

T.C.  
FIRAT ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

116071

GENETİK ALGORİTMALAR İLE DC-AC ÇEVİRİCİLERDE  
HARMONİK ELİMİNASYONU

Barış GÜRSU

T.C. YÜKSEKÖĞRETİM KURULU  
DOKÜMANTASYON MERKEZİ

YÜKSEK LİSANS TEZİ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

ELAZIĞ

2002

116071

T.C.  
FIRAT ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

GENETİK ALGORİTMALAR İLE DC-AC ÇEVİRİCİLERDE  
HARMONİK ELİMİNASYONU

Barış GÜRSU

YÜKSEK LİSANS TEZİ  
ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Bu Tez, 02-09-2007 Tarihinde, Aşağıda Belirtilen Jüri Tarafından Oybirliği /  
~~Oyçokluğu~~ ile Başarılı /Başarısız Olarak Değerlendirilmiştir.

  
(İmza)

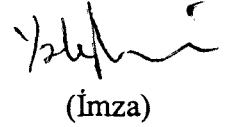
Danışman

Yrd.Doç.Dr.M.Cevdet İNCE

  
(İmza)

Üye

Doç.Dr.Hanifi GÜLDEMİR

  
(İmza)

Üye

Doç.Dr.Yakup DEMİR

Bu tezin kabulü, Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ...../...../.....  
tarih ve ..... sayılı kararıyla onaylanmıştır.

ÖZET

YÜKSEK LİSANS TEZİ

GENETİK ALGORİTMALAR İLE DC-AC ÇEVİRİCİLERDE  
HARMONİK ELİMİNASYONU

Barış GÜRSU

Fırat Üniversitesi

Fen Bilimleri Enstitüsü

Elektrik-Elektronik Mühendisliği Anabilim Dalı

2002, Sayfa:99

Bu tezde, Genetik Algoritma Yöntemi, çaprazlama, mutasyon operatörleri ve gerçekleştirilen programın akış diyagramı verilerek, başlangıç popülasyonunun, uygunluk fonksiyonunun ne olduğu anlatılmıştır.

Ele alınan problem tanımlanmış, DC-AC çeviricilerde harmonik eliminasyonunun gerekliliği üzerinde durulmuştur. Eliminasyonun yapılmadığı durumlarda harmoniklerin olumsuz etkileri ve bu etkileri tanımlamak için kullanılan bazı faktörler verilmiştir. Var olan bu harmonikleri bastırmak için kullanılacak filtreler hakkında kısaca bilgi verilmiştir. Yazılan MATLAB programının temel ilkeleri yarım ve çeyrek dalga simetrisine sahip bir kare dalgada tetikleme açılarının optimum sonucu verecek şekilde nasıl bulunacağı tartışılmış, gerekli matematiksel temeller ortaya konulmuştur.

Problem Genetik Algoritmaya uygulanmış, uygunluk fonksiyonu tanımlanarak programın çalışması adım adım anlatılmıştır. GAHE (Genetik Algoritmalar ile

Harmonik Eliminasyonu) programı ile çözülen problemlerin sonuçları, önceden klasik optimizasyon yöntemleriyle bulunmuş olan bazı çözümlerle ve değişik giriş durumlarında birbirleriyle karşılaştırılarak tablo ve grafik halinde verilmiştir.

Tezde sonuç ve tartışma bölümünden sonra EK-1' de MATLAB ile yazılmış olan GAHE programı verilmiştir.

**ANAHTAR KELİMELER:** Genetik Algoritma, Harmonik Eliminasyonu, MATLAB



**ABSTRACT****MASTERS THESIS****HARMONIC ELIMINATION WITH GENETIC ALGORITHMS OF  
DC-AC CONVERTERS****Bariş GÜRSU****Firat University****Graduate School of Natural and Applied Sciences****Department of Electrical - Electronics Engineering****2002, Page:99**

In this thesis, the Genetic Algorithm Technique, crossing, mutation operators concepts are explained and the initial population and the matched functions are described. The flow chart describing the outline of the technique has also been given in this section.

The necessity of harmonic elimination in DC-AC convertors, the negative effects of the harmonics and factors used to define the harmonic effects are explained. A brief information on filters which are used to suppress the harmonics is given. The developed program for a square wave with half and quarter wave symmetries is introduced.

The Genetic Algorithm Technique is applied for the harmonic elimination. The details of the developed program is explained step by step. The obtained results are given both in tabular and graphical form. The results from Genetic Algorithm are

compared with the results obtained from classical optimization techniques with different initial conditions.

After the result and discussion section in the thesis GAHE (Genetic Algorithm with Harmonic Elimination) program that is written with MATLAB in Addition-1 is given.

**KEY WORDS:** Genetic Algorithm, Harmonic Elimination, MATLAB.



**TEŐEKKÜR**

Bu tez alıőmasında alıőmalarıma yn veren danıőman hocam Sayın Yrd.Do.Dr. Melih Cevdet İNCE' ye teőekkr ederim.



## İÇİNDEKİLER

Sayfa No

<b>1. GENETİK ALGORİTMALAR</b> .....	1
1.1 Giriş .....	1
1.2 Genetik Algoritmaların Çalışma Prensipleri .....	3
1.3 Başlangıç Popülasyonunu Oluşturma .....	7
1.4 Uygunluk Fonksiyonu .....	7
1.5 Seçim ve Kopyalama .....	8
1.6 Çaprazlama .....	9
1.7 Mutasyon .....	10
<b>2. DC-AC ÇEVİRİCİLERDE HARMONİK ELİMİNASYONU</b> .....	11
2.1 Giriş .....	11
2.2 Harmonik Kavramı .....	11
2.3 Harmoniklerle İlgili Bazı Tanımlamalar .....	13
2.3.1 C Faktörü(Crest Factor) .....	14
2.3.2 Toplam Harmonik Distorsiyonu(THD) .....	14
2.3.3 K Faktörü .....	15
2.3.4 Telefon Etki Faktörü .....	16
2.3.5 Motor Bozulma İndeksi .....	16
2.3.6 Distorsiyon Gücü .....	16
2.4 Harmonik Kaynakları .....	17
2.5 Harmonik Filtreleri .....	18
2.5.1 Paralel Filtre Çeşitleri.....	19
2.5.2 Aktif Filtreler .....	20
2.6 Harmoniklerden Kaynaklanan Problemler .....	21
2.6.1 Harmoniklerin Neden Olduğu Ek Isınmalar .....	21
2.6.2 Harmoniklerin Ölçü ve Koruma Düzenlerine Etkileri .....	22
2.6.3 Harmoniklerin Kompanzasyon Sistemine Etkileri .....	23
2.7 DC-AC Çeviricilerde Harmonik Eliminasyon Yöntemi.....	24



<b>3. GENETİK ALGORİTMANIN PROBLEME UYGULANMASI.....</b>	<b>28</b>
3.1 Programın Çalışmasına Örnekler.....	30
3.2 Başka Çözüm Yöntemleriyle Kıyaslama.....	45
3.3 Çeşitli Örnekler.....	49
3.3.1 Harmonik Ağırlıklarının Etkisi.....	49
3.3.2 Eliminasyon İşleminde Açık Sayısının Etkisi.....	54
3.3.3 Farklı Başlangıç Değerlerinin Etkisi.....	58
<b>4. SONUÇ VE TARTIŞMA.....</b>	<b>63</b>
<b>KAYNAKLAR.....</b>	<b>65</b>
<b>EK-1.....</b>	<b>67</b>



## ŞEKİLLER LİSTESİ

	Sayfa
Şekil 1.1 Genetik Algoritma Akış Diyagramı.....	5
Şekil 1.2 Uygunluğun Lineer Ölçeklenmesi.....	6
Şekil 1.3. a) Tek Noktalı, b) İki Noktalı , c) Karışık Çaprazlama Tipleri.....	9
Şekil 1.4. Kromozomdaki Mutasyon.....	10
Şekil 2.1 Ölçme metodu farklı ölçü aletlerinin farklı dalga şekillerinde ölçümleri.....	22
Şekil 2.2 İnverter kutup geriliminin genel şekli.....	25
Şekil 3.1.Örnek 1 Harmonik Spektrumu.....	35
Şekil 3.2 Örnek 2 Harmonik Spektrumu.....	42
Şekil 3.3 Örnek 3 Harmonik Spektrumu.....	44
Şekil 3.4 Örnek 4 Harmonik Spektrumu.....	45
Şekil 3.5 Klasik Optimizasyon Yöntemiyle 5., 7., 11., 13. Harmoniklerin Elenmesi...	46
Şekil 3.6 Genetik Algoritmalar Yöntemiyle 5., 7., 11., 13. Harmoniklerin Elenmesi...	47
Şekil 3.7 Örnek 5 Harmonik Spektrumu.....	50
Şekil 3.8 Örnek 6 Harmonik Spektrumu.....	51
Şekil 3.9 Örnek 7 Harmonik Spektrumu.....	53
Şekil 3.10 Örnek 8 Harmonik Spektrumu.....	54
Şekil 3.11 Örnek 9 Harmonik Spektrumu.....	55
Şekil 3.12 Örnek 10 Harmonik Spektrumu.....	56
Şekil 3.13 Çözüm Açılarından Aynı Olan Açılardan Birini Almadan Hesaplanan Harmonik Spektrumu.....	58
Şekil 3.14 Örnek 11 Harmonik Spektrumu.....	60
Şekil 3.15 Örnek 12 Harmonik Spektrumu.....	61

## TABLOLAR LİSTESİ

	Sayfa
Tablo 3.1 Örnek 1 Başlangıç Generasyonu.....	30
Tablo 3.2 Örnek 1 Çaprazlama Sonucundaki Durum.....	31
Tablo 3.3 Örnek 1 Çaprazlama Sonucunda Üretilen Değerler İle Başlangıç Generasyonundaki Değerlerin Kıyaslanıp En Uygunlarının Seçilmesi (2.Generasyon).....	31
Tablo 3.4 Örnek 1 İkinci Generasyonda Çaprazlama Sonucu Üretilen Değerler.....	32
Tablo 3.5 Örnek 1 İkinci Generasyonda Çaprazlama Sonucunda Üretilen Değerler İle Başlangıç Generasyonundaki Değerlerin Kıyaslanıp En Uygunlarının Seçilmesi (3. Generasyon).....	33
Tablo 3.6 Örnek 1 7. generasyon (Çözüm).....	34
Tablo 3.7 Örnek 1 Çözüm Harmonikleri.....	34
Tablo 3.8 Örnek 2 Başlangıç Generasyonu.....	36
Tablo 3.9 Örnek 2 Çaprazlama Sonucundaki Durum.....	37
Tablo 3.10 Örnek 2 Çaprazlama Sonucunda Üretilen Değerler İle Başlangıç Generasyonundaki Değerlerin Kıyaslanıp En Uygunlarının Seçilmesi.....	38
Tablo 3.11 Örnek 2 İkinci Generasyonda Çaprazlama Sonucu Üretilen Değerler.....	39
Tablo 3.12 Örnek 2 İkinci Generasyonda Çaprazlama Sonucunda Üretilen Değerler İle Başlangıç Generasyonundaki Değerlerin Kıyaslanıp Uygunlarının Seçilmesi (3.Generasyon).....	40
Tablo 3.13 Örnek 2 15. Generasyon (Çözüm).....	41
Tablo 3.14 Örnek 2 Çözüm Harmonikleri.....	41
Tablo 3.15 Örnek 3 Çözüm Harmonikleri.....	43
Tablo 3.16 Örnek 4 Çözüm Harmonikleri.....	45
Tablo 3.17 Klasik Optimizasyon Yöntemi Çözüm Harmonikleri.....	46
Tablo 3.18 Genetik Algoritmalar Yöntemi Çözüm Harmonikleri.....	48
Tablo 3.19 Örnek 5 Çözüm Harmonikleri.....	50
Tablo 3.20 Örnek 6 Çözüm Harmonikleri.....	51

Tablo 3.21 Örnek 7 Çözüm Harmonikleri.....	52
Tablo 3.22 Örnek 8 Çözüm Harmonikleri.....	54
Tablo 3.23 Örnek 9 Çözüm Harmonikleri.....	55
Tablo 3.24 Örnek 10 Çözüm Harmonikleri.....	56
Tablo 3.25 Çözüm Açılarında Aynı Olan Açılardan Birini Almadan Hesaplanan Çözüm Harmonikleri.....	57
Tablo 3.26 Örnek 11 Çözüm Harmonikleri.....	59
Tablo 3.27 Örnek 11 Başlangıç Değerleri.....	59
Tablo 3.28 Örnek 12 Çözüm Harmonikleri.....	60
Tablo 3.29 Örnek 12 Başlangıç Değerleri.....	61



## 1. GENETİK ALGORİTMALAR

### 1.1. Giriş

Yaşayan organizma mükemmel bir problem çözücüdür. Problemlerin çözümünde tabiat bize hayli yardımcı olmaktadır(Karınca koloni sistemi, yapay sinir ağları, genetik algoritma vb.). İnsanlık doğadaki olayları genelde bilinçsiz olarak kopyalayıp buluşlar yapma eğilimindedir. Tıpkı uçmak için kuşları incelemek gibi. Doğadaki sistemleri incelediğimizde bunlar sadece çevreye uyum sağlamakla kalmamış, aynı zamanda optimal seviyeye ulaşmışlardır. Bunun en önemli sebebi 3.5 milyar yıldır doğanın temel kurallarının sabit kalmasıdır. Doğanın amacı uyumdur, en iyileme dolaylı bir sonuçtur.

