder.



Şekil 2.15 Karar değişkenlerinin bir kromozom olarak kodlanması.

Ebeveyn seçimi işlemi, s, t nci jenerasyondaki $p' = (a'_1, \dots, a'_\lambda)$ popülasyondan, bir ara popülasyon olan $p'' = (a''_1, \dots, a''_\lambda)$ üretimidir. Herhangi bir $a''_i = a'_q = s(P')$, verilen bir rastlantısal reel sayı a_i tarafından, aşağıdaki şartı sağlayarak seçilir.

$$0 \leq \alpha_i \leq \sum_{j=1}^{\lambda} f(a'_j) \quad (2.14)$$

q indisi şöyle elde edilir.

$$q = \min \left\{ k \mid \forall k \in \{1, \dots, \lambda\}, \quad s, t, \alpha_i \leq \sum_{k=1}^{\lambda} f(a'_k) \right\} \quad (2.15)$$

P^t popülasyonundaki herhangi seçilmiş kromozom çifti için, ortak bir reel değer, $0 \leq r \leq 1$ rastlantısal olarak oluşturulur. Eğer, r daha önce tanımlanan çaprazlama eşiği olan r_c den daha büyükse çaprazlama işlemi bu çift kromozoma uygulanır. Bu çalışmada 3 değişik çaprazlama stratejisi uygulanmıştır. Birincisi 2 noktalı çaprazlamadır. c_{tp} , ve bu da, p^t popülasyonundan bir ara popülasyon olan p^t yi oluşturur ve şöyle tanımlanır.

$$\begin{aligned} \begin{Bmatrix} a''_i \\ a''_{i+1} \end{Bmatrix} &= c_{tp} \left(\begin{Bmatrix} a'_i \\ a'_{i+1} \end{Bmatrix} \right) \quad \forall i \in \{1, 3, \dots, 2k+1, \dots, \lambda-1\} \\ &= \left\{ \begin{bmatrix} a_{(i+1),1}, a_{(i+1),2}, \dots, a_{(i+1),p_1}, a_{i,(p_1+1)}, \dots, a_{i,p_2}, a_{(i+1),(p_2+1)}, \dots, a_{(i+L),2} \end{bmatrix}^T \right. \\ &\quad \left. \begin{bmatrix} a_{i,1}, a_{i,2}, \dots, a_{i,p_1}, a_{(i+1),(p_1+1)}, \dots, a_{(i+1),p_2}, a_{i,(p_2+1)}, \dots, a_{i,L} \end{bmatrix}^T \right\} \end{aligned} \quad (2.16)$$

Burada, $1 \leq p_1 < p_2 \leq L$ Bu çaprazlama stratejisinde, bir kromozom çiftinde iki pozisyon seçilir. Bu kromozom çifti 3 alt kromozomlara bu iki noktayla bölünür, ve 1 nci ve 3 ncü alt kromozomlar değiş tokuş edilir.

İkinci çaprazlama stratejisi çok nokta çaprazlamadır. c_{mp} , p^t popülasyonundan bir ara popülasyon p^t oluşturur. Ve şöyle tanımlanır.

$$\begin{Bmatrix} a''_i \\ a''_{i+1} \end{Bmatrix} = c_{mp} \left(\begin{Bmatrix} a'_i \\ a'_{i+1} \end{Bmatrix} \right) \quad \forall i \in \{1, 3, \dots, 2k+1, \dots, \lambda-1\}$$

$$= \bigcup_{k=1}^L \left\{ \left(\begin{bmatrix} a_{i,k} \\ a_{(i+1),k} \end{bmatrix}, \text{ ise } p_k \geq p_{mp} \right) \vee \left(\begin{bmatrix} a_{(i+1),k} \\ a_{i,k} \end{bmatrix}, \text{ ise } r_k < r_{mp} \right) \right\} \quad (2.17)$$

Burada, $0 \leq p_k, p_{mp} \leq 1$. Bu çaprazlama stratejisinde, bir çift kromozom içinde birden fazla çaprazlama biti seçilir. Çaprazlama işlemi bit seviyesinde yani çaprazlama prosesi bit bit yapılır. Her çift kromozomdaki çaprazlama noktalarının sayıları ve çaprazlama pozisyonları rastlantısal olarak seçilir, hepsi birbirinden ayrıdır.

Üçüncü çaprazlama stratejisi düzgün çaprazlamadır. c_{un} ve p^i popülasyonundan ara popülasyon olan p^t yi üretir. Öncelikle, maskeleme için L uzunluğunda ikili dizi oluşturulur. L nin gerçek değerleri, $0 \leq r_j < 1$, ($j=1,2,...,L$) rastlantısal olarak oluşturulur. Eğer j nci rastlantısal sayı olan r_j , daha önceden tanımlanan eşik değeri olan r_{ma} değerinden büyük veya eşitse, ikili dizideki j nci elemanın değeri 1 olarak tanımlanır. Diğer durumda 0 olarak kabul edilir.

Maskleme şöyle tanımlanır.

$$ma = \bigcup_{j=1}^L \left\{ \left([1] \text{ eger } r_j \geq p_{ma} \right) \vee \left([0], \text{ eger } r_j \leq p_{ma} \right) \right\} \quad (2.18)$$

Düzgün çaprazlama şöyle tanımlanır;

$$\left\{ \begin{matrix} a''_i \\ a''_{i+1} \end{matrix} \right\} = c_{un} \left(\left\{ \begin{matrix} a'_i \\ a'_{i+1} \end{matrix} \right\} \right) \quad \forall i \in \{1,3,...,2k+1,...,\lambda-1\}$$

$$= \bigcup_{k=1}^L \left\{ \left(\begin{bmatrix} a_{i,k} \\ a_{(i+1),k} \end{bmatrix}, \text{ eger } ma_k = 0 \right) \vee \left(\begin{bmatrix} a_{(i+1),k} \\ a_{i,k} \end{bmatrix}, \text{ eger } ma_k = 1 \right) \right\} \quad (2.19)$$

Çok noktalı çaprazlama stratejisine benzer olarak, çaprazlama prosesi bir çift kromozomda bit bit yapılır. Düzgün çaprazlama stratejisinde, çaprazlama pozisyonları bir maske L uzunluğunda bit dizisi içinde tanımlanır. Bir popülasyondaki bütün kromozomlar için aynı durumlarda çaprazlama yapılır. Diğer taraftan çok noktalı çaprazlama stratejisinde her kromozom çiftine değişik noktalarda çaprazlama yapılır, çünkü önceden tanımlanan herhangi bir maske kullanılmaktadır. Şekil 2.16.'da adı geçen bu 3 çaprazlama işlemleri gösterilmektedir.

Kromozom dizisindeki p^t , herhangi bir kromozomu için $0 \leq r \leq 1$ aralığında rastlantısal olarak reel değeri oluşturulur. r önceden belirlenen mutasyon eşik değerinden r_m , daha küçükse, bu kromozom mutasyon işlemi uygulanır. Basit olarak mutasyon işlemi bir kromozomdaki bir biti 0'da 1'e veya 1'den 0'a değiştirir. Mutasyon işlemi m , p^t kromozom dizisinden, p^t kromozom dizisi aşağıda tanımlanan biçimde oluşturur.

$$a_i^{t'} = m(a_i^t) \quad \forall i \in \{1, \dots, \lambda\}$$

$$a_{i,k}' = \begin{cases} a_{i,k}, & k \in \{1, 2, \dots, p-1, p+1, \dots, L\} \text{ için} \\ \bar{a}_{i,k}, & k = p \text{ için} \end{cases} \quad (2.20)$$

burada $1 \leq p < L$ şeklindedir.

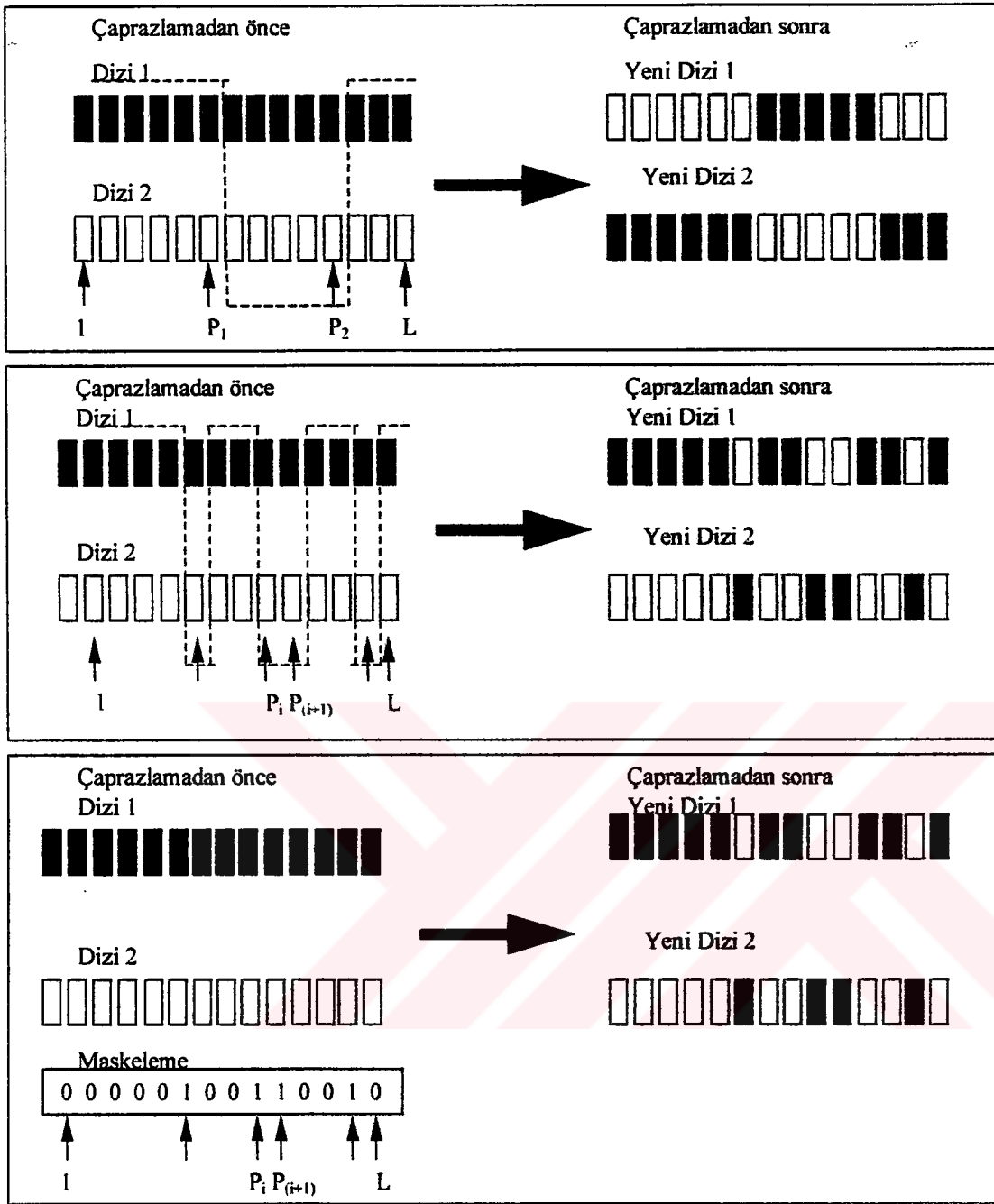
Kısaca bir genetik algoritma aşağıdaki adımlarla ifade edilebilir.

Adım 1 , Karar değişkenleri bir kromozom olarak kodlanır.

Adım 2 , Kromozom dizisini rastlantısal olarak oluşturur.

Adım 3 , Durdurma koşullardan biri sağlanana kadar aşağıda sıralanan işlemlere devam edilir.

- (i) Her bir kromozom için amaç fonksiyonunun değerini hesaplanır.
- (ii) En düşük hata değerlerini veren kromozom grubunu seçilir.
- (iii) Çaprazlama ve mutasyon işlemini uygulanır.
- (iv) Yeni oluşturulan kromozomları, kromozom grubuna eklenir (yeni nesli oluşturur).



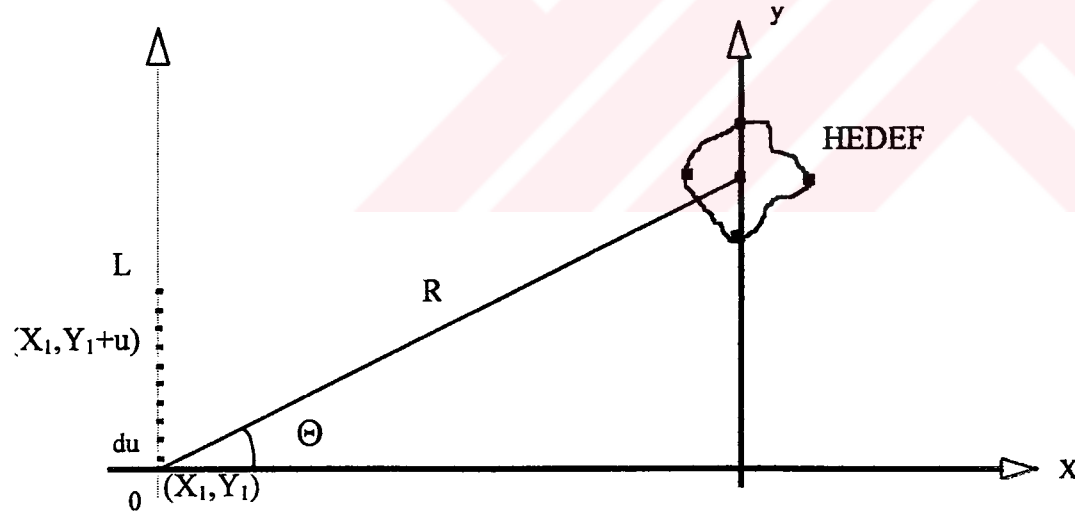
Şekil 2.16 İki Nokta, çok nokta ve düzgün çaprazlama.

3 GENETİK ALGORİTMANIN YAPAY AÇIKLIKLI RADAR TERS DÖNÜŞÜM BAĞINTISINA UYGULANMASI

3.1 UYGULANAN YÖNTEM

Bu bölümde, Bölüm 2.3 de anlatılan global optimizasyon yöntemi Genetik Algoritma Bölüm 2.1 de verilen Yapay Açıklıklı Radar ters dönüşüm bağıntısına uygulanmıştır.

Gözönüne alınan YAR görüntüleme geometrisi Şekil 3.1 de görülmektedir. Hedef nokta saçıcılardan oluşmakta ve radar ile aynı düzlemde bulunmaktadır. Radar (x,y) düzleminde $x=X_1$ hattı boyunca hareket etmektedir. Radar (x,y) düzleminde $u \in [0,L]$ sentezlenmiş açıklıkta (X_1, Y_1+u) koordinatında iletim yapar ve yine aynı koordinatta işareti alır. (X_1, Y_1) bilenen sabitlerdir. Y_1 bakış modu dikey koordinatıdır ve $Y_1=0$ alınmıştır. (X_1, Y_1) kartezyen koordinatları, (θ_p, R_p) polar koordinatlarıyla ifade edilebilir.



Şekil 3.1 Gözönüne alınan YAR görüntüleme geometrisi

İlk adımda, hedef geometrisini tanımlayan cismin yansıma fonksiyonunun ve hedefe gidip gelen işaretin faz gecikmesinin bilindiği kabul edilmektedir. Bilinen bu büyüklükler (2.7) de tanımlanan YAR ters dönüşüm bağıntısında yerine konulduğunda hedefe ilişkin ölçülen toplam yansıyan işaret bilgisi $s_i^m(u, w)$ elde edilir.

Şekil 3.1 de görüldüğü gibi radar (X_1, Y_1) başlangıç konumundan hareket ederek $u \in [0, L]$ yapay açıklığı boyunca, örnek sayısı (nn) kadar işareti gönderip almaktadır. Radar anteninin gerçek boyutu du ile ifade edilirse, yapay açıklık uzunluğu $L=du.nn$ olarak bulunur. (2.7) bağıntısının sağ tarafında bulunan işaretin gidiş geliş faz gecikmesi ifadesindeki kartezyen koordinatlar ve hedefin radara olan uzaklığı cinsinden $X_1 - x = R \cos \theta$, $Y_1 - y = R \sin \theta$ bağıntılarıyla ifade edilir.

Polar koordinatlar (2.7) de yerine konulduğunda ölçülen toplam yansıyan işaret bilgisi $S_i^m(u, w)$ aşağıdaki bağıntı ile gösterilir.

$$S_i^m(u, w) = \sum_{i=1}^{m(L)} f(x, y) \exp \left[j2k \sqrt{(R_i^m \cos \theta_i^m)^2 + (R_i^m \sin \theta_i^m + u)^2} \right] \quad (3.1)$$

İlk adımda, hedef geometrisini tanımlayan cisim yansıma fonksiyonu $f(x, y)$ ve $u \in [0, L]$ açıklığı boyunca hedefe gidip gelen, işaretlerin faz gecikmelerinin bilindiği kabul edilerek ölçülen toplam yansıyan işaret bilgisi $S_i^m(u, w)$ elde edilmiştir.

İkinci adımda, birinci adımda elde edilen, bilinen ölçülen toplam yansıyan işaret bilgisi $S_i^m(u, w)$ ve hedefe gidip gelen işaretlerin faz gecikmeleri kullanılarak, genetik algoritma ile hedef geometrisini tanımlayan cisim yansıma fonksiyonu $f(x, y)$ hesaplanmaktadır.

Hesaplanan bu $f(x, y)$ değeri kullanılarak, $S_i^c(u, w)$ hesaplanan toplam yansıyan işaret bilgisi elde edilir.

$$S_i^c(u, w) = \sum_{i=1}^{m(L)} f(x, y) \exp \left[j2k \sqrt{(R_i^c \cos \theta_i^c)^2 + (R_i^c \sin \theta_i^c + u)^2} \right] \quad (3.2)$$

Daha sonra, ölçülen ve hesaplanan cisim yansıma fonksiyonunun ifade ettiği, cismin konumunu gösteren menzil ve açı değerleri (yani ölçülen ve hesaplanan toplam yansıyan işaret bilgileri) birbiriyle kıyaslanarak hata fonksiyonu (e),

$$e = \sum_{i=1}^{nn} \left\{ \text{Re}[S_i^m(u, w)] - \text{Re}[S_i^c(u, w)] \right\}^2 + \sum_{i=1}^{nn} \left\{ \text{Im}[S_i^m(u, w)] - \text{Im}[S_i^c(u, w)] \right\}^2 \quad (3.3)$$

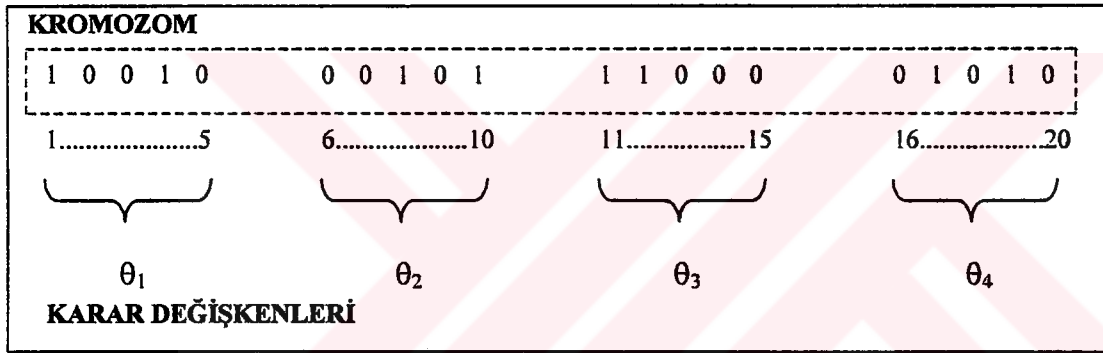
bağıntısıyla gösterilir.

YAR ve genetik algoritmaya ilişkin parametreler , hata fonksiyonunu minimize edecek şekilde optimize edilerek, hedefin konumu belirlenmiştir.

3.2. GENETİK ALGORİTMANIN UYGULAMA ADIMLARI VE AKIŞ DİAGRAMI

Karar değişkenleri olarak kabul edilen R ve θ değerleri ikili sayı sisteminde (0 ve 1 dizisi şeklinde), b (bit sayısı) basamaklı olarak kodlanmıştır. Hedef nokta sayısı kadar kodlanmış karar değişkenleri birleştirilerek kromozom yapısı oluşturulur.

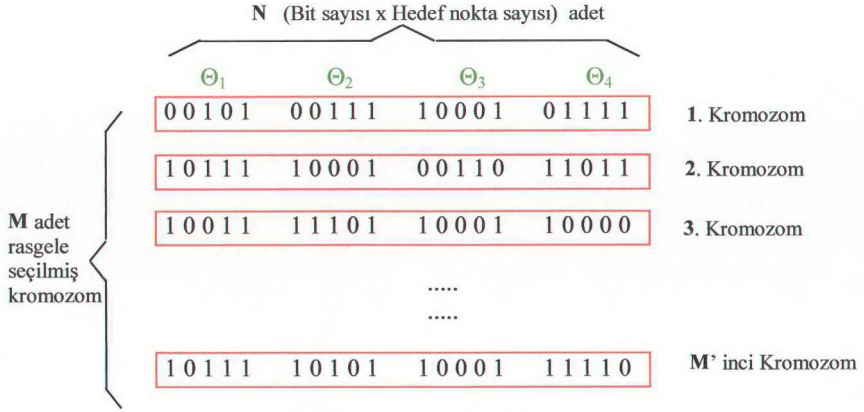
Bir kromozom içindeki toplam bit sayısı, karar değişkeni sayısı çarpı bir karar değişkenini kodlayan bit sayısıdır. Bit sayısı 5 ve 4 hedef nokta için oluşan kromozom örneği Şekil 3.2.'de görülmektedir.



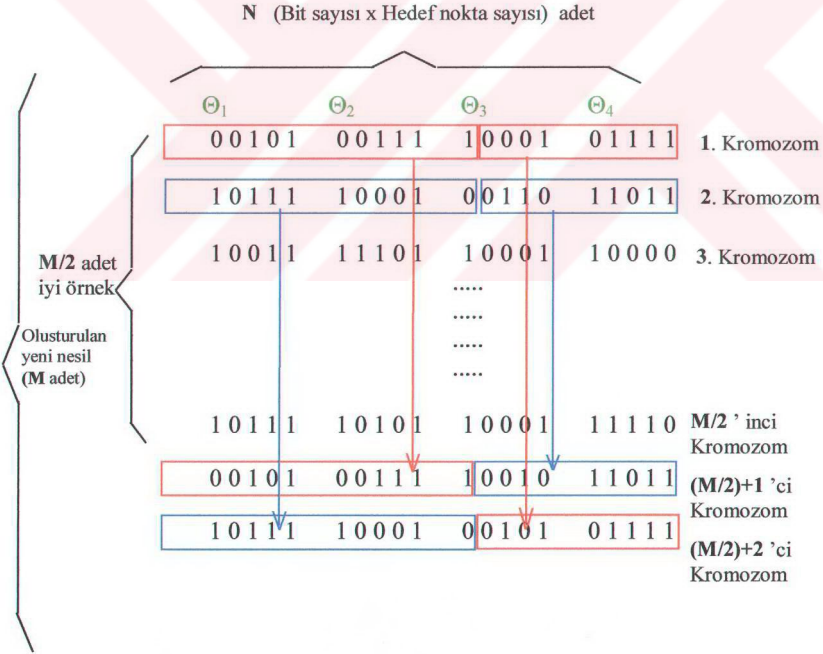
Şekil 3.2 Kromozom örneği.

Başlangıç popülasyonu için M tane rastgele kromozom üretilerek karar değişkeni havuzu oluşturulur. Şekil 3.2.'deki örnek kromozom için oluşturulan havuz Şekil 3.3.'de gösterilmiştir. Her bir kromozom için (3.3.) bağıntısı ile verilen hata fonksiyonu hesaplanır. Bu kromozomlar hata fonksiyonlarına göre küçükten büyüğe doğru sıralanır.

Amaç, minimum hatayı veren R ve θ değerlerini elde etmek olduğundan en kötü değeri veren $M/2$ adet kromozom havuzundan çıkartılır. Görece hatası daha az olan geriye kalan $M/2$ adet kromozm kendi aralarında çaprazlama işlemi uygulanarak, toplam kromozom sayısı M olacak şekilde yeni kromozomlar üretilir.



Şekil 3.3 Karar değişkenleri havuzu.



Şekil 3.4 Bit sayısı=5 ve çaprazlama bit konumu=13 için örnek çaprazlama işlemi

Çaprazlama işlemi şu şekilde yapılır, sıralanan kromozomlar ardışık ikili gruplara ayrılarak bir sonraki nesil için ebeveynleri oluşturur. Tüm ebeveynler aynı bit konumunda çaprazlandığı iki nokta çaprazlama tekniği kullanılmıştır. Çaprazlamanın yapıldığı bit konumu her nesilde yeniden rastgele olarak seçilmektedir. Bu işlem Şekil 3.4'de gösterilmiştir.

Bir sonraki aşama olan mutasyon işlemi için havuzdaki rastgele bir kromozom içindeki rastgele bir bit seçilerek ve bu bit terslenir. (0 ise 1,1 ise 0 yapılır) Bu genetik arama algoritmasının yerel bölgelere takılı kalmasının önüne geçer.

Bu yeni oluşturulan nesil üzerinde anlatılan işlemler iterasyon sayısı kadar tekrarlanır. Tüm iterasyonlar tamamlandığında en az hatayı sağlayan kromozom, hedefe ilişkin karar değişkenleridir.

Bu karar değişkenleri ölçülen değerlerle kıyaslanarak, hatanın değişimi , farklı örnek sayıları ve GA parametreleri gözönüne alınarak incelenmiştir. Bu inceleme YAR , Hooke Jeeves uygulama örnekleri ile birlikte Bölüm 4'te verilmiştir.

Yapay açıklıklı radara, Genetik Algoritmanın uygulanmasını sağlayan programın akış diagramı Şekil 3.5'de verilmiştir.

YAPAY AÇIKLIKLI RADARIN ÇALIŞMA FREKANSINI,
KONUMUNU VE HEDEFİN KONUMUNU BELİRLEYEN
BÜYÜKLÜKLERİ TANIMLA

TANIMLANAN HEDEF İÇİN TOPLAM YANSIYAN İŞARET
BİLGİSİ $\{S^m(u,w)\}$ ÖLÇÜLÜR

GENETİK ALGORİTMA YAR 'a UYGULANARAK KARAR
DEĞİŞKENLERİ TANIMLANIR. BU DEĞİŞKENLERDEN MEYDANA
GELEN M ADET KROMOZOMUN BULUNDUĞU HAVUZ RASTGELE
BİR BİÇİMDE OLUŞTURULUR

OLUŞTURULAN KROMOZOM HAVUZUNDA BİT SAYISINA GÖRE
ÖRNEK KARAR DEĞİŞKENLERİNİ OLUŞTUR.

YAR BAĞINTISI KULLANARAK, TOPLAM YANSIYAN
İŞARET BİLGİSİNİ $\{S^c(u,w)\}$ HESAPLA

ÖLÇÜLEN VE HESAPLANAN TOPLAM YANSIYAN İŞARET
BİLGİLERİ ARASINDAKİ FARKI HESAPLAYARAK
BU FARKLARI TOPLA.

KROMOZOMLARI OLUŞTURDAKİ HATAYA GÖRE KÜÇÜKTEN
BÜYÜĞE DOĞRU SIRALA. SIRALANAN KROMOZOMLARI İKİYE BÖL.
DAHA KÜÇÜK HATAYA SAHİP OLAN BÖLÜMÜ ELDE TUT

$M/2$ ADET ELDE BULUNAN KROMOZOMA ÇAPRAZLAMA İŞLEMİ UYGULAYARAK YENİ
BİREYLERİ OLUŞTUR

YENİ BİREYLER ARASINDAN RASTGELE BİR KROMOZOMUN
RASTGELE BİR BİT 'İNİN DEĞERİNİ DEĞİŞTİR.
(MUTASYON İŞLEMİ)

BİR SONRAKİ İTERASYON İÇİN HAZIRLIK YAP.