En elverişli çözümler için doğayı bilinçli bir biçimde taklit etmek özellikle son yıllarda araştırmacıların dikkatini çekmiştir. Bunun nedeni, son yıllara kadar hala birçoğumuzun evrimi bir rastlantılar dizisi olarak kabul etmesidir. Eğer, durum böyle olsaydı olasılık hesaplarına göre canlıların günümüzdeki farklılığa ulaşması gezegenimizin yaşına eşit olurdu(İnsan genomu her biri yaklaşık 300 baz çiftinden oluşan 50000-80000 arası genden oluşmaktadır).

Yaşamın milyonlarca yıldır basit kurallar ve temel maddelerle geliştirdiği oldukça karmaşık sistemler(yani bizler ve diğer canlılar), kendi oluşturduğumuz teknolojilerimizle yaşamı inceliyor ve her geçen gün daha da ilerliyoruz. Bizleri insan yapan kromozomlarımızdaki yaklaşık 3 milyar baz çiftinin taranarak genetik veri tabanlarına aktarıldığı insan genom projesinin tamamlandığı şu günlerde elimizde artık eskiye göre çok daha fazla veri var. Sistemin moleküler düzeyde nasıl çalıştığını anlamaya başlıyoruz. Genetik bitlerimizin(Adenin, Guanin, Sitozin ve Timin), birleşerek nasıl nesnelere(genleri) ve karşılıklı etkileşerek programlarımızı nasıl oluşturduğunu artık anlayabiliyoruz(Örneğin, krebs çemberi). Bunlara ek olarak eskiden sadece fiziksel özelliklerle(reng,boy,şekil v.s) gözlemleyebildiğimiz evrimi, aslında çok daha kesin ve anlaşılabilir bir şekilde moleküler seviyede izleyebiliyoruz. Sadece izlemekle kalmıyor, öğrendiklerimizi kendi teknolojilerimizde de uygulamaya çalışıyoruz. Genetik Algoritmalar(GA), doğadaki evrimsel süreçleri model olarak kullanan bilgisayara dayalı problem çözme teknikleridir. Geleneksel programlama teknikleriyle çözülmesi güç olan, çok boyutlu optimizasyon problemleri bunların

yardımıyla daha kolay ve hızlı olarak çözülebilmektedir. Çok geniş bir çözüm havzasının taranmasını gerektiren problemler, geleneksel yöntemlerle çok uzun sürmekte, GA ile ise kısa bir sürede kabul edilebilir bir sonuç alınabilmektedir.

Evrimin mekanizmasını ve temel kuralları şu şekilde özetleyebiliriz: Evrim, amacı belli olan ya da kontrollü bir süreç değildir, en azından bu şekilde olduğuna dair elimizde geçerli bir kanıt bulunmamaktadır. Kaynakların kısıtlı olduğu değişken bir ortamda, farklı genetik bilgilere sahip canlılar bunlar için yarışır ve kabaca güçlü(fitness) olanlar hayatta kalıp çoğalarak genetik bilgilerini bir sonraki kuşağa(generation) geçirme şansını yakalarlar. Evrimin motoru üreme sonucu oluşan farklılıktır. Üreme iki şekilde olur: Basit canlılarda eşeysizdir. Yani, yeni oluşan canlı bir öncekinin aynısıdır ve farklılık üreme sırasında genlerdeki istemsiz hatalar(mutasyon) sonucu olur. Yüksek canlılarda ise eşeylidir. Yani, değişik genetik malzemeye sahip iki canlı birleşerek tamamen farklı yeni bir canlı oluşturur, farklılık çoğalma sırasında hem mutasyonlar, hem de yeniden birleşimler(rekombinasyon) sonucu olur(Her iki canlının genetik olarak farklı ama işlevsel olarak eş kromozomları üst üste gelerek bilgi değiş tokuşu yaparlar 'cross-over'). Her iki durumda da oluşan farklı canlılardan bir kısmı, bir önceki kuşağa göre değişen çevreye daha iyi uyum gösterip(adaptasyon) daha güçlü olabilir. Bunların yaşaması ve üreyerek soyunu devam ettirmesine doğal seleksiyon denir.

Genetik Algoritmalar ya da daha geniş kapsamıyla Evrimsel Algoritmalarda, evrimsel süreçlerden en çok yeniden birleşme ve güçlülük tabanlı seçme/üreme üzerinde durulur. Ek olarak, arka planda devamlı oluşan mutasyonlar evrimin gereksinimi olan farklılığı daha da artırır. Altını çizmek gerekirse, GA' lar belli bir problem için rast gele çözümler aramak değildir. Bunlar stokastik(verileri bir miktar hata ve belirsizlik içeren problemlerin çözümünün optimizasyon için kullanılan bir teknik) süreçler kullanarak rast gele sonuçlardan çok daha iyi çözümler üretirler.

GA' lar, bahsedilen evrimsel süreçleri kullanan bir makine öğrenmesi modelidir. Bu modelde, makine içerisinde belli kromozomlarla temsil edilen bireylerden oluşan bir popülasyon(topluluk) oluşturulur. Bu bireyler aslında canlılardaki DNA'dan oluşan kromozomlara eşdeğer karakter dizileridir. Bu bireyler simüle edilmiş bir evrim sürecinden geçerler. Basit bit değişimleriyle kolayca mutasyon, yeniden birleşme ya da başka işlemleri gerçekleştirebiliriz. Karakter dizileri belirli uzunlukta

olmalıdır(Canlılarda da kromozomlar belirli uzunluktadır). Uzunluğu belli olmayan karakter dizileriyle geliştirilen Evrimsel Algoritmalar genetik programlama olarak adlandırılır ve GA' lardan farklıdır.

GA' da temel uygulama en uygun bireylerin hayatta kalması fikrini benimseyen evrim süreci için doğal sistemlerde gözlemlenen yöntemlerin taklit edilmesidir. Bireyin bulunduğu ortamda hayatta kalmak için, kendi kendini değiştirerek ortama uygun hale gelmesi ve bu uyum boyunca, yeni üretilecek nesillere, bu özellikler ile birlikte mümkün olabilecek daha çok değişim aktararak, bireylerin daha çok uyumlu hale getirilmesi GA' nın temel felsefesidir. Herhangi bir problemin GA ile çözümü, problemi sanal olarak evrimden geçirmek suretiyle yapılmaktadır.

## 1.2. Genetik Algoritmaların Çalışma Prensipleri

GA' lar temel olarak üç operatöre sahiptir. Üreme, çaprazlama ve mutasyon. GA' nın çalışmasını şu şekilde gösterebiliriz.

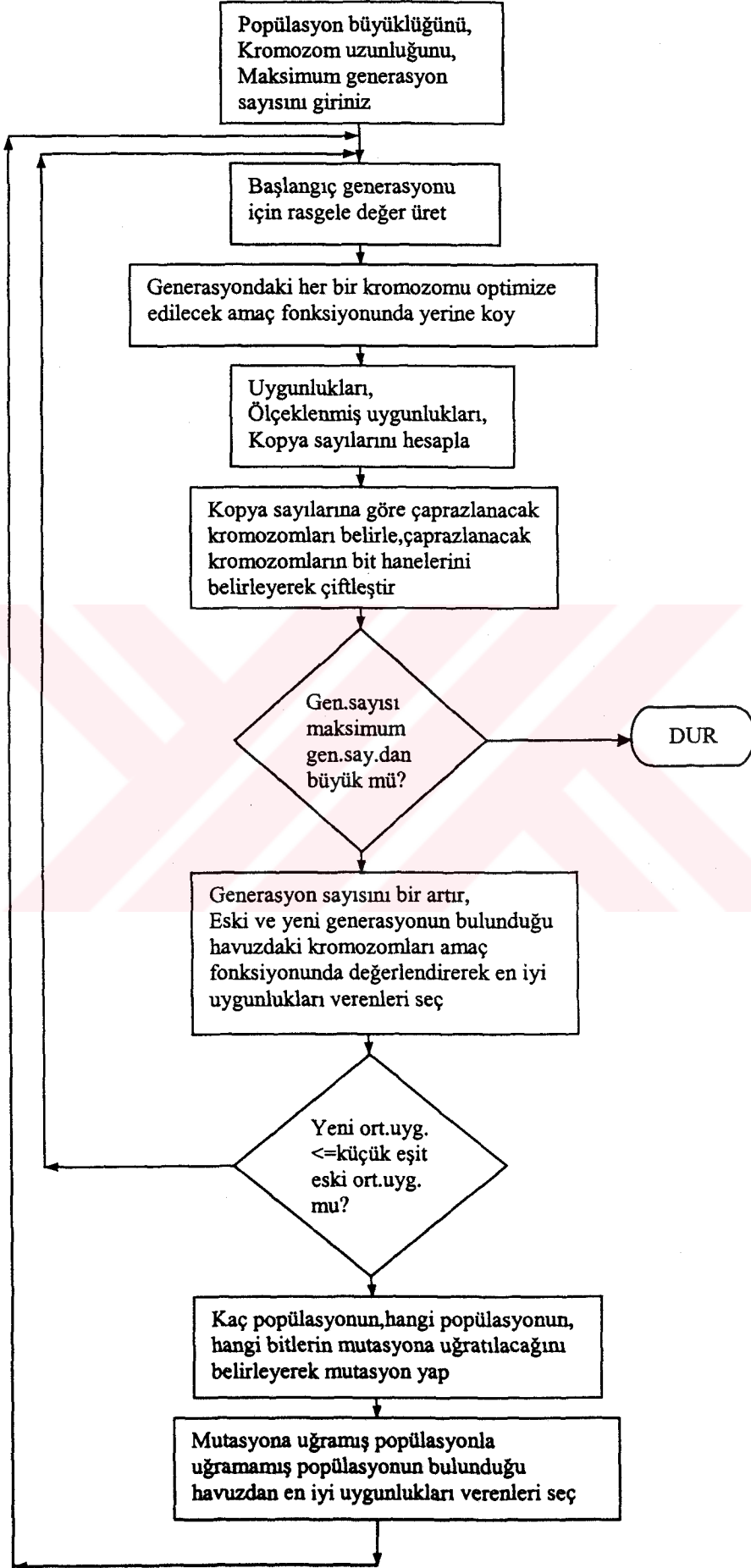
- 1-Çözüm parametrelerini genler şeklinde kodlamak
- 2-Kromozom uzunluğunu belirlemek ve kromozom oluşturmak için genleri dizi haline getirmek
- 3-Popülasyon sayısını belirlemek ve başlangıç popülasyonunu (topluluk) rast gele oluşturmak
- 4-Popülasyona özgü uygunluk (uyumluluk) değerini belirlemek
- 5-Başlangıç generasyonundaki her bir kromozomu uygunluk fonksiyonunda değerlendirip iyiliklerine göre kopya sayılarını hesaplamak
- 6-Kopya sayılarına göre çaprazlanacak kromozomları belirleyip çiftleştirmek
- 7-Çiftleşme sonucu oluşan yeni generasyondaki kromozomlar ile önceki generasyondaki kromozomları uygunluklarına göre tekrar değerlendirip en uygunlarını seçmek
- 8-Yeni generasyondaki ortalama uygunluk, eski generasyondaki ortalama uygunluktan iyi değilse mutasyon işlemine geçmek
- 9-Rast gele seçilen kromozomlarda yapılan mutasyon sonucu oluşan yeni kromozomlarla, mutasyona uğramamış kromozomlardan uygunluğu en iyi olanları seçmek

10-Generasyon sayısı en başta seçilen maksimum generasyon sayısından büyük oluncaya kadar algoritmaya en baştan devam etmek.