TÜM İTERASYONLAR
TAMAMLANDI MI?

E

EN AZ HATAYI
SAĞLAYAN θ 'LARI
GÖSTER

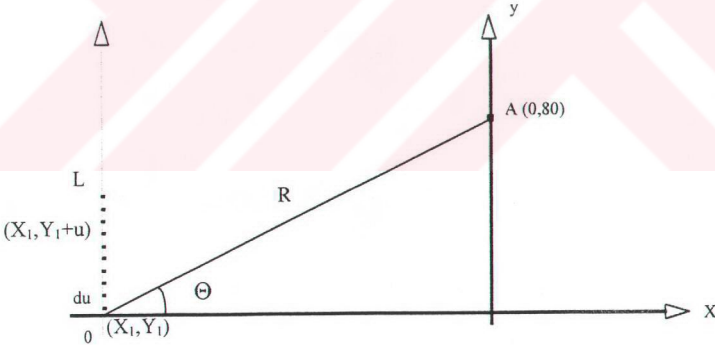
Şekil 3.5 YAR'ın G.A.'ya uygulanmasını sağlayan programın akış diagramı

4. UYGULAMA ÖRNEKLERİ

4.1. Seçilen Uygulama Örnekleri

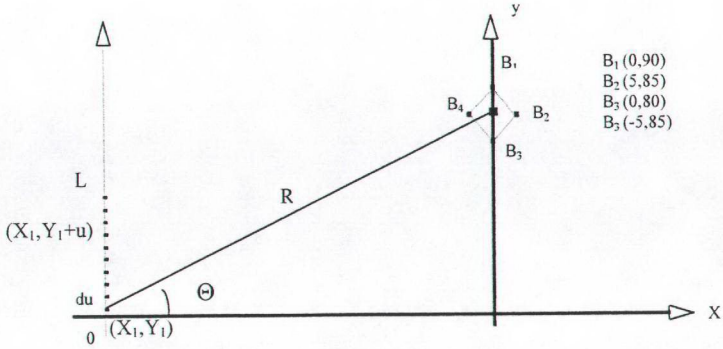
Seçilen uygulama örneklerinde üç farklı hedef dağılım gözönüne alınmıştır. Üç farklı hedef dağılımı sırasıyla tek, dört ve sekiz saçıcı noktadan oluşmaktadır. Radar alıcısı $(X_1, Y_1) = (-200, 0)$ konumunda olup, $f=1$ GHz de çalıştığı kabul edilmiştir. Anten pozitif y eksenini doğrultusunda yapay açıklık boyunca hareket etmektedir. Yapay açıklık uzunlukları Hooke Jeeves ve Genetik Algoritma uygulamalarında $u \in [0, 5]$ alınmıştır. Ancak YAR uygulamasında açıklık boyu ayrıca $u \in [0, 150]$ de alınmıştır. Yapay açıklık boyunca alınan örnek sayısı (alıcı anten konumlarının sayısı) $nn=10, 20, 30$ ve $nn=40$ secilerek, programlarda her anten anı için noktasal hedeflerden yansıyan işaret gözönüne alınmıştır.

Şekil 4.1.'de gösterildiği gibi tek nokta saçıcıdan oluşan hedefin gözönüne alındığı, YAR görüntüleme geometresinde, hedefin konumu A $(0, 80)$ alınmıştır. Bu noktaya karşılık düşen ilk açı değeri $\theta=21,80^\circ$ dir.



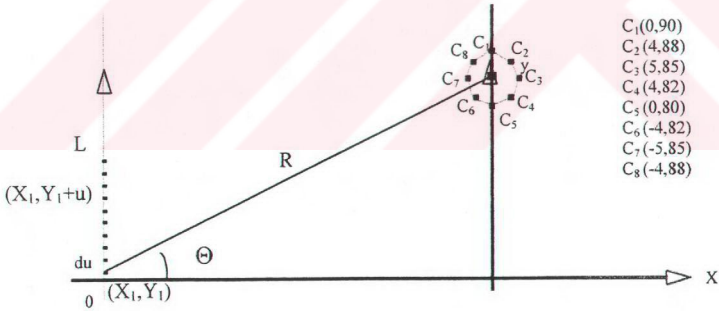
Şekil 4.1 Tek nokta hedef için gözönüne alınan YAR görüntüleme geometrisi.

Şekil 4.2.'de gösterildiği gibi dört nokta saçıcıdan oluşan hedefin gözönüne alındığı, YAR görüntüleme geometrisinde, hedefi oluşturan nokta saçıcıların konumu $B_1(0,90)$, $B_2(5,85)$, $B_3(0,80)$, $B_4(-5,85)$ alınmıştır. Bu noktalara karşılık düşen açı değerleri $\theta_1=24,22^\circ$, $\theta_2=22,52^\circ$, $\theta_3=21,80^\circ$, $\theta_4=23,55^\circ$ dir.



Şekil 4.2 Dört nokta hedef için gözönüne alınan YAR görüntüleme geometrisi.

Şekil 4.3.'de gösterildiği gibi sekiz nokta saçıcıdan oluşan hedefin gözönüne alındığı, YAR görüntüleme geometresinde hedefi oluşturan nokta saçıcıların konumu $C_1(0, 90)$, $C_2(4, 88)$, $C_3(5, 85)$, $C_4(4, 82)$, $C_5(0, 80)$, $C_6(-4, 82)$, $C_7(-5, 85)$, $C_8(-4, 88)$ alınmıştır. Bu noktalara karşılık düşen açı değerleri sırasıyla $\theta_1=24,22^\circ$, $\theta_2=22,70^\circ$, $\theta_3=22,52^\circ$, $\theta_4=21,89^\circ$, $\theta_5=21,80^\circ$, $\theta_6=23,33^\circ$, $\theta_7=23,55^\circ$, $\theta_8=24,17^\circ$ dir.



Şekil 4.3 Sekiz nokta hedef için gözönüne alınan YAR görüntüleme geometrisi.

Uygulamadaki hata miktarı, Genetik ve Hooke Jeeves Algoritmalarının (3.3) bağıntısına uygulanması ile hesaplanan açı ve menzil değerlerinin (θ_i^c , R_i^c), ölçülen

açı ve menzil değerleri (θ_i^m, R_i^m) kullanılarak, $\vec{r}_{c_i}$ vektörü i'nci noktasal hedefin hesaplanan konum vektörü, $\vec{r}_{m_i}$ vektörü i'nci noktasal hedefin ölçülen konum vektörü olmak üzere; aşağıdaki bağıntı ile hesaplanan değerleri alınmıştır.

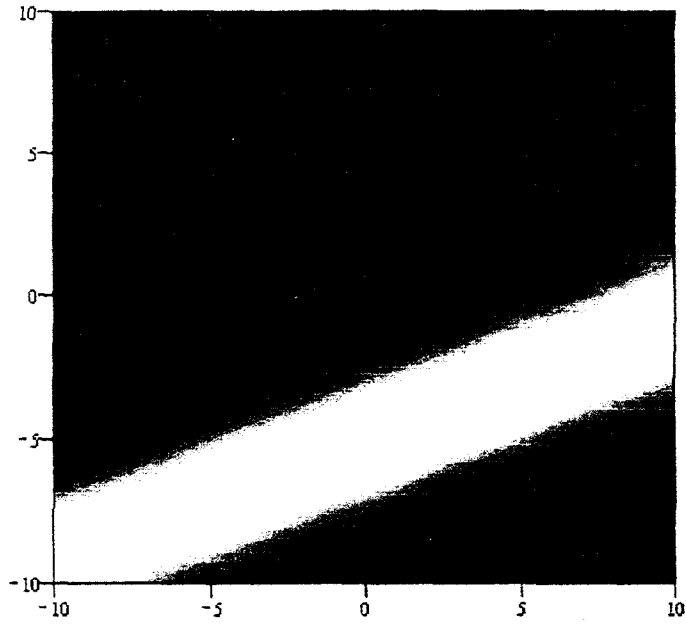
$$\text{hata} = \frac{\sum_{i=1}^n |\vec{r}_{c_i} - \vec{r}_{m_i}|}{n} \quad (4.1)$$

Burada n nokta hedef sayısını gösterir.

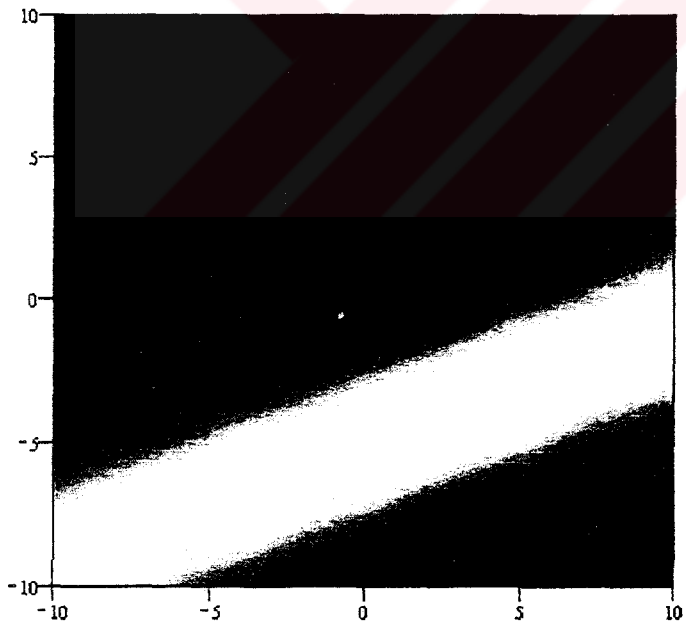
4.2. Yapay Açıklıklı Radar Ters Dönüşüm Algoritması İle YAR da Gerçekleştirilen Uygulama Örnekleri.

YAR görüntüleme algoritması seçilen uygulama örneklerine, üç farklı hedef dağılımı için uygulanmıştır. Her bir hedef dağılımı için yapay açıklık uzunlukları $u \in [0,5]$ ve $u \in [0,150]$, yapay açıklık boyunca alınan örnek sayısı $nn=10$ ve $nn=40$ seçilerek algoritma uygulanmıştır.

Şekil 4.1'de gösterilen tek nokta hedef için gözönüne alınan YAR görüntüleme geometrisine YAR ters dönüşüm algoritması uygulanmıştır. Şekil 4.4'de $u \in [0,5]$ ve $nn=10$ için elde edilen görüntüden görüleceği gibi nokta hedefin konumunu belirleyecek bir görüntü oluşmamıştır. Benzer şekilde Şekil 4.5'de $u \in [0,5]$ ve $nn=40$ için de benzer bir görüntü elde edilmiştir. Ancak konumun net olarak hesaplanabilmesi için , elde edilen hedef dağılımı üzerinde görüntü noktalarının yerinin tam olarak belirlenebilmesi için optimizasyon yöntemine ihtiyaç vardır [43].

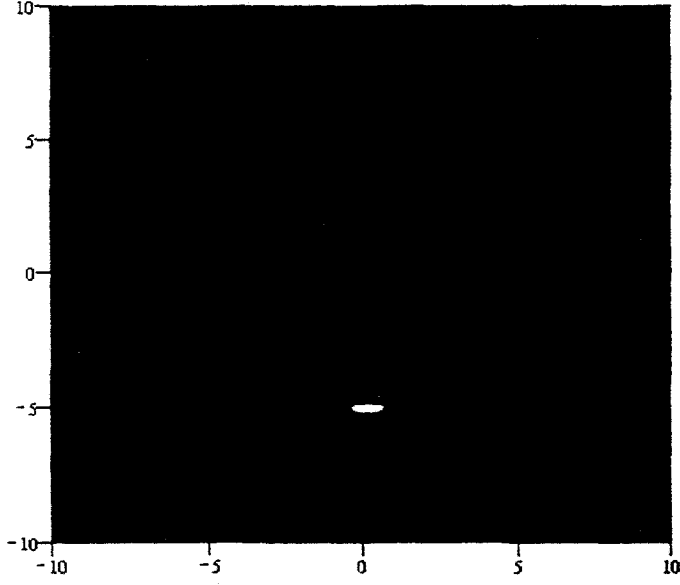


Şekil 4.4 Tek nokta hedef için $u \in [0,5]$ ve $nn = 10$ için elde edilen YAR görüntüsü.

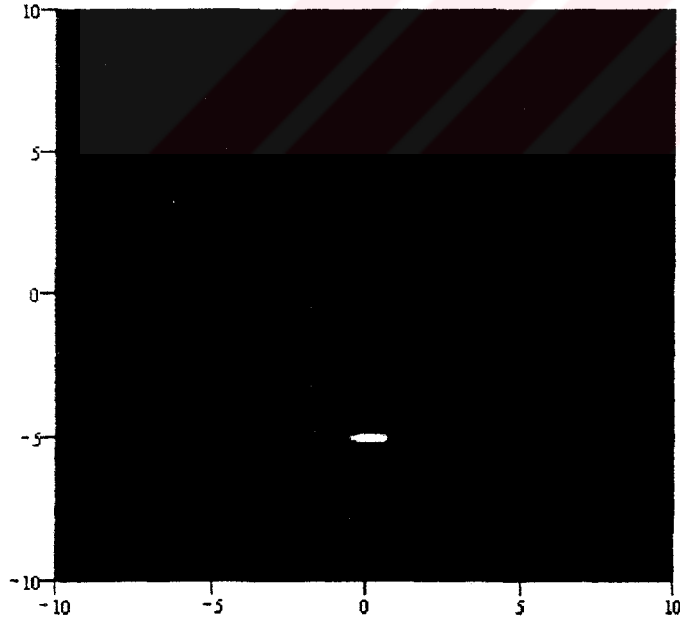


Şekil 4.5 Tek nokta hedef için $u \in [0,5]$ ve $nn=40$ için elde edilen YAR görüntüsü.

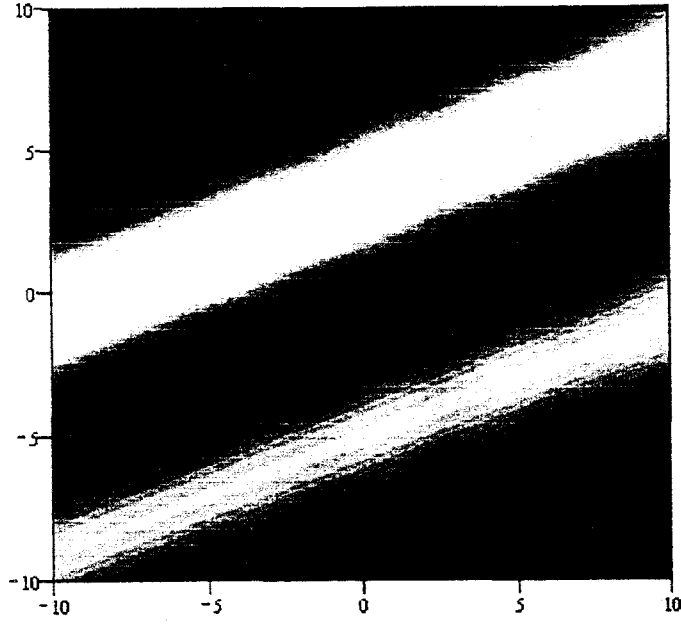
Ancak tek nokta hedef için açıklık boyu $u \in [0,150]$ alındığında sırasıyla $nn = 10$ ve $nn=40$ için Şekil 4.6 ve Şekil 4.7 de gösterilen görüntüler elde edilmiştir. Bu şekillerden görüldüğü gibi konum hesabı yapabileceğimiz bir görüntü elde edilmiştir.



Şekil 4.6 Tek nokta hedef $u \in [0,150]$ ve $nn = 10$ için elde edilen YAR görüntüsü

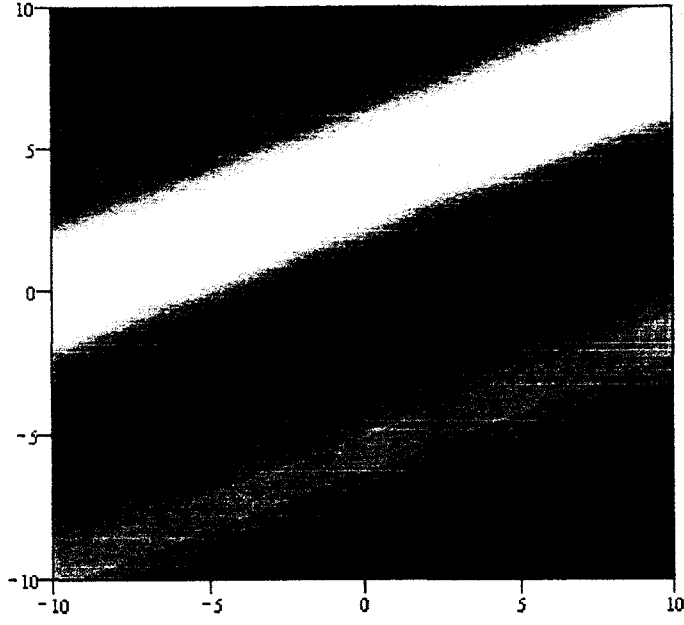


Şekil 4.7 Tek nokta hedef $u \in [0,150]$ ve $nn=40$ için elde edilen YAR görüntüsü.



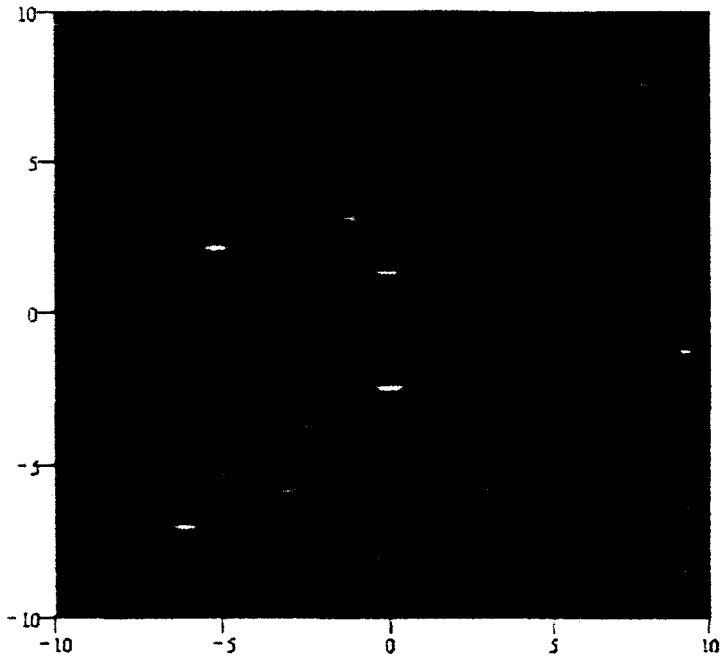
Şekil 4.8 Dört nokta hedef $u \in [0,5]$ ve $n_n = 10$ için elde edilen YAR görüntüsü.

Şekil 4.2'de gösterilen dört nokta hedef için gözönüne alınan YAR görüntüleme geometrisine YAR ters dönüşüm algoritması uygulanmıştır. $u \in [0,5]$ için sırasıyla $n_n = 10$ ve $n_n = 40$ için Şekil 4.8 ve Şekil 4.9'da gösterilen görüntüler elde edilmiştir. Ancak görüntülerden de görülebileceği gibi nokta hedeflerin konumunun hesaplanabileceği bir görüntü elde edilememiştir. Fakat konumun net olarak hesaplanabilmesi için , elde edilen hedef dağılımı üzerinde görüntü noktalarının yerinin tam olarak belirlenebilmesi için optimizasyon yöntemine ihtiyaç vardır.

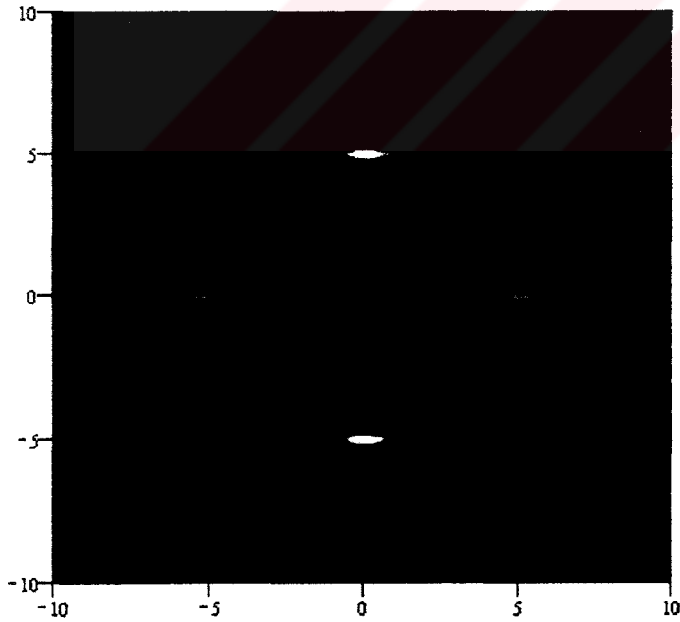


Şekil 4.9 Dört nokta hedef $u \in [0,5]$ ve $n_n = 40$ için elde edilen YAR görüntüsü.

Ancak dört nokta hedef için açıklık boyu $u \in [0,150]$ alındığında sırasıyla $n_n = 10$ ve $n_n = 40$ için Şekil 4.10 ve Şekil 4.11'de gösterilen görüntüler elde edilmiştir. Bu şekillerden de görüldüğü gibi konum hesabı yapabileceğimiz görüntüler elde edilmiştir. Ancak elde edilen görüntüler tek nokta hedef için elde edilen görüntülere göre daha iyi olmamasına karşın, dört nokta hedefin konumlarının belirlenebileceği görüntülerdir.

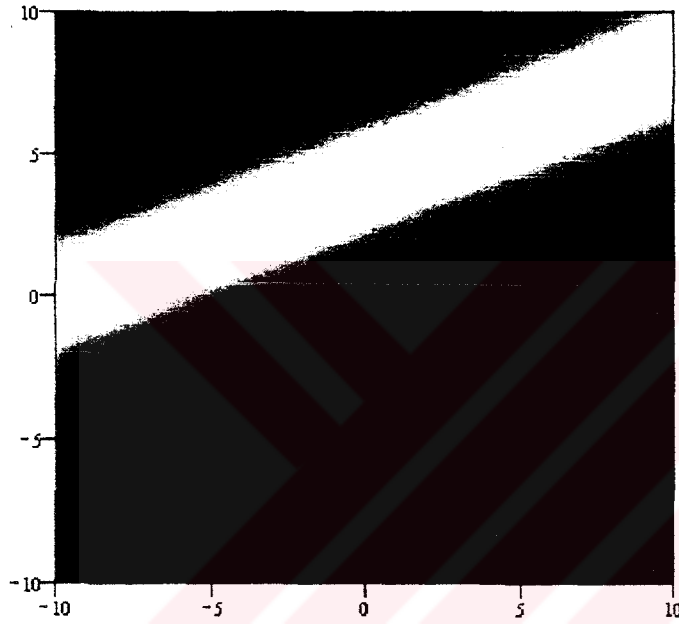


Şekil 4.10 Dört nokta hedef $u \in [0, 150]$ ve $nn = 10$ için elde edilen YAR görüntüsü.



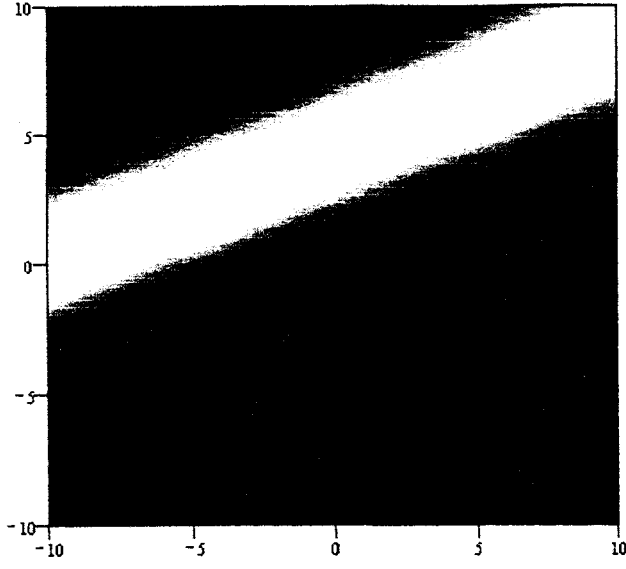
Şekil 4.11 Dört nokta hedef $u \in [0, 150]$ ve $nn = 40$ için elde edilen YAR görüntüsü.

Şekil 4.3'de gösterilen sekiz nokta hedef göz önüne alınan YAR görüntüleme geometresine YAR ters dönüşüm algoritması uygulanmıştır. $u \in [0,5]$ için sırasıyla $nn=10$ ve $nn=40$ için Şekil 4.12 ve Şekil 4.13'de gösterilen görüntüler elde edilmiştir. Ancak elde edilen görüntülerden de görülebileceği gibi nokta hedeflerin konumunun hesaplanabileceği bir görüntü elde edilememiştir. Burada da konumun net olarak hesaplanabilmesi için, elde edilen hedef dağılımı üzerinde görüntü noktalarının yerinin tam olarak belirlenebilmesi için optimizasyon yöntemine ihtiyaç vardır.

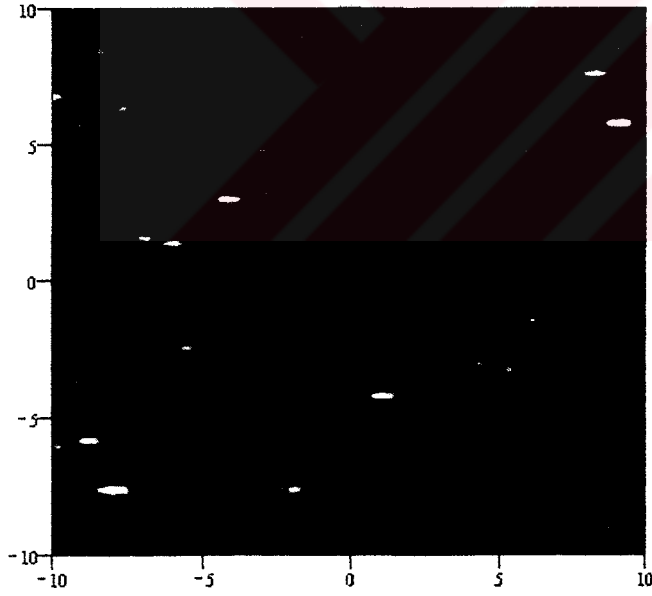


Şekil 4.12 Sekiz nokta hedef $u \in [0,5]$ ve $nn=10$ için elde edilen YAR görüntüsü.

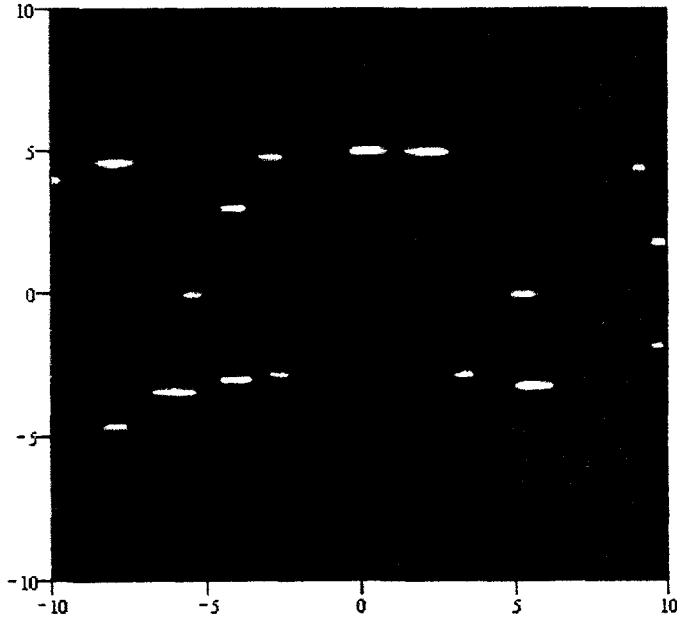
Ancak sekiz nokta hedef için açıklık boyu $u \in [0,150]$ alındığında, sırasıyla $nn=10$ ve $nn=40$ için Şekil 4.14 ve Şekil 4.15 te gösterilen görüntüler elde edilmiştir. Elde edilen görüntülerden anlaşılacağı gibi sekiz nokta hedefin konumlarını belirleyerek, konum hesabı yapmak oldukça güçtür. Çünkü, sekiz nokta hedefin net olarak elde edilen görüntüdeki konumları belirgin değildir. Bu nedenle elde edilen görüntülerdeki nokta hedeflerin genlikleri en yüksek olan noktalar alınarak, nokta hedeflerin konumları hesaplanabilir.