Problem için her çözüm bir değerler vektörü ile temsil edilir. Çözüm genellikle ikili kodlanır. Bu kodlama, üzerinde işleyen üreme planını bağımsız hale getirir. Popülasyondaki genetik materyalin yeniden dağıtımı için bir üreme planı (algoritması) seçilir. Çiftleşme amacına göre ilkin ebeveynler seçilir, sonraki generasyon için yavru üretmek üzere birleştirilirler. Çaprazlamada her iki ebeveynin de özelliklerini alan yavrular oluşturmak üzere ebeveynlerin genetik bilgileri birleştirilir. Mutasyon ise araştırmada yeni noktalar bulmak için kullanılan yaygın bir yeniden üretme operatörüdür. Mutasyonda seçilen kromozomun genlerin bazıları rast gele seçilerek bu genler yeniden düzenlenmiş olur. Mutasyonla denge noktasına (en optimum çözüme) yakınsama yavaşlarken algoritmanın lokal minimuma düşmesi engellenmiş olur. (Ataman ve diğ., 1998) İstenilen düzeyde başarılı bir birey bulunana, popülasyon ortalama başarıda artış sağlayamaz bir duruma gelene ya da önceden belirlenen evrim sayısı tamamlanana değin bu süreç devam ettirilir (İnce, 1996).

Genetik Algoritmanın çalışma prensibini açıklayan akış diyagramı şekildeki gibi verilmiştir.





Şekil 1.1 Genetik Algoritma Akış Diyagramı



### 1.3. Başlangıç Popülasyonunu Oluşturma

Algoritma, ilk olarak popülasyon (topluluk) diye tabir edilen bir çözüm seti ile başlatılır. Popülasyon, problemin tipine göre özel bir biçimde kodlanmış olan kromozomlardan oluşur. Her bir kromozom ise problem için bir çözüm teşkil eder. Rastlantısal olarak üretilen başlangıç popülasyonunda bulunacak birey sayısı için bir standart yoktur. Popülasyon büyüklüğü GA' nın performansını etkileyen bir unsurdur. Popülasyon büyüklüğü seçiminde, yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir. Popülasyonu daha da büyültmek çalışma zamanını uzatır, bunun yanında azaltmak ise kromozomların çeşitliliğinde eksiklik göstereceğinden çözümün hassasiyeti azalır. GA dizaynında bir problemin çözümünü bulmaya gerekli olan generasyonların sayısına karşılık, popülasyon büyüklüğünün dengelenmesi gerekir. Örneğin, 100 kromozomlu bir popülasyon, çözümü sadece 10 generasyonla bulabilir. Buna karşılık 20 generasyonda çözümü bulan 20 kromozomlu bir popülasyondan 4 kat daha uzun zaman alabilir. Böylelikle popülasyon boyutu daha küçük olan, diğerine göre çözümü daha fazla generasyon sayısında bulmasına rağmen süre açısından daha verimlidir.

GA uygulamalarında popülasyonu oluşturan kromozomların kodlanması büyük önem taşır. Problemin tipine göre değişebilen kodlama stratejisi, GA' nın etkinliğini ve çalışma hızını etkiler. Çözümün elde edilmesini kolaylaştırdığı için, bu çalışmada binary (ikili) kodlama stratejisi kullanılmıştır. Birey sayısı belirlendikten sonra, toplumu oluşturan çözüm kodları (kromozomlar) rastlantısal olarak oluşturulur.

### 1.4. Uygunluk Fonksiyonu

Kromozomların ne kadar iyi olduğunu bulan fonksiyona uygunluk fonksiyonu (uyumluluk, fitness function) denir. Bu fonksiyon GA' nın beynini oluşturmaktadır. GA' da probleme özel çalışan tek kısım bu fonksiyondur. Başlangıç topluluğu bir kez oluşturulduktan sonra evrim başlar. GA, bireylerin uygunluk ve iyiliklerine göre ayrılıp fark edilmesine gerek duyar. Uygunluk, topluluktaki bir kısım bireyin problemi nasıl çözeceği için iyi bir ölçüdür. Çözüm katarları, ondalık eşdeğerlerine dönüştürülür ve

her aday çözüm, uygunluk fonksiyonuna göre test edilir. Bütün popülasyonun (topluluk) uygunluk fonksiyon değerleri belirlendiğinde sonuçlandırma kriterinin sağlanıp sağlanmadığına bakılır. Bu kriter birkaç şekilde seçilebilir. Bir seçenek, sonlu bir nesil sayısına ulaşıldığında, algoritmayı durdurmak ve elde edilen sonucu en uygun sonuç olarak almaktır. Bir başka seçenek ise, tüm topluluğun ortalama uyumluluk değerinin, en iyi çözümün uyumluluk değerinin belli bir kesri olmasıdır. Eğer bu kriterler sağlanmazsa üç genetik operatörle (seçim ve kopyalama, çaprazlama, mutasyon) işleme devam edilir. Çoğu zaman GA' nın başarısı uyumluluk fonksiyonunun verimli ve hassas olmasına bağlı olmaktadır.

### 1.5. Seçim ve Kopyalama

Seçme operatörü, bir popülasyonda çaprazlamaya uğrayacak olan kromozomların belirlenmesi görevini yerine getirir. Bu operatör, doğal seçme operatörü olarak adlandırılan ve kaliteli olanların hayatta kalmaları prensibine göre çalışan bir elemandır. İyi olan bireyler yaşamını sürdürmeli ve bu bireylerden yeni bireyler oluşmalıdır. Seçim modeli, doğanın hayatta kalabilmek için uygunluk mekanizması modelidir. Kopyalama işleminde bireyler onların uygunluk fonksiyonlarına göre kopya edilirler. Uygunluk fonksiyonu, mümkün olduğu kadar yükseltilmesi gereken bazı faydalı ve iyi ölçülerdir. Topluluk uzayındaki her bir bireyin uygunlukları baz alınarak ne kadar sayıda kopyasının olacağına karar verilir. Kaliteli olanların hayatta kalmaları prensibine göre çalışan bu operatör, en iyi bireylerden daha fazla kopya alırken, en kötü bireylerden kopya almaz.

Seçim ve kopyalama işlemi, topluluk içerisinde uygunlukları düşük olan bireyleri eleyip, elenenler yerine uygunlukları yüksek olan bireylerden daha fazla kopya alarak tamamlanmaktadır.

## 1.6. Çaprazlama

Çaprazlama, eski bireylerden iyi taraflar alınarak elde edilen yeni bireylerin daha iyi olması umuduyla yapılır. Çaprazlama, iki kromozomun bir araya gelerek genetik bilgi değişimi yapmasıdır. GA' nın çok önemli bir elemanıdır. Bu eleman, değişik kromozom konfigürasyonlarının test edilmesine ve hızlı şekilde tekrar yapılanmasına olanak sağlar. Bu sayede, iyi çözümlerden daha iyi çözümler üretmek için değişik yapıların elemanları birleştirilebilir. Topluluktan rastlantısal olarak seçilen iki binary katar arasında çaprazlama yapılır. Basitçe olay, iki ebeveyn kromozomun arasında, gene rastlantısal olarak seçilen genlerin yer değiştirmesidir. Oluşan yeni birey ebeveynlerin bazı özelliklerini almış ve bir bakıma ikisinin kopyası olmuştur. Çaprazlama toplumda çeşitliliği sağlar. İyi özelliklerin bir araya gelmesini kolaylaştırarak en iyiye yaklaşmayı sağlar.

101 10100 → 10100110  
001 00110 → 00110100

(a)

101 101 00 → 10100100  
001 001 10 → 00110110

(b)

10110100 → 00100110  
00100110 → 10110100

(c)

Şekil 1.3. a) Tek Noktalı, b) İki Noktalı , c) Karışık  
Çaprazlama Tipleri

Binary dizilerde çaprazlama, rast gele bir bitin yer değiştirilmesiyle sağlanabilir. Çok düşük bir çaprazlama olasılığı, toplumda bazı özelliklerin kaybolmasına neden olabilir. Bu da en iyi sonuçların bulunmasına engeldir. Ancak, yüksek bir çaprazlama olasılığı da eldeki çözümleri bozarak sonuca ulaşmayı zorlaştırır. İşte problem çözümüne hızlı ve doğru bir biçimde erişebilmek için, çaprazlama için kullanılacak olan operatörün tipinin belirlenmesi oldukça önem taşır. Bu yüzden probleme uygun, tek noktalı(single-point), iki noktalı(double point) ve karışık(shuffle) çaprazlama operatörleri kullanılır. Şekil 1.3'de görüldüğü gibi tek noktalı çaprazlamada bir, iki noktalı çaprazlamada iki çaprazlama noktası belirlenerek, genetik materyaller bu çaprazlama noktalarına göre karşılıklı olarak yer değiştirmektedir. Karışık çaprazlamada ise, kromozomlara ait genler karışık olarak birbiriyle yer değiştirir.

### 1.7. Mutasyon

Sınırlı bir topluluk üzerinde çalışıldığında, toplulukta birkaç genetik bilginin erkenden kaybolma ihtimali bulunmaktadır. Örnek olarak, bir kromozomu oluşturan genlerin tamamı 0 ya da 1 olabilmektedir. Böyle bir kromozomu çaprazlama operatörü ile değiştirmek mümkün olmamaktadır. İşte algoritmadaki son genetik operatör olan mutasyon, çaprazlama ile dikkatli kullanıldığında algoritmanın başarısında temel bir rol oynar. Çaprazlama vasıtasıyla üretilmeyen uygunluk değeri yüksek kromozomları, mutasyon vasıtasıyla üretmek mümkün olmaktadır. Mutasyon, oluşan yeni çözümlerin önceki çözümü kopyalamasını önlemek; yeni, görülmemiş, araştırılmamış çözüm elemanlarını bulmak ve sonuca daha hızlı ulaşmak amacıyla yapılır. Kromozomlar birbirine benzemeye başladığında hala çözüm noktalarının uzağında bulunuyorsa mutasyon işlemi, GA'nın sıkıştığı yerden kurtulmak için tek yoldur. Çalışmanın lokal bir noktada sıkışıp kalmamasını sağlar. Rastlantısal olarak rast gele bir binary katar seçilir ve bu katardaki bir bit 0' dan 1' e ya da 1' den 0' a değiştirilir.

1	0	1	1	0	1	0	0	Eski kromozom
1	0	1	0	0	1	0	0	Yeni kromozom

Şekil 1.4. Kromozomdaki Mutasyon

## 2. DC- AC ÇEVİRİCİLERDE HARMONİK ELİMİNASYONU

### 2.1. Giriş

Eviriciler(inverter) bir doğru gerilim kaynağından etkin değer ve frekansı değiştirilebilen değişken gerilimler oluşturmak için kullanılır.Örneğin, bir endüksiyon motoru, doğru akım kaynağından bir inverter yardımı ile beslenerek ve elde edilen alternatif akımın frekansı istenilen değerlere ayarlanarak çok geniş hız kademeleri arasında çalıştırılabilir. DC-AC çeviricilerin uygulama alanları çok değişik olup,

- Değişken hızlı asenkron motor sürücü sistemleri
  - Kesintisiz güç kaynakları
  - Yedek güç kaynakları
  - Uçaklarda güç kaynakları
  - Yüksek gerilimde d.c. iletim (HVDC) sistemlerinde çıkış katı
  - Endüksiyon ısıtması
- birer örnek sayılabilir.

Uygulamaların çoğunda, çıkış gerilimi ve çıkış frekansını beraber denetlemek gerekir. Örneğin asenkron motorlarda hız denetimi için, frekans düşürülecek olursa manyetik akının sabit kalabilmesi için, gerilimin de düşürülmesi gerekir. Eğer d.c. giriş gerilimi değiştirilebiliyorsa, a.c. çıkış frekansı ile d.c. giriş gerilimi arasındaki oran sabit tutularak yalnız frekans denetlenebilir. Fakat d.c. giriş gerilimi sabitse, d.c. kıyıcıları ile darbe genişlik modülasyonu (DGM) kullanmak gerekir. (Kaynak, 1988)

### 2.2. Harmonik Kavramı

Güç sistemine bağlanan elemanların akım ve gerilim şeklinin sinüsoidal olması istenir. Bu durum sisteme sinüsoidal kaynak ve lineer elemanların bağlanması sonucu elde edilebilir. Günümüzde gittikçe artan sayıdaki nonlineer elemanlar,güç sisteminde



nonsinüsoidal büyüklüklere sebep olurlar. Nonsinüsoidal büyüklüklerin bulunması harmoniklerin güç sisteminde bulunması demektir.

Elektrik güç sistemlerinde enerji kalitesi tanımı, şebekenin gerilim ve frekansındaki değişmeler ile şebekeden çekilen akımdaki dalga şekli bozulmaları ve bunların neden olduğu gerilim dalga şekli bozulmalarının belirtilmesi amacıyla kullanılır. Elektrik enerjisi iletimi ve dağıtımı çoğunlukla radyal hatlarla yapılmaktadır. Uzun radyal hatların empedansı büyük olduğundan, hat sonlarında gerilimin nominal değerinde tutulması zor olmaktadır. Bu tür hatlardan beslenen dinamik yükler gerilimin genlik ve dalga şeklini bozarak enerji kalitesini etkilemektedir ve bunun sonucu olarak komşu diğer tüketicilere olumsuz etki yapmaktadır. Ayrıca bazı tüketicilerin ürettiği harmonikler ile dengesiz yükler sonucu gerilimde oluşan dalga şekli bozulmaları aynı hattan beslenen tüketiciler açısından sorun oluşturmaktadır. Bu nedenle enerji kalitesini etkileyen faktörlerin en önemlisi harmoniklerdir.

Harmoniklerin meydana çıkması enerji sistemleri için istenen bir durum olmamakla beraber çoğu işletmelerde önlem alınmazsa ortaya çıkması kaçınılmazdır. Harmonikler başlıca generatör ve şebeke geriliminin bozulmasına, gerilim düşümünün artmasına, enerji sistemindeki elemanlarda ve yüklerde kayıpların artmasına ve özellikle de rezonans olaylarına sebep olmaktadır. Harmoniklerin sistemde oluşturduğu olumsuz etkilerini gidermek için çeşitli önlemler alınmaktadır. İşletme öncesinde harmonik kaynaklarının tasarımı sırasında alınan önlemlerle ve sonrasında da filtreleme işlemiyle harmoniklerin bu olumsuz etkileri giderilmektedir.

Eviricilerin çıkış gerilimleri kare dalga formundadır. Eviriciden elde edilen gerilim tam sinüs olmadığı için çıkış işaretinin Fourier serisine açılımının belirttiği frekanslarda, belirli genliklerde harmonikler meydana gelecektir. Ortaya çıkan bu harmonikler, işletilen cihazda çoğu kez istenen performansın alınabilmesini engellediği için istenmez. Örneğin eviriciden beslenen motorlarda istenmeyen akım harmonik bileşenleri oluşması sonucunda ek kayıplar, moment dalgalanmaları, verim düşüklüğü gibi problemler ortaya çıkar. Uygulamalarda harmonikleri yok etmek için yüksek dereceden bir filtre kullanılması maliyet bakımından iyi bir çözüm değildir. Bu nedenle evirici devresindeki anahtarlama elemanlarının, tetikleme frekanslarının ve iletim sürelerinin değiştirilmesi ile bazı harmoniklerin yok edilmesi çözümüne gidilir. Bütün harmoniklerin evirici



devresinde yok edilmesi imkansızdır. Bu yüzden çıkışta alçak geçiren filtre devresi kullanılır.

DC-AC çeviricide üretilmiş olan dalgalar şekillerine bağlı olarak farklı harmonik spektrumlarına sahiptir. Elde edilen dalganın bir AC tüketicisi sağlıklı olarak besleyebilmesi için, o tüketicinin tasarlandığı frekanstaki sinüsoidal gibi davranması istenir. Bu nedenle belirli bir kurala bağlı olarak istenmeyen harmonikler bastırılmalıdır.

Eviricilerde çıkış dalgasındaki harmonikleri azaltmak için çeşitli yöntemler kullanılmaktadır. Bunlardan en yaygın olan darbe genişlik modülasyonunun(DGM) yanı sıra darbe genlik modülasyonu ya da çok seviyeli çalışma (Multilevel Operation) da sıkça kullanılan bir yöntem olarak karşımıza çıkmaktadır.

Burada Darbe Genişlik Modülasyonu yöntemiyle harmonik eliminasyonu yöntemi kullanılacaktır.

### 2.3. Harmoniklerle İlgili Bazı Tanımlamalar

Enerji sistemlerindeki harmonik bozulmalar lineer olmayan yükler tarafından meydana getirilir. Lineer olmayan bir yükte, akım ile gerilim arasındaki ilişki lineer değildir. Herhangi bir akım veya gerilimin dalga şekli, frekansı, temel frekansın tam katları olan saf sinüs dalgalarının toplamı şeklinde ifade edilebilir. Fourier teorisine göre, periyodik dalga şekilleri genlikleri ve fazları değişken bir dizi sinüs biçimli dalganın toplamı olarak yazılabilir. Bu dalgalara temel frekansın harmonikleri denir.

$$m(t) = a_0 + \sum_1^n \left[ \sqrt{2} a_n \cos(n\omega t + \theta_n) \right] \quad (2.1)$$

(2.1) denkleminde  $a_n$ , n. harmoniğin genliğini  $\theta_n$ , faz açısını  $\omega$ , ise temel bileşenin frekansını göstermektedir. Enerji sistemleri analizi için genellikle 25 ve 50 den yüksek dereceli harmonikler ihmal edilebilir. (Sisteme bağlı olarak) Bu harmonikler düşük güçlü cihazları etkilemekte, güç sistemlerine etkisi genellikle olmamaktadır. (Gürsoy ve diğ., 1999)

Bir dalga şeklinin harmonik bozulmasını belirten çeşitli tanımlar vardır. Bu tanımlar aşağıda verilmiştir.

### 2.3.1. C Faktörü(Crest Factor)

En basit harmonik tahmin etme yöntemi olan bu katsayı, akım veya gerilimin tepe değeri ile temel bileşenin efektif değeri (True RMS) arasındaki oranı tanımlamak için kullanılır. Dalga şeklinin saf sinüs olması durumunda bu oran 1.414'tür.

$$C \text{ faktörü} = \text{Tepe Değer} / \text{Temel Bileşenin Efektif Değeri} \quad (2.2)$$

### 2.3.2. Toplam Harmonik Distorsiyonu(THD)

Sinüs formundan uzaklaşmış herhangi bir dalga şeklinin harmonik içeriğini bulmak için birçok yöntem vardır. Böyle bir dalga şeklinde toplam harmonik distorsiyonu, akım veya gerilimdeki yüzde olarak harmonik oranını belirtir.

Gerilim için THD,

$$V_{\text{THD}} = \frac{\sqrt{\sum_{n=2}^n V_n^2}}{V_1} \quad (2.3)$$

denklemleri ile verilir.

(2.3) denkleminde,

$V_1$  = rms olarak temel bileşenin gerilimi

$V_n$  = n. Harmonik gerilimi

N = harmonik derecesini göstermektedir.

Gerilim bozulma faktörü, ölçümün bir etkin değeridir ve dalga şekli bozulmuş bir gerilimin bir dirence uygulandığında meydana gelen ek ısınma hakkında fikir vermesi açısından önemlidir. Ancak bir kondansatörün ne kadar zorlandığını belirtmesi açısından iyi değildir. Çünkü zorlanma gerilimin tepe değerine bağlıdır. Harmonik gerilimi, sırasıyla bir direnç(R) yüküne ve seri bir RL yüküne uygularsak;

$$\frac{P_h}{P_f} = (V_{THD})^2 \quad , \quad \frac{P_h}{P_f} < (V_{THD})^2 \quad \text{şeklinde elde edilir.}$$

Burada  $V_{THD}$  , yük uçlarındaki gerilim bozulma faktörüdür. Benzer şekilde akım bozulma faktörü;

$$I_{THD} = \frac{\sqrt{\sum_{n=2}^n I_n^2}}{I_1} \quad (2.4)$$

(2.4) denkleminde,

$I_1$  = rms olarak temel bileşenin akımı

$I_n$  = n.harmonik akımını göstermektedir.

Saf bir direnç yükünde temel güç için harmonik bağıntısı için ise:

$$\frac{P_h}{P_f} = (I_{THD})^2 \quad \text{şeklinde verilir. Burada } V_{THD}, \text{ genellikle sisteme paralel bağlı}$$

çalışan cihazlarda,  $I_{THD}$  ise kesici veya transformatör gibi seri bağlı cihazlarda uygulanır.

### 2.3.3. K Faktörü

K faktörü, harmoniklerden kaynaklanan transformatör kayıplarının belirlenmesinde kullanılan önemli bir katsayıdır. Bu katsayı aynı zamanda transformatörlerin dayanabileceği harmonik akımlarını, harmonik derecesi ve oranı ile birlikte vermektedir.

$$K = \sum_{h=1}^{\infty} I_h (\text{pu})^2 \cdot h^2 \quad (2.5)$$

Doğru akım motorlarının hız kontrolünde ve diğer uygulamalarda kullanılan doğrultucu devreler genellikle karakteristik harmonikler üretirler. Bu harmonikler ile akım bileşenleri doğrultucu türüne göre değişir ve

$$H = kn \pm 1 \quad I_h = \frac{I_1}{h} \quad \text{bağıntısıyla verilir.}$$

Burada,

$h$ = harmonik derecesini

$k$ = sabit sayısı

$n$ = doğrultucu devrenin devre sayısını

$I_1$ = akımın temel bileşenini

$I_h$ = harmonik akımını

gösterir.

#### 2.3.4. Telefon Etki Faktörü

$$TIF = \frac{\sqrt{\sum_{n=2}^{\infty} (V_n \cdot W_n)^2}}{V_1} \quad (2.6)$$

(2.6) denkleminde  $W_n$  değişik frekanslardaki ses etkileri için verilmiştir.