Şekil 4.13 Sekiz nokta hedef $u \in [0,5]$ ve $nn=40$ için elde edilen YAR görüntüsü



Şekil 4.14 Sekiz nokta hedef $u \in [0,150]$ ve $nn=10$ için elde edilen YAR görüntüsü



Şekil 4.15 Sekiz nokta hedef $u \in [0,150]$ ve $n_n=40$ için elde edilen YAR görüntüsü.

YAR ters dönüşüm algoritması ile elde edilen görüntüler genel olarak incelendiğinde , görüntüsünün net olarak elde edilebilmesi için yapay açıklık boyunun büyük , nokta hedef sayısının az ve yapay açıklık boyunca alınan örnek sayısının fazla olması gerekir. Ancak yine de nokta hedeflerin konumunu hesaplayabilmek için elde edilen görüntü üzerinde konum hesaplanması yapmak için optimizasyon algoritmasına gerek vardır.

4.3. Hooke Jeeves Algoritması İle Yapay Açıklıklı Radarda Gerçekleştirilen Uygulama Örnekleri.

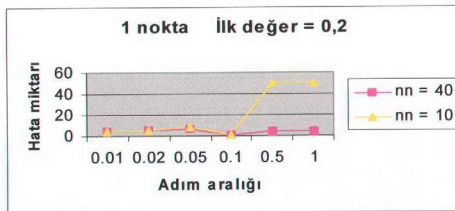
Hooke Jeeves algoritması seçilen uygulama örneklerine, üç farklı hedef dağılımı için uygulanmıştır. Her bir hedef dağılımı için yapay açıklık boyunca alınan örnek sayısı $n_n=10$ ve $n_n=40$ seçilerek algoritma uygulanmıştır.

Hooke Jeeves optimizasyon algoritmasında optimize edilecek parametreler için tanımlanan hata fonksiyonu bu parametrelerin zayıf bir fonksiyonu ise bu optimizasyon algoritmasında seçilen (atanan) ilk değerler büyük önem kazanmaktadır. Bu nedenle nokta hedeflerinin konumunu belirleyen açılar için konum açısına yakın ve uzak iki değer, ilk değer olarak sırasıyla 0,2 radyan ve 0,4 radyan alınmıştır.

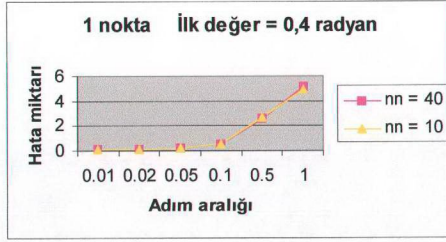
Hooke Jeeves lokal optimizasyon algoritması, tanımlanan hata fonksiyonunun minimumlarından yalnızca birine yakınsamaktadır. Bu minimum yaklaşırken atanan değerlerin hassasiyetini arttırmak için farklı adım aralıkları sırasıyla $H=0.01$, $H=0.02$, $H=0.05$, $H=0.1$, $H=0.5$ ve $H=1$ alınmıştır.

Şekil 4.1'de gösterilen tek nokta hedef için gözüne alınan YAR görüntüleme geometresine Hooke Jeeves algoritması uygulanmıştır. İlk değer 0.2 radyan ve 0.4 radyan için elde edilen hata değerleri, değişik adım aralıkları için sırasıyla Şekil 4.16 ve Şekil 4.17'de verilmiştir. Tek nokta hedefin konumuna yakın alınan ilk değere (0.4 Radyan) ilişkin hata değerleri daha düşük neticeler elde edilmiştir. Şekil 4.17'de elde edilen bu düşük hata değerleri yapay açıklık boyunca alınan örnek sayısı ($nn=40$) daha fazla olduğu durumda elde edilmiştir.

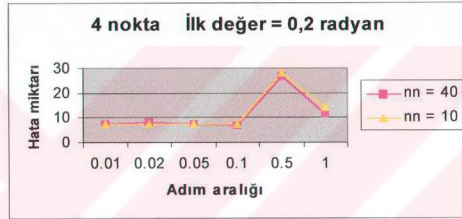
Şekil 4.2'de gösterilen dört nokta hedef için gözüne alınan YAR görüntüleme geometresine Hooke Jeeves algoritması uygulanmıştır. İlk değer 0,2 radyan ve 0,4 radyan için elde edilen hata değerleri, değişik adım aralıkları için sırasıyla Şekil 4.18 ve Şekil 4.19'da verilmiştir. Dört nokta hedef için de hedeflerin konumuna yakın alınan ilk değere (0.4 radyan) ilişkin hata değerleri daha düşük neticeler vermiştir. Şekil 4.19'da gösterilen bu düşük hata değerleri, yapay açıklık boyunca alınan örnek sayısı ($nn=40$) daha fazla olduğu durumda elde edilmiştir.



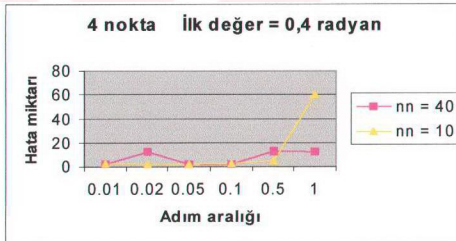
Şekil 4.16 Tek nokta hedef için ilk değer 0.2 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı



Şekil 4.17 Tek nokta hedef için ilk değer 0.4 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı

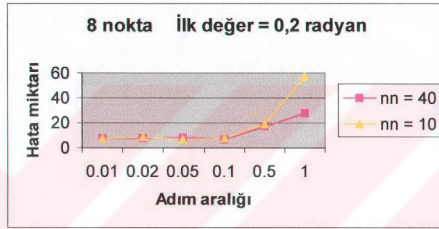


Şekil 4.18 Dört nokta hedef için ilk değer 0.2 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı.

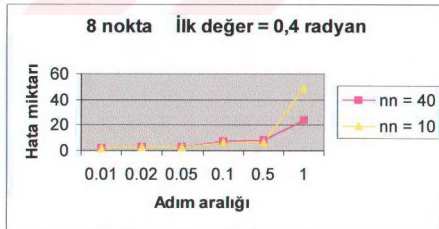


Şekil 4.19 Dört nokta hedef için ilk değer 0.4 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı.

Şekil 4.3'de gösterilen sekiz nokta hedef için gözönüne alınan YAR görüntüleme geometresine Hooke Jeeves algoritması uygulanmıştır İlk değer 0,2 radyan ve 0,4 radyan için elde edilen hata değerleri, değişik adım aralıkları için sırasıyla Şekil 4.20 ve Şekil 4.21'de verilmiştir. Sekiz nokta için de 0,4 radyan için elde edilen hata değerleri daha düşük neticeler vermiştir. Şekil 4.21'de gösterilen bu düşük hata değerleri yine yapay açıklık boyunca alınan örnek sayısı (nn =40) daha fazla olduğu durumda elde edilmiştir.



Şekil 4.20 Sekiz nokta hedef için ilk değer 0.2 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı.



Şekil 4.21 Sekiz nokta hedef için ilk değer 0.4 radyan iken Hooke Jeeves algoritması ile elde edilen hata miktarı.

Ayrıca üç farklı hedef dağılımı için elde edilen hata değerlerini gösteren Şekiller genel olarak incelendiğinde, nokta hedef sayısı arttığında hata değerleride artmaktadır. Genel olarak en iyi hata değerleri tek nokta hedef dağılımı için ilk değer 0,4 radyan ve ($nn=40$) yapay açıklık boyunca alınan örnek sayısının daha fazla olduğu durum için elde edilmiştir. (Şekil 4.17)

4.4. Genetik Algoritma İle Yapay Açıklıklı Radarda Gerçekleştirilen Uygulama Örnekleri

Genetik Algoritma seçilen uygulama örneklerine, üç farklı hedef dağılımı için uygulanmıştır. Her bir hedef dağılımı için yapay açıklık uzunluğu $u \in [0,5]$ yapay açıklık boyunca alınan örnek sayısı $nn=10,20,30$ ve $nn=40$ seçilerek algoritma uygulanmıştır. Menzil için ilk değer 211 m, açı için ilk değer 20° olarak alınmıştır.

Uygulanan algoritmada her anten anında, optimizasyon algoritması ile belirlenen noktasal hedef konumları için hesap edilen işaret ve (3.1) ifadesi ile gerçek hedef konumları için ölçülen yansıyan işaret arasındaki fark hata olarak tanımlanmıştır. Her anten anı için bulunan karesel hataların toplamı, toplam hata olarak alınmıştır. Bu hatayı azaltan hedef açı değerlerine doğrudan algoritma yardımıyla ulaşmaya çalışılmıştır.

Genetik algoritmada kullanılan değişkenler, bit sayısı, kromozom sayısı, iterasyon sayısıdır. Genetik algoritma kullanılarak YAR görüntüleri elde etmek için genetik algoritmaya ait bit sayısı, kromozom sayısı ve iterasyon sayısı, görüntüleme algoritmasına ait ise yapay açıklık boyunca alınan örnek sayıları (anten konumu sayısı) değişken kabul edilmiştir.

Optimizasyon yöntemi olarak kullanılan GA incelendiğinde, değişken parametre olarak alınan bit sayısı doğrudan kullanılan parametrelerin değişim aralığına bağlı olduğu görülür. Öyle ki, bulmak istediğimiz karar değişkeni (θ_i, R_i) değerinin tanımladığı değer aralığına bağlı olarak, seçilen bit sayısı bu aralıktaki hesaplanacak karar değişkeni değerlerinin duyarlılığını belirleyecektir. Yani bit sayısını arttırdıkça karar değişkeninin değişim aralığı içinde alabileceği değer sayısını arttırmış olacağız. Burada önemli bir sınırlama görüntüleme algoritmasının kullanıldığı parametrelere

bağlı olarak, (θ_i, R_i) üzerinde belirlenebilecek çözünürlük değeridir. Öyle ki, bit sayısı ile belirlenen (θ_i, R_i) üzerindeki duyarlılığın, bu çözünürlükten daha iyi olması söz konusu değildir. Dolayısıyla, uygun bit sayısı ile çalışmanın ulaşılabilir minimum hata değeri için yeterli olabileceği anlaşılr.

Kromozom sayısının artmasıyla global minimum bulunma olasılığı da artmakta, fakat buna karşılık olarak minimumu bulmak için daha fazla işlem miktarı gerekmektedir. Bit sayısı (b) , karar değişkeni sayısı (hedef nokta sayısı), (k) olmak üzere $N=b \times k$ ile tanımlanan bir kromozomdaki bit sayısı gözönüne alındığında oluşabilecek durum sayısı 2^N dir. Seçilen kromozom sayısına bağlı olarak, kromozomlar bu küme içinden rastgele belirlenmektedir. Dolayısıyla kromozom sayısı artırıldıkça oluşabilecek durumları içeren kümeden , daha fazla sayıda örnek durum alınmış olacaktır. Aranılan global minimumu bulmak için daha fazla durum sayısı üzerinde inceleme yapmak global minimuma ulaşma olasılığını arttırmakla birlikte , gerekli olan işlem sayısında beraberinde arttıracaktır.

Genetik algoritmada kullanılan üçüncü değişken parametre olan iterasyon sayısının artırılması, yapılan her iterasyonda, başlangıçta seçilen kromozomlar üzerinden yeni kromozomlar (yeni durumlar) oluşturulduğundan, durum kümesi içindeki diğer örnek durumlarla ifade etmemizi sağlayacaktır. Yani global minimuma ulaşmak için algoritmanın kullanacağı durum sayısı artırılmış olmaktadır.

Her iterasyon adımında oluşturulan yeni kromozomlar bir önceki adımda elde edilen kromozomlar içinden en az hatayı verenler seçilerek, bu kromozomlar ile belirlendiğinden minimum hataya ulaşma hızımızı yada olasılığımızı arttıracaktır. Ancak bazı durumlarda, bu yeni kromozomlar bir önceki adımdaki kromozomlardan rastgele bir süreçle belirlendiğinden her zaman için bu yeni elde edilen kromozomların, iterasyon sayısının artması ile daha az hatalı sonuç vermesi beklenmemelidir.

Son olarak YAR parametresi olan yapay açıklık boyunca alınan örnek sayısındaki (nn) değişim , algoritmanın başarısı üzerinde etkili olacaktır. Öyleki, yapay açıklık boyunca alınan örnek sayısının artırılması çapraz çözünürlüğü iyileştirdiğinden, (θ_i, R_i) 'lerin belirlenmesi üzerindeki hatayı azaltması beklenmektedir. Ancak, nn sayısının çok yüksek tutulmasının getireceği ek bilgi yanında, işlem hatalarının

toplamsal etkisi daha yüksek olursa, bu deęişim iyi yönde olmayabilir. Dolayısıyla, nn sayısının uygun deęerde belirlenmeside algoritmanın başarısı açısından önemlidir.

Daha önce tanımlanan sistem geometrisi gözönüne alınarak, algoritma parametreleri için bit sayısı ($b = 5,7,9,11$), kromozom sayısı ($M = 100,200,300$), iterasyon sayısı ($L = 50, 75,100$) ve yapay açıklık boyunca alınan örnek sayısı ($nn = 10,20,30,40$) deęerleri alınarak bu parametre deęerleri YAR algoritması üzerine uygulanmıştır.

Seçilen üç farklı hedef dağılımı için yukarda belirtilen bu parametre deęerlerine karşı algoritmanın gösterdiği cevap, gerçek deęerlerle hesaplanan deęerler arasındaki fark gözönüne alınarak incelenmiştir. Bu fark için tanımlanan karasel hata ile parametrelerin deęişimi arasındaki ilişki Tablo 4.1 Tablo 4.2 ve Tablo 4.3 'de gösterildiği sırada şekillerle verilmiştir. Şekil 4.22, Şekil 4.23, Şekil 4.24 ve Şekil 4.25'de sırasıyla $nn= 10,20,30$ ve $nn=40$ alınarak, farklı iterasyon sayıları için, yatay eksen kromozom sayısını gösterecek şekilde bit sayılarına baęlı olarak, tek nokta hedef dağılımı için hata miktarlarının deęişimi çizilmiştir.