#### 2.3.5. Motor Bozulma İndeksi

Motor bozulma indeksi aşağıda verilen denklem ile ifade edilmektedir. Düşük harmonik derecelerinde bu değer ihmal edilebilir.

$$MDI = \frac{\sqrt{\sum_{n=5}^{\infty} \frac{V_n^2}{\sqrt{n^3}}}}{V_1} \quad (2.7)$$

#### 2.3.6. Distorsiyon Gücü

Akım ve gerilimdeki bozulma, görünür gücü (S) ve reaktif gücü (Q) etkilemektedir. Harmonikli durumda güç sistemlerinde görünür gücü oluşturan aktif güç (P) ve reaktif güç dışında bir güç bileşeni daha ortaya çıkmaktadır. Bu güce distorsiyon gücü

(Distortion Power-D) denilmektedir. Distorsiyon gücünün birimi Voltamperdir. Fakat sistemde güç olarak akmadığı için buna tam olarak güç denilemez. Q, her frekanstaki reaktif güç değerinin toplamı; D ise, hiçbir aktif güç oluşturmayan, değişik frekanslardaki akım ve gerilimlerin vektörel çarpımlarını ifade eder. Buna göre sistemdeki görünür, reaktif, aktif ve distorsiyon (bozulma) güçleri sırasıyla;

$$S = \sqrt{P^2 + Q^2 + D^2} \quad (2.8)$$

$$Q = \sum_{k=0}^{k_{maks}} V_k \cdot I_k \cdot \sin \vartheta_k \quad (2.9)$$

$$P = \sum_{k=0}^{k_{maks}} V_k \cdot I_k \cdot \cos \vartheta_k \quad (2.10)$$

$$D = \sqrt{S^2 - P^2 + Q^2} \quad (2.11)$$

biçiminde verilmektedir. Güç faktörü ise;

$$P_f = \frac{P}{S} \quad (2.12)$$

biçimde verilir. Gerilimin dalga şekli saf sinüsoidal ise,

$$P_f = \frac{I_1}{I_{ef}} \cdot \cos \vartheta_1 \quad (2.13)$$

biçimindedir. Akım ve Gerilimin saf sinüsoidal olması durumunda güç faktörü,

$P_f = \cos \vartheta_1$  biçiminde verilmektedir.

#### 2.4. Harmonik Kaynakları

Şebekede harmoniklerin oluşmasına neden olan lineer olmayan yükleri şöyle sınıflandırabiliriz:

- Doğrultucular,
- Doğru akım motor sürücüleri,
- Alternatif akım motor hız kontrol devreleri,
- Kesintisiz güç kaynakları,

- Ark fırınları,
- Statik reaktif güç kompanzasyonu,
- Frekans dönüştürücüler,
- Statik motor yol alma devreleri,
- Elektronik balastlar,
- Anahtarlama güç kaynakları

Lineer olmayan yüklerin oluşturduğu harmonikler sorunlara yol açarlar. Örneğin;

- Seri ve paralel rezonanslar
- Güç faktörünün kötüleşmesi
- Motor güçlerinin kötüleşmesi
- Kablo ve iletkenlerdeki hatalar
- Simetrik olmayan yükler
- Trafoların ısınması, fazla yüklenmesi
- Sayaçta ortaya çıkan hatalar
- Gerilim bozuklukları
- Telefon kablolarındaki hatalar

## 2.5. Harmonik Filtreleri

Harmonik akımlarının şebekeye geçmesini engelleyen ve bu akımların süzülmesini sağlayan devrelere harmonik filtresi adı verilmektedir. Harmonik filtrelerinin genel amacı bir ya da daha fazla frekanstaki akım veya gerilimlerin harmonik distorsiyonunu azaltmak veya yok etmektir. Harmonik filtreleri pasif ve aktif filtre olmak üzere ikiye ayrılır. Pasif filtreler kondansatör, endüktans ve bazı durumlarda dirençten meydana gelen elemanlar olup kaynak ile alıcı arasına yerleştirilerek temel frekans dışındaki bileşenleri yok ederler. Pasif filtrelerde amaç, yok edilmek istenen harmonik bileşen ile rezonansa gelecek L ve C değerlerini belirlemektir. Her bir harmonik bileşen için onu rezonansa getirecek ayrı bir filtre kolu tasarımı gerekli olup bu işlem en etkin harmonik bileşenler için yapılır. Pasif filtreler kendi içinde seri ve paralel filtre olmak üzere ikiye ayrılır.

Seri filtreler harmonik kaynağıyla şebeke arasına seri olarak bağlanır ve harmonik akışına yüksek empedans gösterirler. Bu yüzden seri filtrelerin ayarlı frekanslarında

yüksek empedansı vardır. Seri filtre belirli bir frekansa ayarlandığı için sadece o ayarlı frekans bileşeni için yüksek empedans gösterir.

Paralel filtreler harmonik kaynağıyla şebeke arasına paralel bağlanırlar. Düşük bir paralel empedans yoluyla istenmeyen harmonik akımların yönünün saptırılması sağlanır. Paralel filtreler harmonik akımlarına çok düşük bir empedans göstermek için tasarlanır. Ayrıca paralel filtreler temel frekansta reaktif güç sağlayarak güç faktörü düzeltiminde de kullanılır. Paralel filtre kullanımının en büyük sakıncası güç sistemiyle rezonansa girmesidir. Bu durum tasarım esnasında filtre bileşenlerinin değerlerinin ayarlanması ile ortadan kaldırılabilir.

### 2.5.1. Paralel Filtre Çeşitleri

Paralel filtreler kendi içerisinde tek ayarlı, çift ayarlı ve yüksek geçiren sönümlü filtreler olmak üzere üç kısma ayrılır.

a- Tek ayarlı filtreler, kısa devre yolu oluşturarak istenen özel bir frekanstaki harmonik akımının hattan saptırılmasını sağlarlar. Bu işlemi genellikle tek bir frekans değeri için yaparlar. Seri R-L-C devresinden meydana gelmektedirler. Bazen R direnci ilave edilmeksizin filtreleme işlemi tek ayarlı L-C filtresiyle gerçekleştirilebilir.

Filtre bileşenlerinin tayinine geçmeden önce rezonans olayına kısaca değinmek gerekmektedir. Nonlineer yüklerin ürettiği harmonik frekanslarından birinde veya yakınında filtre bileşenlerinden kondansatör grubunun kapasitif reaktansı ile sistemin endüktif reaktansı eşit olduğunda paralel rezonans meydana gelir. Bu durum kondansatör uçlarındaki gerilimin aşırı yükselmesine sebep olmakta ve kondansatöre zarar vermektedir. Böyle bir olayın meydana gelmesini önlemek bakımından filtrenin seri rezonans frekansı veya ayarlı frekansı paralel rezonansın meydana geldiği harmonik frekansından %3 ile %10' dan daha düşük seçilir. Örneğin 6 darbeli güç konverterinde 5. Harmonik frekansında paralel rezonans meydana geliyorsa 60 Hz' lik sistemde 300 Hz' lik frekansa karşılık gelir. Bu sisteme uygulanan filtre 270 ile 290 Hz arasında ayarlı frekansa sahip olacaktır. Bu durumdaki filtrenin ayarlı frekansı 282 Hz' dir, yani 4.. 7. Harmoniğe karşılık gelir.

Tek ayarlı filtre için filtre bileşenlerinin tayininde şu eşitlikleri verebiliriz.

$$\omega_n = 2\pi f_n \quad (2.14)$$

$$\omega_n = 2\pi(nf_0) \quad (2.15)$$

(2.14) ve (2.15) denklemlerinde,

$\omega_n$  = n.harmonik açısal frekansı

$f_0$  = güç sisteminin temel frekansı

$f_n$  = filtrenin n.harmoniğe ayarlı frekansıdır.

$C_f$  Filtrenin kapasitesi ve  $L_f$  filtrenin endüktansı olmak üzere, ayarlanmış frekansta filtrenin kapasitif ve endüktif reaktansları eşit olacaktır.

$$\frac{X_{C_f}}{n} = n \cdot X_{L_f} \quad (2.16)$$

Burada  $\omega_0$  sistemin açısal frekansı olmak üzere yazılabilir.

$$\frac{1}{n \cdot \omega_0 \cdot C_f} = n \cdot \omega_0 \cdot L_f \quad (2.17)$$

yazılabilir.

Bu eşitlik kullanılarak filtrenin  $L_f$  ve  $C_f$  değerleri belirlenebilir.

b- Çift ayarlı filtrenin empedansı rezonans frekansları yakınında iki tek ayarlı filtrenin eşdeğer empedansı ile pratik olarak aynıdır. Çift ayarlı filtre iki harmoniği elimine edebilir. Tek ayarlı filtre ile karşılaştırıldığında temel frekanstaki güç kaybının azlığı bu filtrelerin en önemli özelliğidir.

c- Yüksek geçiren sönümlü filtreler birinci, ikinci, üçüncü dereceden ve C tipi filtre devreleri olmak üzere dört çeşittir. Bu tip filtrelerde harmonik derecesi(frekans değeri) arttıkça empedans değerleri düşüş gösterir.

### 2.5.2. Aktif Filtreler

Harmoniklerin filtrelenmesi için pasif filtrelerin kullanılmasında karşılaşılan en önemli iki problem vardır. Birincisi, pasif filtrelerin mevcut dağıtım şebekesi için özel olması sebebiyle mevcut dağıtıma ekler gelmesi veya yüklerin artması durumunda ilk yatırımı geçersiz kılması; ikincisi ise, sisteme montajlarının oldukça zor olmasıdır. Pasif filtrelerin bu olumsuzluklarının giderilmesi için tercih edilen aktif filtreler yük tarafından çekilen harmonikleri analiz ederler ve uygun bir fazda yüke aynı harmoniği



ters fazda enjekte ederler. Böylece istenilen noktada harmonik akımlarının sınırlandırılmasına yani harmoniklerin şebekeden çekilmemesine yararlar. Güç elektroniği elemanları ve uygulamaları geliştikçe aktif filtreler de gelişmektedir. Bununla birlikte aktif filtreler içerisinde kullanılan güç elektroniği elemanlarının pahalı olmasından dolayı pasif filtrelere nazaran ekonomik değildir.

## 2.6. Harmoniklerden Kaynaklanan Problemler

### 2.6.1. Harmoniklerin Neden Olduğu Ek Isınmalar

Harmonikli akımlar, iletkenlerin ısınmasına iki şekilde etki eder. Birincisi, harmonikler nedeniyle akımın efektif değerinin artması,  $I^2R$  ile orantılı joule kayıplarının artmasıdır. İkincisi, iletkenlerdeki akım dağılımının değişimi nedeniyle oluşan deri ve yakınlık etkisi ile iletken direncinin (R) artmasıdır.

Transformatördeki kayıplar boşa çalışma ve yüke bağlı kayıpları içerir. Bakır kayıpları, kablolarda olduğu gibi, harmoniklerin akımın efektif değerini arttırması sonucu artar. Kaçak akının transformatör kazanında ve sargılarda meydana getirdiği girdap (fuko, eddy current) akımları nedeniyle de kayıp oluşur. Transformatörde girdap akımlarından kaynaklanan kayıp, harmonik akım ve bu akımın harmonik mertebesinin karelerinin çarpımıyla orantılıdır. Yüksek frekanslı harmonik akımlarında, transformatör nüvesinde meydana gelen histerezis kayıpları da artar. Bütün bu kayıplar transformatörde ısınmaya, aşırı yüklenmeye ve transformatör ömrünün kılmasına neden olmaktadır.

Ayrıca gerilim dalga şeklindeki bozulma kondansatörlerde,

$$\sum_{n=1}^{\infty} C.(\tan \delta).w_n.V_n^2 \quad (2.18)$$

ifadesi ile verilen ek ısınmaya sebep olmaktadır. Burada  $\tan \delta = R/(1/wC)$  kayıp faktörü,  $w_n = 2\pi f_n$  ve  $V_n$ , n. harmoniğin efektif değeridir.

## 2.6.2. Harmoniklerin Ölçü ve Koruma Düzenlerine Etkileri

Akımdaki harmonik bozulması, anahtarın akım kesme yeteneğini etkilemektedir. Bozulma, akımın sıfır geçişinde temel frekanstaki, normal sinüs değerine göre daha yüksek bir değişim hızına yol açabilmekte, bu ise kesmeyi zorlaştırmaktadır. Anahtarların çalışmasındaki aksaklık elektromanyetik endüksiyon bobinlerinin harmoniklerden dolayı hatalı çalışmasından kaynaklanmaktadır. Mesela sigortaların çalışma karakteristikleri de harmonik akımlarının meydana getirdiği ısınma nedeni ile değişmektedir. Şekil 2.1’de değişik ölçme aygıtlarının çeşitli dalga şekillerindeki ölçüm sonuçları verilmiştir.

	Ölçüm Tipi		
	Gerçek RMS	Tepe Değer Metodu	Ortalama Değer Metodu
	Hesap Yöntemi		
	RMS	Tepe Değer/1,44	Sinüs Ort 1,11
Sinüs	100%	100%	100%
Kare	100%	82%	110%
Üçgen	100%	121%	96%
Ayarlı Hız Sürücüsü Dalgası	100%	127%	86%
Bilgisayar Akım Dalga Şekli	100%	184%	60%
Aydınlatma Dimmer Dalgası	100%	113%	84%

Şekil 2.1 Ölçme metodu farklı ölçü aletlerinin farklı dalga şekillerinde ölçümleri

Burada kullanılan etkin değer ölçüm metodları;

**Gerçek RMS:** Bir işaretin efektif değeri, bir dirence uygulanan gerilim sonunda oluşan ısının ölçüsü olarak verilmektedir.

**Tepe Değer Yöntemi:** Bu yöntemde işaretin sinüsoidal olduğu kabul edilerek, ölçme cihazı, işaretin tepe değerini alıp etkin değerini elde etmek için 1.414 ile bölmektedir.

**Ortalama Değer Yöntemi:** Cihaz, işaretin ortalama değerini almaktadır. Saf sinüsoidal işaret için bu ortalama değer, bir katsayı ile etkin değere eşitlenmektedir.

Bütün bu değişik yöntemler, saf sinüs için aynı değeri verse de, bozulmuş bir işaret için farklı sonuçlar vermektedir. Uygulamada faz ve nötr akımlarının dalga şekillerinde harmonik akımlar nedeniyle bozulma olduğundan bu akımların doğru olarak ölçülmesi önemli olmaktadır.

Harmoniklerin ölçümü için en iyi yöntemlerden birisi de osilaskop kullanmaktır. Osiloskop ekranından akım ve gerilime ait dalga şekilleri görülmesine karşın bu şekiller sistemdeki mevcut harmonikler hakkında bilgi vermezler. Bunun için harmonik çözümleyici kullanmak gerekir. Harmonik çözümleyici, dalgayı harmonik bileşenlerine ayırır ve büyüklüğü ile birlikte her bir harmoniğin temel frekansla olan faz farkını vermektedir. Bundan başka sistemin harmonik toleranslarının belirlenmesi için çok önemli olan toplam harmonik bozulmayı vermektedir.

### 2.6.3. Harmoniklerin Kompanzasyon Sistemine Etkileri

Güç faktörünü düzetmek için kullanılan kondansatörler, harmonik frekanslarında seri veya paralel rezonans olayına sebep olabilirler. Bu durumda aşırı akım ve gerilimleri oluşur.

Bir alçak gerilim enerji sisteminde rezonans frekansı;

$$n = \sqrt{\frac{Q_s}{Q_c}} \quad (2.19)$$

Burada n, rezonansın meydana gelebileceği harmoniğin derecesini;  $Q_s$ , sistemin kondansatör grubunun bulunduğu noktadaki kısa devre gücünü;  $Q_c$  kondansatör gücünü göstermektedir.

Harmonikli sistemlerde kompanzasyon için kullanılan kondansatörler harmonikler göz önüne alınarak aynı zamanda filtre olarak tasarlanmalıdır. Sistem

harmonik frekanslarına uygun olarak tasarlanmış bir harmonik akım filtresi, harmonik bozulmayı azaltıp, gerilim ve güç faktörünün düzeltilmesini sağlar (Gürsoy ve diğ., 1999)

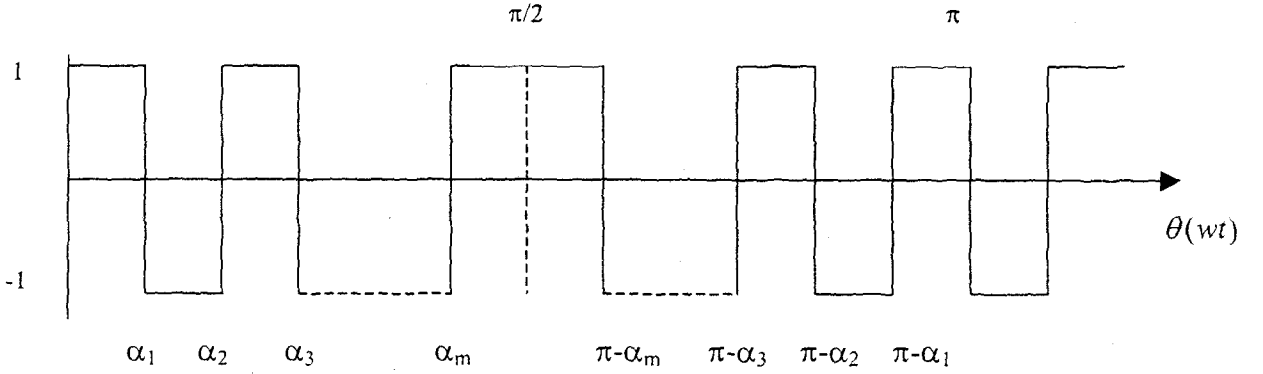
Girişteki gerilimin büyüklüğünü ve/veya frekansını kontrol etmek için kullanılan anahtarlama elemanlarının hızlı kontrolü ile, anahtarlanarak kontrol edilen çıkış geriliminin dalga şeklinde harmoniklerin ve dolayısıyla ilave kayıpların oluşmasına neden olur ve bu da sistem güç faktörünün azalmasına yol açar. Alternatif akım makineleri gibi nonlineer yüklerde bu harmonikler gürültü titreşim ve enerji kayıplarına yol açar

DC-AC çevirici çıkışında istenmeyen harmonikler, harmonik eliminasyon yöntemiyle ortadan kaldırılabilir.

## **2.7. DC- AC Çeviricilerde Harmonik Eliminasyonu Yöntemi**

Bu yöntem DGM'lu (Darbe Genişlik Modülasyon) inverter çıkış gerilimindeki istenmeyen harmonikleri eleyen ve ana harmoniğin (temel bileşenin) genliğini kontrol edilebilen anahtarlama açılarının önceden hesap edilmesi esasına dayanır. Dolayısıyla elenecek harmonik sayısı ile anahtarlama sayısı arasında kesin bir ilgi vardır. (Tatar, 1994)

Yarım ve çeyrek dalga simetrisine sahip, genelleştirilmiş bir DGM'lu tek fazlı inverter kutup gerilimi Şekil 2.2'deki gibidir. Burada  $\alpha_1, \alpha_3$  gibi tek sayı indisli anahtarlama açıları negatife geçiş,  $\alpha_2, \alpha_4$  gibi çift sayı indisli anahtarlama açıları da pozitive geçiş açıları olarak tanımlanır. Bu işaret ( $f(\theta)$ ), Fourier serisine açılırsa, dalga şeklinin simetrilüğünden dolayı sadece tek sayılı sinüslü bileşenlerden oluştuğu görülür.



Şekil 2.2 İverter kutup geriliminin genel şekli

Elimine edilebilecek harmonik sayısı, hesaplanması gereken  $\alpha$  sayısından bir eksiktir. Örneğin sadece 5. ve 7. harmonikler elimine edilmek istenirse  $\alpha_1, \alpha_2$  ve  $\alpha_3$  hesaplanır ve bir yarım çevrimde 6 aktarım gerçekleştirilir.

Bu işaretin harmonik bileşenleri fourier analizi ile hesaplanır. Genel olarak fourier denklemini;

$$f(\theta) = \frac{a_0}{2} + \sum_{n=1}^{\infty} [(a_n \cos n\theta) + (b_n \sin n\theta)] \quad (2.20)$$

$$a_0 = \frac{2}{T} \int_0^T f(\theta) d\theta \quad (2.21)$$

$$b_n = \frac{2}{T} \int_0^T f(\theta) \cos n\theta d\theta \quad (2.22)$$

$$a_n = \frac{2}{T} \int_0^T f(\theta) \sin n\theta d\theta \quad (2.23)$$

Buradan hareketle, çıkış geriliminin temel harmoniğini kontrol eden ve aynı zamanda istenmeyen harmonik bileşenlerini de yok eden anahtarlama açılarının hesabı iki aşamada gerçekleştirilir. Birinci aşama temel bileşeni maksimum yapan  $m$  tane anahtarlama açısının bulunmasıdır. İkinci aşama da anahtarlama açılarının aynı zamanda istenmeyen  $m - 1$  tane harmonik bileşenini yok etmesidir.

Çeyrek dalga simetrisinden dolayı  $b_n = 0$ ;

$$a_n = \frac{2}{2\pi} \int_0^{2\pi} \text{Sinn}\theta d\theta$$

Dalga şekli  $\pi/2$  etrafında simetrik olduğundan;

$$a_n = \frac{4}{\pi} \left[ \int_0^{\alpha_1} \text{Sinn}\theta d\theta - \int_{\alpha_1}^{\alpha_2} \text{Sinn}\theta d\theta + \int_{\alpha_2}^{\alpha_3} \text{Sinn}\theta d\theta - \int_{\alpha_3}^{\alpha_4} \text{Sinn}\theta d\theta + \int_{\alpha_4}^{\pi/2} \text{Sinn}\theta d\theta + \dots \right] \quad (2.24)$$

Denklemini maksimum yapılacak temel harmonik  $a_1$  için çözelim.

$$a_1 = \frac{4}{\pi} \left[ -\text{Cos}\theta \Big|_0^{\alpha_1} - (-\text{Cos}\theta \Big|_{\alpha_1}^{\alpha_2}) - \text{Cos}\theta \Big|_{\alpha_2}^{\alpha_3} - (-\text{Cos}\theta \Big|_{\alpha_3}^{\alpha_4}) - \text{Cos}\theta \Big|_{\alpha_4}^{\pi/2} - \dots \right]$$

$$a_1 = \frac{4}{\pi} \left[ -\text{Cos}\alpha_1(-1) + \text{Cos}\alpha_2 - \text{Cos}\alpha_1 - \text{Cos}\alpha_3 - (-\text{Cos}\alpha_2) + \text{Cos}\alpha_4 - \text{Cos}\alpha_3 - \text{Cos}\pi/2 - (-\text{Cos}\alpha_4) - \dots \right]$$

$$a_1 = \frac{4}{\pi} \left[ 1 - 2\text{Cos}\alpha_1 + 2\text{Cos}\alpha_2 - 2\text{Cos}\alpha_3 + 2\text{Cos}\alpha_4 - \dots \right] \quad (2.25)$$

(2.1) denklemini  $a_3$  için çözelim (elimine edilecek 3. harmonik)

$$a_3 = \frac{4}{\pi} \left[ \left( \frac{-1}{3} \text{Cos}3\theta \Big|_0^{\alpha_1} \right) - \left( \frac{-1}{3} \text{Cos}3\theta \Big|_{\alpha_1}^{\alpha_2} \right) + \left( \frac{-1}{3} \text{Cos}3\theta \Big|_{\alpha_2}^{\alpha_3} \right) - \left( \frac{-1}{3} \text{Cos}3\theta \Big|_{\alpha_3}^{\alpha_4} \right) + \left( \frac{-1}{3} \text{Cos}3\theta \Big|_{\alpha_4}^{\pi/2} \right) - \dots \right]$$

$$\begin{aligned}
a_3 &= \frac{4}{\pi} \left[ \left( \frac{-1}{3} \text{Cos}3\alpha_1 - \left( \frac{-1}{3} \right) \right) - \left( \frac{-1}{3} \text{Cos}3\alpha_2 - \left( \frac{-1}{3} \text{Cos}3\alpha_1 \right) \right) + \left( \frac{-1}{3} \text{Cos}3\alpha_3 - \left( \frac{-1}{3} \text{Cos}3\alpha_2 \right) \right) \right. \\
&\quad \left. - \left( \frac{-1}{3} \text{Cos}3\alpha_4 - \left( \frac{-1}{3} \text{Cos}3\alpha_3 \right) \right) + \left( \frac{-1}{3} \text{Cos}3\pi/2 - \left( \frac{-1}{3} \text{Cos}3\alpha_4 \right) \right) + \dots \right] \\
a_3 &= \frac{4}{\pi} \left[ \frac{1}{3} - \frac{2}{3} \text{Cos}3\alpha_1 + \frac{2}{3} \text{Cos}3\alpha_2 - \frac{2}{3} \text{Cos}3\alpha_3 + \frac{2}{3} \text{Cos}3\alpha_4 - \dots \right] \\
a_3 &= \frac{4}{3\pi} [1 - 2\text{Cos}3\alpha_1 + 2\text{Cos}3\alpha_2 - 2\text{Cos}3\alpha_3 + 2\text{Cos}3\alpha_4] \quad (2.26)
\end{aligned}$$