Tablo 4.1 Tek Nokta dağılımı için algoritma parametrelerine göre hata deęişimini gösteren şekiller.

Hata Örnekleri	nn	M	b	L
Şekil 4.22	10	100,200,300	5,7,9,11	50,75,100
Şekil 4.23	20	100,200,300	5,7,9,11	50,75,100
Şekil 4.24	30	100,200,300	5,7,9,11	50,75,100
Şekil 4.25	40	100,200,300	5,7,9,11	50,75,100

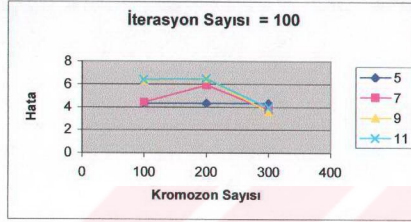
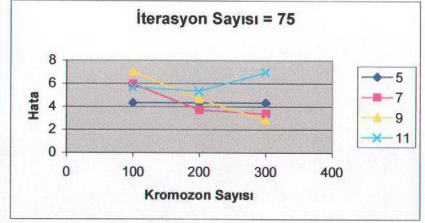
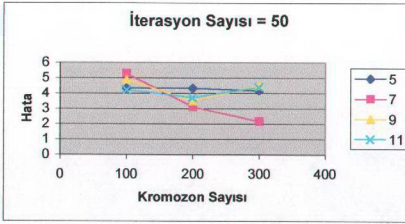
Tek nokta hedef dağılımı için elde edilen sonuçların gösterildiği şekiller incelendiğinde, iterasyon sayısını belli bir deęerden sonra deęiştirmenin hatayı iyileştirici yönde çok etkili olmadığı görülmektedir. Gerçekten belirleyeceğimiz

değişken sayısı için seçtiğimiz iterasyon sayısının en küçük değeri olan $L=50$ için yeterli kalmaktadır. Bu nedenle seçtiğimiz diğer iterasyon büyüklükleri olan $L=75$ ve 100 değerleri için dikkate değer bir iyileşme izlenmemiştir. Tek nokta hedef dağılımı için elde edilen en iyi sonuçlar Şekil 4.22'de iterasyon sayısı $L=50$ için hata miktarı $e=2.17$, Şekil 4.23'te iterasyon sayısı $L=100$ için hata miktarı $e=2.06$ dır. Hata miktarları arasındaki bu farktan da görülebileceği gibi iterasyon sayısını belli bir değerden sonra değiştirmenin hata miktarını iyileştirici yönde bir etkisi olmamaktadır.

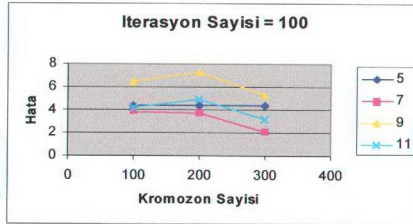
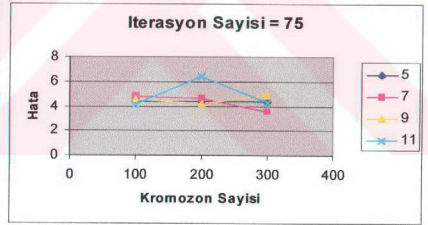
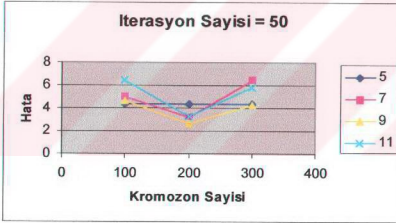
Şekil 4.22 ve Şekil 4.23' ten görülebileceği gibi tek noktalı hedef dağılımı için anten anı sayısı (yapay açıklık boyunca alınan örnek sayısı) az olduğu durumlarda, bit sayısının küçük olduğu değerler daha az hataya sebep olmaktadır. Çünkü tek nokta hedef dağılımı için oluşabilecek durum sayısı (çözüm kümesinin eleman sayısı) az olduğundan bu küme elemanlarını ifade etmek için düşük bit sayısı yeterli olmaktadır. Ayrıca bit sayısının büyümesi kestirimi verilen değerlerdeki duyarlılığı gereğinden fazla büyüteceğinden, optimizasyon işlemi sırasında salınımlar ortaya çıkarmaktadır. Bu durum optimizasyon işleminin yakınsama hızını düşürmektedir. Tablo 4.1'de gösterilen şekillere ilişkin elde edilen sonuçlar incelendiğinde bit sayısının $b=9$ ve $b=11$ olduğu durumlar için bu yakınsamanın izlenemediği görülmektedir. Oysa bit sayısının $b=5$ ve $b=7$ olduğu durumlarda kromozom sayısının değişimi ile birlikte sonuçlardaki hatanın izlenebilir olduğu görülmüştür.

Örnek sayısı $nn=10$ ve $nn=20$ iken kromozom sayısındaki artışın sonuçlardaki hatayı azalttığı elde edilen en iyi sonuçlardan görülmektedir Şekil 4.22 ve Şekil 4.23 için daha önce verilen hata miktarı değerleri kromozom sayısı $M=300$ için elde edilen değerlerdir. Örnek sayısının daha büyük olduğu ($nn=30$ ve $nn=40$) durumlarda, yeterli bilgi miktarına sahip olunduğundan, Şekil 4.24 ve Şekil 4.25'ten görülebileceği gibi kromozom sayısındaki artmanın sonuçlardaki hatayı belirgin bir şekilde değiştirmedeği görülmektedir.

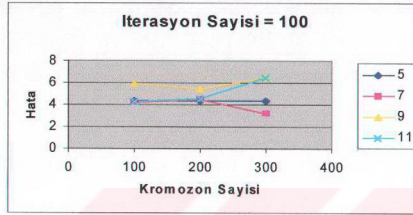
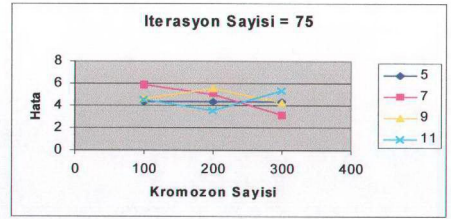
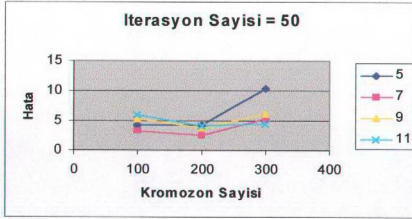
Tek nokta hedef dağılımı için GA ile elde edilen en iyi hata miktarına ilişkin hedef konumunun gerçek hedef konumuna göre durumu, Şekil 4.26'da verilmiştir.



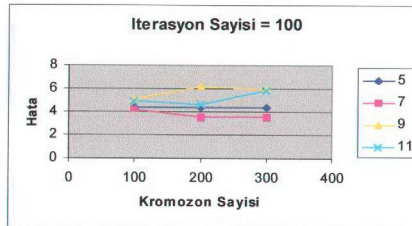
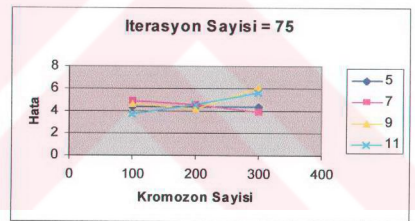
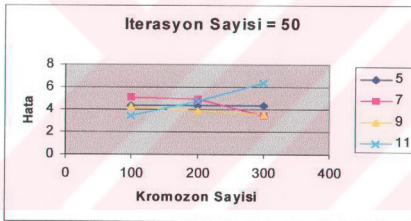
Şekil 4.22 Tek nokta hedef için $nn=10$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.



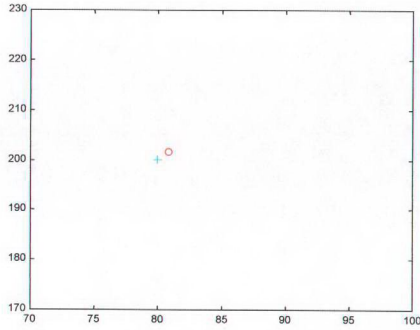
Şekil 4.23 Tek nokta hedef için $nn=20$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.



Şekil 4.24 Tek nokta hedef için $nn=30$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.



Şekil 4.25 Tek nokta hedef için $nn=40$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.



Şekil 4.26 Tek nokta hedef için GA ile elde edilen hedef konumu (0) gerçek hedef konumu (+)

Şekil 4.27, Şekil 4.28, Şekil 4.29 ve Şekil 4.30'de sırasıyla $nn=10,20,30$ ve $nn=40$ alınarak, farklı iterasyon sayıları için, yatay eksen kromozom sayısını gösterecek şekilde bit sayılarına bağlı olarak, dört nokta hedef dağılımı için hata miktarlarının değişimi çizilmiştir.

Tablo 4.2. Dört nokta dağılımı için algoritma parametrelerine göre hata değişimi gösteren şekiller.

Hata Örnekleri	nn	M	b	L
Şekil 4.27	10	100,200,300	5,7,9,11	50,75,100
Şekil 4.28	20	100,200,300	5,7,9,11	50,75,100
Şekil 4.29	30	100,200,300	5,7,9,11	50,75,100
Şekil 4.30	40	100,200,300	5,7,9,11	50,75,100

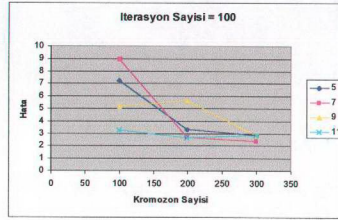
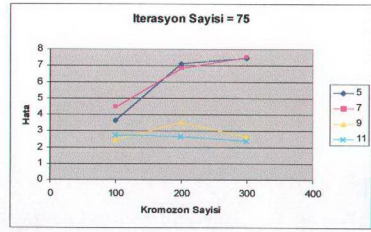
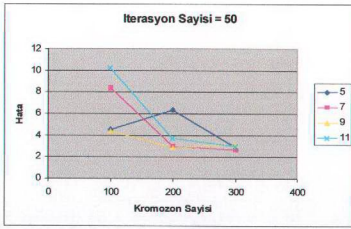
Dört nokta hedef dağılımı için elde edilen sonuçların gösterildiği şekiller incelendiğinde iterasyon sayısını belli bir değerden sonra değiştirmenin hatayı iyileştirici yönde çok etkili olmadığı görülmektedir. Gerçekten belirleyeceğimiz değişken sayısı için seçtiğimiz iterasyon sayısının en küçük değeri olan $L=50$ için yeterli kalmaktadır. Dört nokta hedef dağılımı için iterasyon sayısına göre elde edilen en iyi sonuçlar Şekil 4.29'da iterasyon sayısı $L=75$ için hata miktarı $e=2.04$, Şekil 4.30'da sırasıyla iterasyon sayıları $L=50$ ve $L=75$ için hata miktarları $e=2.61$ ve $e=2.4$ dür. Hata miktarları arasındaki bu farktanda görülebileceği gibi iterasyon sayısını belli bir değerden sonra değiştirmenin, hata miktarını iyileştirici yönde bir etkisi olmamaktadır.

Genel olarak sonuçlar incelendiğinde bit sayısının artmasının hata oranını azalttığı gözlenmiştir. Bit sayısı artırıldığında optimize edilecek parametrelerin değişim aralığı içerisinde kalacak şekilde çözüm uzayında oluşabilecek durum sayısı arttığından minimum hatayı verecek parametre değerlerine yakınsama (gerçek değerlere olabildiğince yakın örnek değerleri oluşturma) mümkün olmaktadır.

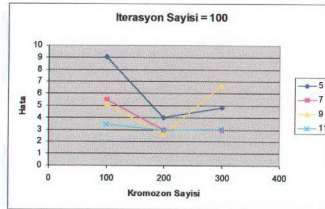
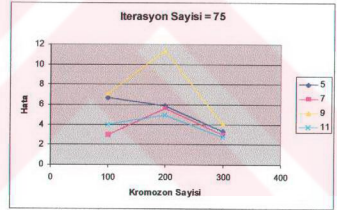
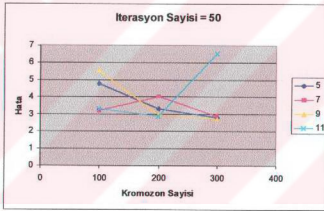
Kromozom sayısı $M=100$ seçildiğinde GA'nın diğer tüm parametrelerinin değişiminin hatayı iyileştirmediği ve hata değerinin büyük çıktığı görülmüştür. Bu durum, Şekil 4.27'de iterasyon sayısı $L=50$, bit sayısı $b=11$ için hata miktarı $e=10.27$, Şekil 4.30'da iterasyon sayısı $L=100$, bit sayısı $b=9$ için hata miktarı $e=10.57$ değerleri ile görülmektedir. Elde edilen sonuçlardan, hata miktarının azaltılabilmesi için, kromozom sayısının $M=100$ 'den büyük tutulması gerekli olduğu görülmüştür. Kromozom sayısının artması, bit sayısında olduğu gibi çözüm uzayında elde edilebilecek durum sayısını arttırdığından beklendiği gibi sonuçlardaki hatayı azaltmaktadır. Örnek sayısı düşük iken ($nn=10$ ve $nn=20$) elde edilebilecek bilgi miktarı yetersiz olduğundan, kromozom sayısını arttırmak, hatanın minimize edilmesinde beklenen etkiyi getirmemiştir. Gerçekten Şekil 4.27 ve Şekil 4.28'deki sonuçlara bakıldığında, kromozom sayısı $M=200$ için elde edilen hata değerleri, kromozom sayısı $M=300$ 'e göre dikkate değer farklılık göstermemektedir. Örnek sayısı arttığında ($nn=30$ ve $nn=40$) elde edilen bilgi miktarı büyüdüğünden, kromozom sayısının $M=200$ olarak alındığı durumda bile hata minimizasyonu başarılabilmektedir. Gerçekten Şekil 4.29 ve Şekil 4.30'daki sonuçlara bakıldığında,

kromozom sayısı $M=200$ için elde edilen sonuçlar, kromozom sayısı $M=300$ deki hata miktarlarına yakın olarak elde edilebilmektedir.