(2.1) denklemini  $a_5$  için çözelim. (Elimine edilecek 5. harmonik)

$$\begin{aligned}
a_5 &= \frac{4}{\pi} \left[ \left( \frac{-1}{5} \text{Cos}5\theta \frac{a_1}{0} \right) - \left( \frac{-1}{5} \text{Cos}5\theta \frac{a_2}{a_1} \right) + \left( \frac{-1}{5} \text{Cos}5\theta \frac{a_3}{a_2} \right) - \left( \frac{-1}{5} \text{Cos}5\theta \frac{a_4}{a_3} \right) + \left( \frac{-1}{5} \text{Cos}5\theta \frac{\pi/2}{a_4} \right) \right] \\
a_5 &= \frac{4}{\pi} \left[ \left( \frac{-1}{5} \text{Cos}5\alpha_1 + \frac{1}{5} \right) - \left( \frac{-1}{5} \text{Cos}5\alpha_2 + \frac{1}{5} \text{Cos}5\alpha_1 \right) + \left( \frac{-1}{5} \text{Cos}5\alpha_3 + \frac{1}{5} \text{Cos}5\alpha_2 \right) \right. \\
&\quad \left. - \left( \frac{-1}{5} \text{Cos}5\alpha_4 + \frac{1}{5} \text{Cos}5\alpha_3 \right) + \left( \frac{-1}{5} \text{Cos}5\pi/2 + \frac{1}{5} \text{Cos}5\alpha_4 \right) \right] \\
a_5 &= \frac{4}{\pi} \left[ \frac{1}{5} - \frac{2}{5} \text{Cos}5\alpha_1 + \frac{2}{5} \text{Cos}5\alpha_2 - \frac{2}{5} \text{Cos}5\alpha_3 + \frac{2}{5} \text{Cos}5\alpha_4 - \dots \right] \\
a_5 &= \frac{4}{5\pi} [1 - 2\text{Cos}5\alpha_1 + 2\text{Cos}5\alpha_2 - 2\text{Cos}5\alpha_3 + 2\text{Cos}5\alpha_4 - \dots] \quad (2.27)
\end{aligned}$$

$$a_n = \frac{4}{n\pi} [1 - 2\text{Cos}n\alpha_1 + 2\text{Cos}n\alpha_2 - 2\text{Cos}n\alpha_3 + 2\text{Cos}n\alpha_4 \dots \mp 2\text{Cos}n\alpha_m] \quad (2.28)$$

DC-AC çeviricide harmonik eliminasyonunun amacı olan;

(2.28) genel denkleminde,  $a_1$  temel harmoniğini maksimum yapan ve istenmeyen diğer harmonikleri eleyen (minimize eden) anahtarlama açılarının bulunması, çok değişkenli ve çok sınırlayıcı bir optimizasyon problemidir. Bunun çözümünde bu çalışmada GA yöntemi kullanılarak, GA' ya dayalı yazılmış olan MATLAB programı kullanılacaktır.

### 3.GENETİK ALGORİTMANIN PROBLEME UYGULANMASI

Temel harmoniği maksimum, elenmesi istenen harmonikleri de minimum yapan anahtarlama açılarının hesabı optimizasyonu, aslında bir maksimizasyon problemidir. Şöyle ki genetik algoritmada tanımladığımız amaç fonksiyonu:

*Amaç fonksiyonu=1.harmoniğin mutlak değeri – elenmesi istenen diğer harmoniklerin toplamının mutlak değeri*

şeklindedir. Buradan görüldüğü gibi 1. harmoniğin en büyük, diğer harmoniklerin de en küçük olabilmesi için, GA' da belirlediğimiz amaç fonksiyonunun büyük olması gerekir. Amaç fonksiyonumuz ne kadar büyük olursa, 1. harmonik o kadar büyük, diğer harmoniklerde o kadar küçük olacaktır. Genetik algoritmada belirlediğimiz uygunluk fonksiyonu da amaç fonksiyonuna bağlıdır.

**Uygunluk fonksiyonu=amaç fonksiyonu+uygunluk fonksiyonunu negatif yapmayacak sayı**

Öyleyse genetik algoritmaya dayalı yapmış olduğumuz program uygunluk fonksiyonuna, dolayısıyla amaç fonksiyonuna göre çözüm bulacaktır. Uygunluk fonksiyonunun en büyük değeri, 1.harmoniği en büyük, diğerlerini de en küçük yapacaktır. Amacımız uygunluk fonksiyonunu maksimize etmektir.

Başlangıçta rast gele sayılarla çözüm aranmaya başlanır. Başlangıçtaki rast gele sayılar  $a_n$  genel denklemden harmonikler hesaplanır ve harmoniklere göre uygunluklar hesaplanır. Generasyon sayısı arttıkça algoritma rast gele sayılardan uygunluk fonksiyonunu artırarak en iyi çözümü bulmaya çalışır.

Genetik algoritmaya dayalı yapılan MATLAB programı şu şekilde çalışmaktadır:

**a-Elimine edilecek harmonik sayısı,**

Popülasyon sayısı,

Maksimum generasyon sayısı,

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği,

Elimine edilecek harmonik sayısından kaç fazla açı alınarak eliminasyon işleminin yapılacağı,

3. harmonikten itibaren harmonik katsayıları,

Grafik ve değer olarak kaç harmoniğin görülmek istendiği,

değerleri girilir.



**b-** Başlangıç generasyonu için popülasyon sayısı kadar ayrı ayrı, rast gele elimine edilecek aç sayısından kaç fazla aç alınacak kadar 0 ile 90 arasında sayı üretilir. Her popülasyon için ayrı olarak üretilen bu rast gele aç değerleri  $a_n$  genel denkleminde yerine konarak 3 ve 3'ün katlarının elimine edilip edilmemesine göre harmonikler hesaplanır. Bu harmonikler amaç ve uygunluk fonksiyonlarında yerine konur. Amaç ve uygunluk değerleri ve uygunluk ortalaması hesaplanır. Ölçeklenmiş uygunluklar ve bunlara göre en iyilere 2, en kötülere 0 kopya sayısı vermek üzere kopya sayıları hesaplanır. Yaşama ihtimali yüksek olanın dolaylı olarak üreme ihtimali de yüksek olur. Dolayısıyla algoritma her generasyonda hesaplanacak uygunluklara dolayısıyla kopya sayılarına göre, otomatik olarak en çok üreyen ırk üzerine gelişir de gelişir.

**c-** Hesaplanan kopya sayılarına göre en iyi açlar binary sayılara dönüştürülür. Belirli bir kurala bağlı olarak başlangıç generasyonundaki en iyi aç değerleri çaprazlanır. Programımızda yapmış olduğumuz çaprazlamada her aç değerinin tek bir biti ve bu bitin de her çaprazlamada rast gele bir hanesi çaprazlamaya uğratılmaktadır. Hangi bitin çaprazlamaya uğratılacağı her seferinde rast gele olarak belirlenmektedir.

**d-** Generasyon sayısı başta girilen maksimum generasyon sayısından küçükse, generasyon sayısı bir artırılıyor ve çaprazlama neticesinde oluşan yeni aç değerleri ikiliden tekrar ondalık sayılara dönüştürülerek elde ediliyor. Çaprazlama neticesinde oluşan bu yeni aç değerlerine göre yeni harmonikler, yeni amaç fonksiyonları ve yeni uygunluklar hesaplanıyor. Tabi ki her seferinde her popülasyonun ayrı uygunluk fonksiyonu hesaplanıyor. Ayrıca çaprazlama neticesinde hesaplanan yeni aç değerlerinin 90' ı geçme ihtimaline karşılık 90' dan büyük olanların yerine rast gele 0-90 arasında değer üretiliyor.(0-90 arasındaki sayılar ikilik sistemde 7 bit ile ifade edilmektedir.90 sayısı=1011010 Burada 0 olan bitlerin herhangi biri 1 olursa sayı 90'dan büyük olacaktır.)

**e-**Önceki hesapladığımız uygunluklar ile, çaprazlama neticesinde hesapladığımız her popülasyonun ayrı ayrı uygunluğu kıyaslanıp bunların içerisinden uygunluğu en büyük olan popülasyonların açları seçilir. Bu seçilen açların popülasyonlarının tekrar uygunlukları hesaplanır ve bu uygunluklara göre de uygunluk ortalaması hesaplanır.

**f-**Bu uygunluk ortalaması, önceki uygunluk ortalamasından küçük veya eşitse(yani uygunluk artırmada çaprazlama neticesinde bir başarı sağlanamamışsa) mutasyon işlemine geçilir. En son elde edilen açlar tekrar binary sayı sistemine çevriliyor. En son

elde edilen açıların rast gele seçilen bir biti mutasyona uğratarak, yeni açılar elde ediliyor.

g- Mutasyon sonucunda oluşan yeni açılara göre yeni uygunluklar hesaplanıyor. Bu yeni uygunluklar eski uygunluklardan büyükse her büyük olan uygunluğun kendi popülasyonuna açıları seçiliyor.

h-Generasyon sayısı maksimum generasyon sayısından büyükse birinci popülasyondan en son popülasyona kadar tüm generasyonlardaki uygunluğu en büyük olan açılar seçiliyor. Bu en iyi açılara göre kaç harmonik görülmek isteniyorsa , tüm harmonikler bu açılara göre hesaplanıyor ve bu harmoniklerin spektrumu çiziliyor.

### 3.1. Programın Çalışmasına Örnekler

Örnek 1.

Elimine edilecek harmonik sayısı:4

Popülasyon sayısı:7

Maksimum generasyon sayısı:7

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilsin

Elimine edilecek harmonik sayısından kaç fazla açı alınarak eliminasyon yapılacağı:1

Toplam görülmek istenen harmonik sayısı:10

Harmonik ağırlıkları 1 alınmıştır.

Değerleri EK-1 de sunulan GAHE programında girilerek çalıştırıldığında;

Tablo 3.1 Örnek 1 Başlangıç Generasyonu

Pop.	Anahtarlama Açıları					Uygunluk	Ölç.Uyg.	Kop.
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı			
1.Pop.	18.2383	37.6785	47.2637	60.4924	76.1599	19.1780	36.9733	2
2.Pop.	1.7676	34.1533	61.3149	74.8616	75.4307	18.0623	7.1404	0
3.Pop.	17.0688	27.4156	38.6003	45.2532	63.8524	18.2942	13.3393	1
4.Pop.	13.5786	17.4088	27.2488	48.7506	61.4001	18.8576	28.4052	2
5.Pop.	34.0536	53.4207	62.8109	76.8290	77.4010	18.4689	18.0126	1
6.Pop.	44.6897	58.0419	73.6177	73.9466	80.9792	18.0406	6.5581	0
7.Pop.	26.0753	30.7074	30.7774	48.0671	59.4205	18.5050	18.9777	1

Görüldüğü gibi uygunluğu ve dolayısıyla ölçeklenmiş uygunlukları yüksek olan popülasyonların kopya sayıları da yüksek olmuştur. Kopya sayıları 0 olanlar çaprazlamaya uğratılmaz. Kopya sayıları 2 olan popülasyonlarda her açı 2 defa kullanılıp 2 defa çaprazlanır. Popülasyon sayısı her zaman aynı olacaktır. Yani kopya sayılarının toplamı popülasyon sayısına her zaman eşittir.