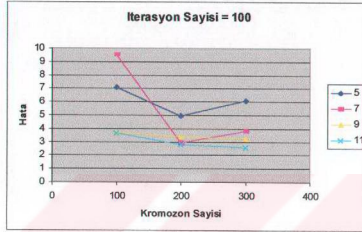
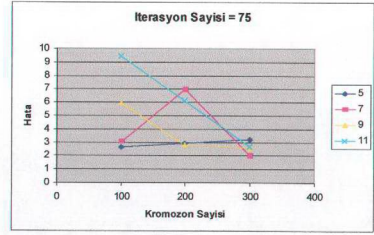
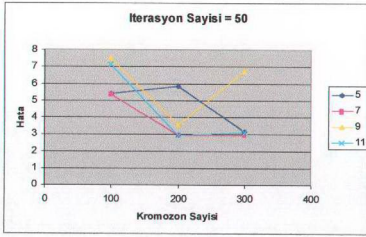
Diğer tüm parametreler sabit kalmak üzere örnek sayısı artırıldığında elde edilen hata değerlerinde belirgin bir değişim görülmemiştir. Ancak, düşük örnek sayıları için kromozom sayısı artırıldığında, gözönüne alınan hedef konumlarının minimum hata ile belirlenebildiği görülmüştür. Bu durum, klasik YAR algoritmasının yetersiz veri eldesi durumlarında GA'nın parametre değişimlerinin sonuçlar üzerindeki hata değerlerinde belirleyici olduğunu göstermektedir. Dört nokta hedef dağılımı için GA ile elde edilen en iyi hata miktarına ilişkin hedef konumunun gerçek hedef konumuna göre durumu Şekil 4.31'de verilmiştir.



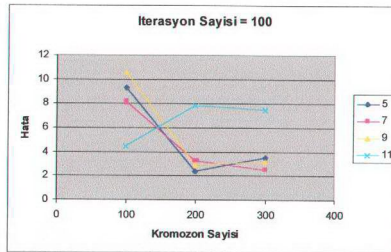
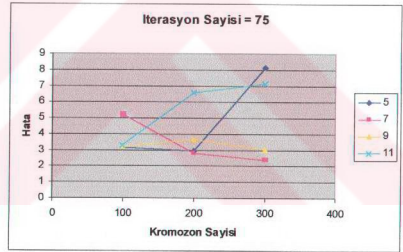
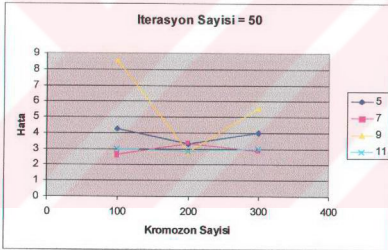
Şekil 4.27 Dört nokta hedef için $nn=10$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.



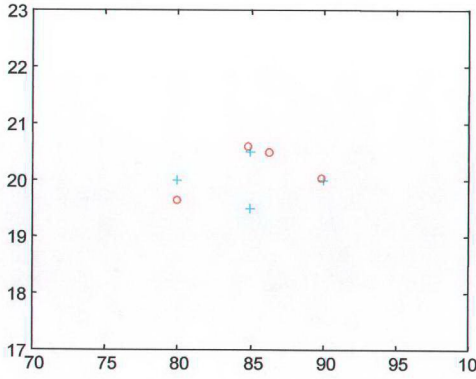
Şekil 4.28 Dört nokta hedef için $nn=20$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.



Şekil 4.29 Dört nokta hedef için $nn=30$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.



Şekil 4.30 Dört nokta hedef için $nn=40$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.



Şekil 4.31 Dört nokta hedef için GA ile elde edilen hedef konumu (o) gerçek hedef konumu (+)

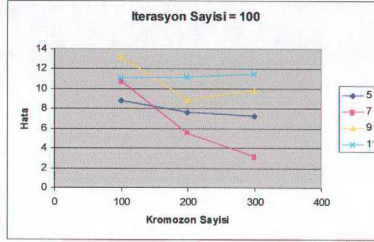
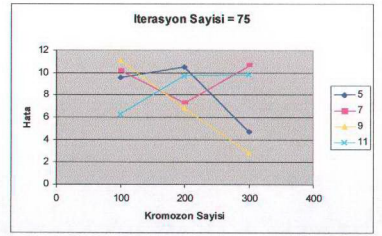
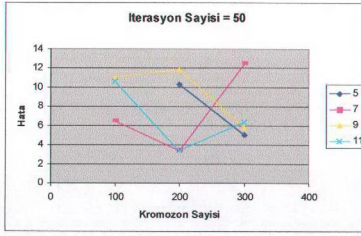
. Şekil 4.32, Şekil 4.33, Şekil 4.34 ve Şekil 4.35'de sırasıyla $nn=10, 20, 30$ ve $nn=40$ alınarak, farklı iterasyon sayıları için, yatay eksen kromozom sayısını gösterecek şekilde bit sayılarına bağlı olarak, sekiz nokta hedef dağılımı için hata miktarlarının değişimi çizilmiştir.

Tablo 4.3. Sekiz nokta dağılımı için algoritma parametrelerine göre hata değişimini gösteren şekil.

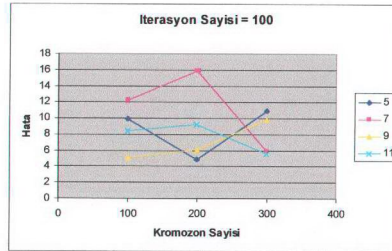
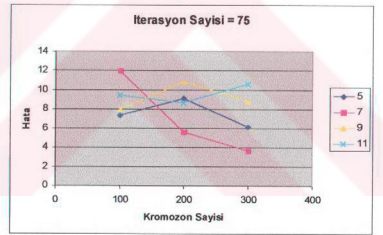
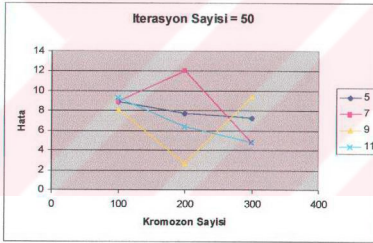
Hata Örnekleri	Nn	M	b	L
Şekil 4.32	10	100,200,300	5,7,9,11	50,75,100
Şekil 4.33	20	100,200,300	5,7,9,11	50,75,100
Şekil 4.34	30	100,200,300	5,7,9,11	50,75,100
Şekil 4.35	40	100,200,300	5,7,9,11	50,75,100

Sekiz nokta hedef dağılımı için elde edilen sonuçlarının gösterildiği şekiller incelendiğinde, iterasyon sayısının değişiminin sonuçlar üzerindeki etkisinin dört noktalı hedef dağılımı için elde edilen sonuçlar ile benzer olduğu görülmüştür.

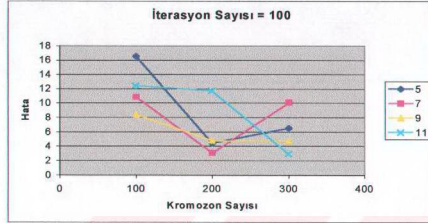
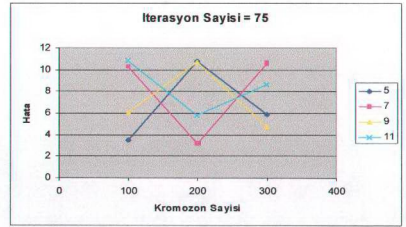
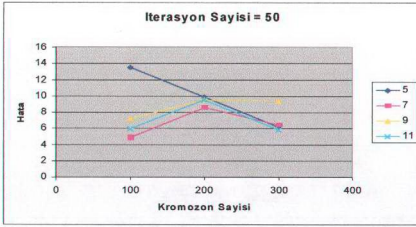
Genel olarak sonuçlar incelendiğinde, noktasal hedef sayısının artması GA'nın hata minimizasyonu üzerindeki başarısını düşürmüştür. Gözönüne alınan GA parametrelerinin değişim aralığı sekiz noktalı hedefin belirlenmesi için yeterli değildir. Bu örnek uygulamada minimizasyon işlemindeki başarının artması için elde edilebilen bilgi miktarının artmasına ihtiyaç duyulmaktadır. Gerçekte klasik YAR algoritması incelendiğinde dört noktalı hedef için yapay açıklık boyu 5 m'den 150 m'ye çıkarıldığında dört noktalı hedefin konumu $nn=40$ için Şekil 4.11'de görüldüğü gibi belirlenebilmektedir. Ancak sekiz noktalı hedef dağılımı için yapay açıklık boyu 150 m ve örnek sayısı $nn=40$ iken bile Şekil 4.15'de görüldüğü gibi doğru olarak belirlenememektedir. Dolayısıyla, bu çalışmada GA'da kullandığımız 5 m'lik yapay açıklık boyu için sekiz noktalı hedef dağılımının, hata minimizasyonunun başarılabilmesi beklenen bir durumdur.



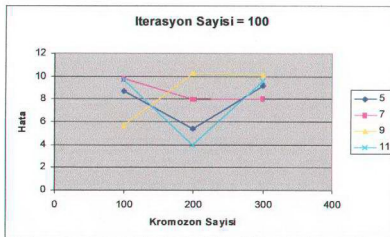
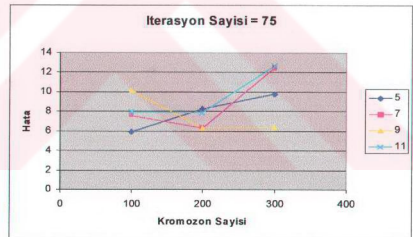
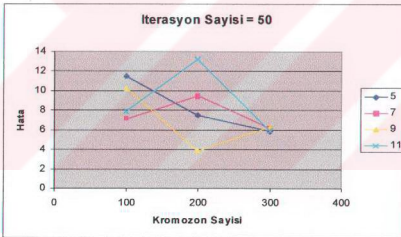
Şekil 4.32 Sekiz nokta hedef için $nn=10$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.



Şekil 4.33 Sekiz nokta hedef için $nn=20$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.



Şekil 4.34 Sekiz nokta hedef için $nn=30$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.



Şekil 4.35 Sekiz nokta hedef için $nn=40$ iken bit ve kromozon sayısına göre genetik algoritma ile elde edilen hata miktarı.

5. SONUÇLAR VE TARTIŞMA

Bu tez çalışmasında, YAR görüntüleme geometrisi gözönüne alınarak, klasik YAR algoritmasında eksik bilgi durumunda, doğru olarak belirlenemediği noktasal hedef konumlarının, seçilen optimizasyon yöntemi ile daha doğru bir şekilde belirlenmesi sağlanmıştır.

Yapay açıklığın oluşturulabilmesi için radar anteninin, hedeflerin tanımlandığı iki boyutlu uzayda y eksenine paralel doğrultuda hareket ettiği gözönüne alınmıştır. YAR'a ait iki boyutlu uzayda radar anteninin yeri tanımlanmış ve YAR'a ilişkin ters dönüşüm algoritması verilmiştir. Ters YAR algoritmasında nokta hedeflerin konumlarını belirlemede kullanılan bir lokal optimizasyon yöntemi Hooke Jeeves ve bir global optimizasyon yöntemi Genetik Algoritma yöntemleri uygulanmıştır. Bu yöntemler Bölüm 4.1'de değişik nokta hedef dağılımlarını içeren, seçilen örneklerle uygulanmıştır.

Klasik YAR ters dönüşüm algoritması ile radar anteninde ölçülen işaret (2.7) bağıntısı kullanılarak, radar anten konumunun değişimine göre simüle edilmiştir. Elde edilen bu işaret kullanılarak, ters dönüşüm algoritmasına ait (2.12) bağıntısı ile hedef dağılımı belirlenmiştir.

Bölüm 4.2'de verilen YAR ters dönüşüm algoritması ile gerçekleştirilen uygulama örnekleri incelendiğinde, Fourier domenî ters dönüşüm algoritması ile elde edilen nokta hedeflerin konumlarının yeterli bir şekilde tespit edilebilmesi için yapay açıklığın ve açıklık boyunca alınan örnek sayısının büyük tutulması gerektiği görülür. Ayrıca tanımlanan hedef dağılımındaki, nokta hedeflerin sayısı arttığında Fourier yöntemine dayalı algoritmanın konum belirleme üzerindeki başarısı düşmektedir. Bu durum Şekil 4.11'de dört nokta hedef, $u \in [0,150]$ ve $nn=40$ için elde edilen görüntü ile Şekil 4.12'de sekiz nokta hedef, $u \in [0,5]$ ve $nn=10$ için elde edilen görüntüler karşılaştırıldığında açıkça görülmektedir. Şekil 4.11'de dört nokta hedef için elde edilen görüntüde, nokta hedeflerin konumları doğru olarak belirlenmiştir. Ancak Şekil 4.12'de sekiz nokta için elde edilen görüntüde, nokta hedeflere ilişkin konumların belirlenmesi mümkün değildir.

Klasik YAR algoritması sekiz nokta hedef için Şekil 4.15'de görüldüğü gibi parametrelerin $u \in [0,150]$ ve $nn=40$ olarak seçilmesi durumunda bile başarılı olamamaktadır. Bu tez çalışmasında kullanılan genetik optimizasyon yönteminin bu nedenle sekiz nokta hedef için $u \in [0,5]$ olması durumunda yetersiz kalması beklenen bir durumdur.

YAR ters dönüşüm algoritması ile gerçekleştirilen uygulamalarda, nokta hedef sayısının ve hedef konumlarının belirlenemediği durumlarda, bu nokta hedeflerin yerlerine ait parametre kestirimleri lokal optimizasyon yöntemi Hooke Jeeves ve global optimizasyon yöntemi Genetik Algoritma kullanılarak gerçekleştirilmiştir. Bu optimizasyon algoritmalarında bilinmeyen parametreler olan nokta hedeflerin hedef merkezlerine göre olan konumları optimizasyon parametresi olarak ele alınmıştır. Öyleki optimizasyon algoritmalarının uygulanmasında, bu hedef dağılımı kullanılarak hesaplanan işaret ile gerçek ölçülen işaret arasındaki fark minimize edilecek şekilde hedeflerin konumları, aç ve menzilleri ile belirlenmiştir.

Bölüm 4.3'de lokal optimizasyon algoritmalarından olan Hooke Jeeves optimizasyon yöntemi ile bilinmeyen parametrelerin kestirimi yapılmıştır. Bu optimizasyon algoritması lokal bir optimizasyon algoritması olduğundan, herhangi bir hata fonksiyonunun var olan minimumlarından bir tanesine yakınsayabilmektedir. Bu nedenle hangi minimumun belirleneceği algoritmadaki ilk değer atamasına sıkı bir şekilde bağlıdır. Ayrıca hata fonksiyonuna ait minimumlardan hangisine ulaşılabileceği optimizasyonun adım büyüklüğü parametresi ile de doğrudan ilişkilidir. Bu bölümde elde edilen sonuçlar incelendiğinde ilk değer atamaları gerçek değere yakinken, yakınsamanın iyi olduğu, aksi durumda ise istenilen değere ulaşamadığı görülmüştür. Hooke Jeeves algoritması lokal minimuma ulaşabildiğinden yapay açıklık boyu büyük tutulsa bile (yapay açıklık boyunca alınan örnek sayısı) yakınsama başarısı yeterli değildir. YAR görüntüleme algoritmasında olduğu gibi burada da nokta hedef sayısı arttığında yakınsama başarısı azalmaktadır. Şekil 4.17'de tek nokta hedef için ilk değer ataması 0.4 radyan (22.93 derece) için elde edilen hata miktarları, Şekil 4.16'da tek nokta hedef için ilk değer ataması 0.2 radyan (11.46 derece) için elde edilen hata miktarlarından daha düşüktür. Yine Şekil 4.17'de gösterilen hata miktarları, Şekil 4.21'de sekiz nokta hedef için ilk değer 0.4 radyan için elde edilen hata miktarlarından da daha düşüktür. Bölüm 4.1'deki seçilen uygulama örneklerinden görülebileceği gibi

0.4 radyan, noktasal hedeflerin konumlarının ortalama değeri için uygun bir değerdir. Bu nedenle en az hata miktarları, tek nokta hedef olması ve ilk değer atamasının gerçek değere yakın olması nedeniyle Şekil 4.17’de elde edilmiştir. Ayrıca, Şekil 4.17 incelendiğinde adım aralığının en hassas alındığı değerde (0.01), en az hata miktarı elde edilmiştir.

Bu tez çalışmasında önerilen Genetik Algoritma ise global bir optimizasyon algoritmasıdır. Bu algoritma ile belirlenen sonuçlar, tanımlanan hata fonksiyonuna ait global minimum değerine yakınsamayı sağlamaktadır. Yakınsama hızı global algoritma olması nedeniyle, düşük olmasına rağmen, nokta hedeflerin konumlarının daha doğru olarak belirlenebilmesini sağlamaktadır.

Bölüm 4.4’de global optimizasyon algoritması olan GA yöntemi ile noktasal hedeflere ait bilinmeyen parametrelerin kestirimi yapılmıştır. Elde edilen sonuçlardan görülebileceği üzere GA’nın kromozom sayısı, bit sayısı, iterasyon sayısı gibi parametrelerin uygun seçilmesi durumunda, gerçek değerlere yakınsanabildiği ve ters YAR algoritmasının başarısının arttığı görülmektedir. GA Şekil 4.26’dan görülebileceği gibi tek nokta hedef dağılımı için gerçek değere çok yakın değerler elde ederken, Şekil 4.5’deki sonuçlar incelendiğinde klasik YAR algoritmasının tek nokta hedef için başarısız kaldığı görülmektedir. Klasik YAR algoritması Şekil 4.9’da görüldüğü gibi dört nokta hedefin belirlenmesi için $u \in [0,5]$ ve $nn=40$ olması durumunda başarısız olmaktadır. Oysaki dört nokta hedef dağılımı için GA Şekil 4.31’den görülebileceği gibi gerçek hedef konumuna çok yakın değerlere ulaşmıştır. Hooke Jeeves algoritması ile elde edilen hata değerleri ile kıyaslandığında, hemen hemen gerçek değere yakın sonuç veren GA’nın çok daha etkin olduğunu görülmektedir. Ayrıca Hooke Jeeves algoritması gibi lokal optimizasyon algoritmalarında adım büyüklüğü ve ilk değer ataması gibi parametreler algoritmanın başarısını belirlerken, GA seçilen ilk değerden bağımsız olarak, yapay açıklığın küçük olduğu ve nokta hedef sayısının fazla olduğu durumlarda bile konuma ait istenilen yeterli çözünürlükleri sağlayabilmektedir. Görüldüğü gibi Fourier uzayında ters dönüşüm algoritmasının kullanıldığı yöntemlerde yapay açıklık boyu ve yapay açıklık boyunca alınan örnek sayısı yöntemin başarısı açısından oldukça önemlidir. Oysa ki, nokta hedef sayısının ve hedef merkezinin belirlenebildiği uygulamalarda klasik bir parametre kestirim problemine dönüşen nokta hedeflerinin yerinin belirlenmesi, bu

alışmada nerilen optimizasyon algoritmaları ile gereklenebilmektedir. Bu algoritmalar bilgi uzayının yeterince doldurulmadığı durumlarda da başarılı olabilmektedir.

İncelenen hedef dağılımı ile ilgili n bilgilerin kullanılabilmesi durumunda GA parametreleri uygun seilerek iřlem zamanının azalması ve yntem başarısının artması saėlanabilir.

KAYNAKLAR

- [1] Skolnik, M.I., 1981. Introduction to Radar Systems, Mc. Graw-Hill, New York.
- [2] Mensa, D.L., 1991. High Resolution Radar Cross Section Imaging, Artech House, Norwood.
- [3] Ausherman, D.A., Kozma, A., Walker, J.L., Jones, H.M. and Poggio, E.C., 1984. Development in Radar imaging, IEEE Trans. On Aerospace and Electronic systems, 4, 363-400.
- [4] Wehner, D.R., 1995. High-Resolution Radar, Artech House, Norwood.
- [5] Carrara, W.G., Goodman, R.S., and Majeski, R.M., 1995. Spotlight Synthetic Aperture Radar, Artech House, Norwood.
- [6] Brown, W, M., and L. J. Pocell, An Introduction to Syntetic Aperture Radar, IEEE Spectrum, September 1969, p. 52.
- [7] Merrill I. Skolnik, Radar Handbook. New York, Mc Graw Hill, 1979.
- [8] Soumekh H. M., 1994. Fourier Array Imaging, PTR Prentice Hall Englewood Cliffs, New Jersey.
- [9] Soumekh H. M., A System Model and Invention for SAR Imaging. In Proc. ICASSP'90, April 1990.
- [10] Soumekh H. M., Bistatic Syntetic Aperture Radar Inversion with Application in Dynamic Object Imaging. IEEE Trans. Signal Process Vol 39, No. 9, September 1991.
- [11] Mital, K. V., Optimization Method in Operations Research and System Analysis, Wiley Eastern Limited, (1987).
- [12] Torn, A., Zilinkas, A., Global Optimization, Springer-Verlag, pp. 18-21, (1989).
- [13] Günel, M.T., 1993 Yeni iki Global Karma Optimizasyon Algoritması ve Bu Algoritmaların Mikrodalga Devrelerin Tasarımına Uygulanması, Doktora Tezi İ.T.Ü Fen Bilimleri Enstitüsü.
- [14] Holland, H., 1975. Adaptation in Naturel and Artificial systems, Ann Arbor, The University of Michigan Press.
- [15] S. Kirkpatrick, C.D. Gellat Jr., and M.P. Vecchi, "Optimization by Simulated Annealing", Science, Vol. 220, No. 4598, pp. 671-680, 1983.
- [16] H.P. Schwefel, "Numerical Optimization of Computer Models", John Wiley, Chichester, 1981.
- [17] D.B. Fogel, System Identification through Simulated Evolution: A machine Learning Approach to Modelling, Ginn Press, Needham Heights, MA 02194, 1991.
- [18] L.J. Fogel, A. J. Owens, and M.J. Walsh, Artificial Intelligence through Simulated Evolution, John Wiley & Sons, New York, 1966.
- [19] K.S. Tang, K.F. Man, and C.Y. Chan, "Fuzzy Control of Water Pressure Using Genetic Algorithm", IFAC Workshop on Safety, Reliability and Applications of Emerging Intelligent Control Technologies, pp. 15-20, Hong-Kong, Dec. 1994.

- [20] **K.S. Tang, K.F. Man, and C.Y. Chan** "Genetic Structure for NN Topology and Weight Optimization", 1st IEEE/IEEE Int. Conf. On GAs in Engineering Systems: Innovations and Application, pp. 250-255, 12-14 Sept., 1995.
- [21] **Y.T. Chan, J.M. Riley and J.B. Plant**, "Modellinf of Time Delay and its Application to Estimation of Nonstationary Delays", IEEE Trans. Acoust. Speech, Signal Processing, Vol. ASSP-29, pp. 577-581, June 1981.
- [22] **N.S. Jayant and P. Noll**, Digital Coding of Waveforms, Prentice Hall, Englewood Cliffs, N.J. 1984.
- [23] **S. Kadambe and G. F. Bondreaux, Bartels**, "Applications of Wavelet Transform for Pitch Dedection of Speech Signals", IEEE Trans. Inform. Theory, Special Issue on Wavelet Transforms and Multiresolution Signal Analysis, I. Daubechies, S. Mallat and A.S: Willsky, Editors, 38 (2), pp. 917-924, 1992.
- [24] **T. Bedford, F. M. Dekking, M. Breeuwer, M.S. Keane, D. Van Schooneveld**, "Fractal Coding of Monochrome Images", Signal Processing Image Communication 6, pp. 405-419, 1994.
- [25] **D.R. Morgan**, "An Analysis of Multiple Correlation Cancellation Loops with a filter in the Auxiliary Path", IEEE Trans. Acoustic, Speech and Signal Processing, ASSP-228, pp. 454-467, 1980.
- [26] **S.J. Elliot and P.A Nelson**, "Active Noise Control", IEEE Signal Processing Magazine, Oct. 1993.
- [27] **Haupt, R.L.:** An Introduction to genetic Algorithms for Electromagnetics. IEEE Antennas and Propagation Magazine, 137 (1995), 7-15.
- [28] **Weile, D.S., Michielssen, E.:** Genetic Algorithm Optimization Applied to Electromagnetics: A review. IEEE Transactions on Antennas and Propagation, 45(1997), 343-353.
- [29] **T. Günel, S. Kent**, "An Optimization Approach for the Synthesis of Microscript Radial Stub", AEÜ (International Journal of Electron. Commun.), Vol.52, No.5, pp. 337-340, 1998.
- [30] **T. Günel**, "A New Approach for the Synthesis of Nonreciprocal and Nonsymmetric Nonuniform Transmission Line Impedance Matching Sections", AEÜ (International Journal of Electron. Commun.), Vol.52, No.4, pp.274-276, 1998.
- [31] **Sedef Kent, Tayfun Günel**, "Dielectric Permittivity Estimation of Cylindrical Objects Using Genetic Algorithm", Journal of Microwave Power and Electromagnetic Energy, Vol.32, No.2, pp.109-113, 1997.
- [32] **Soumekh, M.**, Echo Imaging Using Physical and Synthesized Arrays, Opt. Eng. May 1990, Vol. 29, pp. 545-554.
- [33] **Morse, P., Feshbach, H.**, Methods of Theoretical Physics New York Mc. Graw Hill, 1968.
- [34] **Somuekh, M.** 1994. Fourier Array Imaging, Prentice Hall, New Jersey.
- [35] **Bunday, B.D.**, 1984. Basic Optimization Methods, Edward Arnold, Baltimore.
- [36] **Tavukçu, D.** 2000. MonteCarlo Yönteminin Sayısal integrallere ve Elektromanyetik denklem integrallerine uygulanması, Yüksek Lisans Tezi, İ.T.Ü.

- [37] Hooke, R., Jeeves, T.A., Direct Search Solution of Numerical and Statistical Problems, J. Assn. Comp. Mach., Vol. 8, pp. 212-229, 1961.
- [38] Goldberg, David E., "Genetic Algorithms in Search, Optimization and Machine Learning" pp. 1-18.
- [39] Price, W.L., "Global Optimization by Controlled Random Search", Journal of Optimization Theory and Applications, Vol. 40, No.3, July 1983.
- [40] Davis, L., Ed. 1991. "Handbook of Genetic Algorithms in Control System Engineering."
- [41] Fleming P.J., C.M. Fonseca, "Genetic Algorithms in Control System Engineering."
- [42] Hojjat, A., Hung, S. L., 1995. Machine Learning: neural networks, genetic algorithms and fuzzy systems. New York Wiley.
- [43] Tu M., Gupta I.J., Wolton E.K., 1997. Application of Maximum Likelihood Estimation to Radar Imaging. IEEE Trans. On Ant. And Prop. 45(1), 20-27,

ÖZGEÇMİŞ

Serdar KARGIN, 1962 yılında Ulukışla'da doğmuştur. İlk öğrenimini İzmir Kemal Reis İlkokulunda, orta ve lise öğrenimini İzmir Karataş Lisesinde tamamlamıştır. 1981 yılında İ.T.Ü. Elektronik ve Haberleşme Mühendisliği bölümüne girmiş ve 1985 yılında mezun olmuştur. Öğrenimini Hava Kuvvetleri Komutanlığı namına yaptığı için mezun olduğu yıl Mühendis Teğmen olarak Hava Kuvvetleri Komutanlığında göreve başlamıştır. 1989 yılına kadar Ankara Hava İkmal Bakım Merkezi Fabrika Müdürlüğünde proje subayı olarak çalışmıştır. 1989 yılında İ.T.Ü. Fen Bilimleri Enstitüsünde yüksek lisans programına başlamış ve 1993 yılında yüksek mühendis olarak mezun olmuştur. Aynı yıl İ.T.Ü. Fen Bilimleri Enstitüsü Elektronik ve Haberleşme Mühendisliği Ana Bilim Dalında doktora programına başlamıştır. Halen Hava Harp Okulu Dekanlığında öğretim üyesi olarak görev yapmakta olan Serdar KARGIN iki çocuk babasıdır.