Tablo 3.2 Örnek 1 Çaprazlama Sonucundaki Durum

Pop.	Anahtarlama Açıları					Uygunluk
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı	
1.Pop.	6.5581	18.9777	21.0000	27.4063	38.0938	19.7811
2.Pop.	6.5581	18.9777	25.3750	27.7188	29.8125	20.4977
3.Pop.	6.5581	18.9777	38.0313	52.4063	61.4688	19.2420
4.Pop.	6.5581	18.9777	22.2188	37.5313	47.2500	19.8469
5.Pop.	6.5581	18.9777	26.0938	30.5625	30.7500	20.5257
6.Pop.	6.5581	18.9777	29.1875	36.5313	49.7500	19.3617
7.Pop.	6.5581	11.9688	13.7500	17.3750	18.9777	20.5234

Kopya sayıları 0 olanların kullanılmadığı ve kopya sayıları 2 olanların 2 defa kullanıldığı çaprazlama işleminden sonra yeni açı değerleri elde edilir. Burada yapılan çaprazlamada tek noktalı çaprazlama kullanılmıştır. Yani açıların tam kısmının ve ondalık kısmının sadece 1 biti çaprazlanmıştır. Tam kısımlar kendi arasında, ondalık kısımlar da kendi arasında çaprazlanmıştır. Çaprazlama sırasında hangi bitin çaprazlanacağı her zaman rast gele belirlenmiştir.

Tablo 3.3 Örnek 1 Çaprazlama Sonucunda Üretilen Değerler İle Başlangıç Generasyonundaki Değerlerin Kıyaslanıp En Uygunlarının Seçilmesi (2.Generasyon)

Pop.	Anahtarlama Açıları					Uygunluk	Ölç.uyg.	Kop.
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı			
1.Pop.	6.5581	18.9777	26.0938	30.5625	30.7500	20.5257	39.9367	2
2.Pop.	6.5581	11.9688	13.7500	17.3750	18.9777	20.5234	39.8525	2
3.Pop.	6.5581	18.9777	25.3750	27.7188	29.8125	20.4977	38.9307	2
4.Pop.	6.5581	18.9777	22.2188	37.5313	47.2500	19.8469	15.6169	1
5.Pop.	6.5581	18.9777	21.0000	27.4063	38.0938	19.7811	13.2605	0
6.Pop.	6.5581	18.9777	29.1875	36.5313	49.7500	19.3617	0.0000	0
7.Pop.	6.5581	18.9777	38.0313	52.4063	61.4688	19.2420	0.0000	0

Burada Tablo 3.2. deki çaprazlama sonrasındaki açılar ile Tablo 3.1. deki başlangıç generasyonundaki açılarının uygunlukları kıyaslanmış ve bu her 2 durumdaki en iyi popülasyonlar, dolayısıyla en iyi açılar seçilmiştir. Burada çaprazlama sonrasındaki tüm popülasyonların uygunluklarının başlangıçtaki popülasyonlara göre daha iyi olduğu ve dolayısıyla seçildiği görülmektedir. Bu şekilde yapılarak çaprazlama sonrasında çözümden çok uzak yerlere gidilmesi de engellenmiştir. Programımız her zaman uygunlukları baz alarak arama yapmaktadır. Gerçek çözümden çok çok uzak yerlere gider ve oralarda da her zaman arama yapar. Ancak eğer çözüm o uzak yerlerde değilse, bu şekilde yapılarak oralarda kalmamasını sağlar. Gerçek çözüme yaklaşmayı sağlar. Burada dikkat edilecek başka bir husus ta, uygunluk ortalamalarıyla ilgilidir.

Başlangıç generasyonu için uygunluk ortalaması=18.4867

Programda hem her generasyonun başlangıcında ve hem de çaprazlama sonrasında yeni üretilen değerler ile öncekilerin kıyaslanıp en uygunlarının seçilmesinden sonra uygunluk ortalamaları hesaplanmaktadır. Bu durumda hesaplanan uygunluk ortalaması=19.9684

Görüldüğü gibi uygunluk ortalaması artmıştır. Programımızda uygunluk ortalaması arttığı için bu durumda mutasyona geçilmeyecektir. Her seferinde uygunluk ortalamalarını kıyaslayıp eğer uygunluk ortalaması önceki generasyondan küçükse mutasyona geçilecektir. Burada mutasyon işlemine gerek kalmadığından buradaki açılar aynı zamanda 2. generasyonun başlangıç açılarıdır ve yine uygunlukları, ölçeklenmiş uygunlukları, kopya sayıları hesaplanmıştır.

Tablo 3.4 Örnek 1 İkinci Generasyonda Çaprazlama Sonucu Üretilen Değerler

Popülasyon	Anahtarlama Açıları					Uygunluk
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı	
1.Pop.	14.5938	22.8438	26.2188	33.6563	39.4063	20.1778
2.Pop.	6.5938	18.7188	22.0938	30.8438	30.9063	20.6606
3.Pop.	6.7188	11.4688	25.4688	27.9688	29.7813	19.7353
4.Pop.	6.7813	13.9688	16.7188	18.5938	19.9688	20.5288
5.Pop.	6.8438	18.9688	26.2188	27.5938	29.5313	20.4354
6.Pop.	6.7813	18.9688	25.4688	30.8438	30.9063	20.5540
7.Pop.	6.5938	9.9688	13.9688	18.7188	19.4688	20.4121

2. generasyonda yine 1. generasyonda olduğu gibi kopya sayılarına göre çaprazlama yapılır. Yeni aç değerleri elde edilir.

**Tablo 3.5 Örnek 1 İkinci Generasyonda Çaprazlama Sonucunda Üretilen Değerler İle Başlangıç Generasyonundaki Değerlerin Kıyaslanıp En Uygunlarının Seçilmesi (3.Generasyon)**

Pop.	Anahtarlama Açları					Uygunluk	Ölç.uyg.	Kop.
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı			
1.Pop.	6.5938	18.7188	22.0938	30.8438	30.9063	20.6606	41.0645	2
2.Pop.	6.7813	18.9688	25.4688	30.8438	30.9063	20.5540	24.0161	1
3.Pop.	6.7813	13.9688	16.7188	18.5938	19.9688	20.5288	19.9879	1
4.Pop.	6.5581	18.9777	26.0938	30.5625	30.7500	20.5257	19.4956	1
5.Pop.	6.5581	11.9688	13.7500	17.3750	18.9777	20.5234	19.1194	1
6.Pop.	6.5581	18.9777	25.3750	27.7188	29.8125	20.4977	15.0031	1
7.Pop.	6.8438	18.9688	26.2188	27.5938	29.5313	20.4354	5.0391	0

2. generasyonda uygunluk ortalaması=19.9684

Çaprazlama sonrasında seçilen en uygunların uygunluk ortalaması=20.5322

Görüldüğü gibi uygunluk ortalaması önceki generasyonun uygunluk ortalamasından büyüktür ve dolayısıyla yine mutasyona gerek kalmamıştır. Generasyonda uygunluk ortalamasının yükselmesi aranan çözümü bulma yolunda herhangi bir sıkışma olmadığını gösterir. Yani GA çok geniş bir çözüm havzasını taradığı için belli bir noktaya bağlı kalmamıştır. Çok geniş alanları tarayarak çözümü bulacaktır. Bu şekilde tüm generasyonlarda en uygun çözümler alınarak çözüme daha da yaklaşıp yaklaşıp en sonunda başta girmiş olduğumuz, maksimum generasyon sayısından sonra genetik algoritma artık o ana kadarki tüm çözümlerin içerisinde en iyisini çözüm olarak alacaktır. 7. generasyondan sonra çözüm aşağıda gösterilmiştir. Uygunluğu en yüksek olan popülasyon en uygun çözümdür.

Tablo 3.6 Örnek 1 7. generasyon (Çözüm)

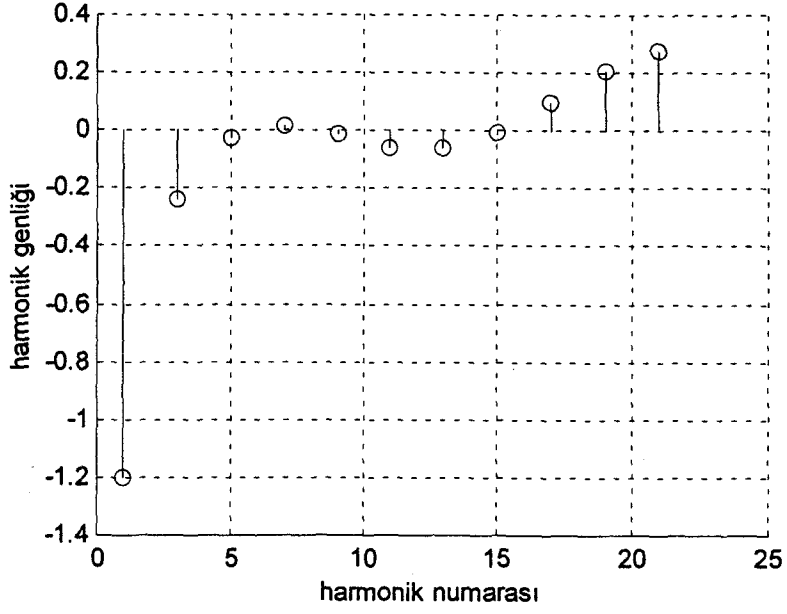
Popülasyon	Anahtarlama Açıları					Çözüm
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı	
1.Pop.						
2.Pop.	5.0391	14.5313	15.0031	18.5938	22.2188	
3.Pop.	4.7188	7.8339	8.4050	18.5938	22.0938	
4.Pop.	6.9063	18.7188	18.8438	22.0938	25.4063	
5.Pop.	6.7813	7.8339	8.4050	22.9688	25.4688	
6.Pop.	6.7813	18.9688	25.4688	30.8438	30.9063	
7.Pop.	6.7813	13.9688	16.7188	18.5938	19.9688	

Aslında burada diğer popülasyonlar da optimizasyon işlemini en uygun popülasyona çok yakın olarak yapmıştır. Yani diğer popülasyonlar da 1. harmoniği en büyük diğerlerini de en küçük yapmıştır. Ama uygunluğu en yüksek olan popülasyon programımızda en iyi çözüm olarak alınmıştır.

En iyi çözümün harmonikleri şu şekildedir:

Tablo 3.7 Örnek 1 Çözüm Harmonikleri

<b>1.har.</b>	<b>3.har.</b>	<b>5.har.</b>	<b>7.har.</b>	<b>9.har.</b>	<b>11.har.</b>
-1.2027	-0.2403	-0.0253	0.0182	-0.0150	-0.0580
<b>13.har.</b>	<b>15.har.</b>	<b>17.har.</b>	<b>19.har.</b>	<b>21.har.</b>	
-0.0607	-0.0034	0.0994	0.2086	0.2796	



Şekil 3.1.Örnek 1 Harmonik Spektrumu

Örnek 2.

Elimine edilecek harmonik sayısı:5

Popülasyon sayısı:15

Maksimum generasyon sayısı:15

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilsin

Elimine edilecek harmonik sayısından kaç fazla açı alınarak eliminasyon yapılacağı:1

Toplam görülmek istenen harmonik sayısı:10

Harmonik ağırlıkları 1 alınmıştır.

Değerleri EK-1 de verilen GAHE programında girilerek çalıştırıldığında;



Tablo 3.8 Örnek 2 Başlangıç Generasyonu

Pop.	Anahtarlama Açıları						Uyg.	Ölç.Uy.	Kop.
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı	6.Açı			
1.Pop.	20.8035	43.7384	54.6158	68.5887	80.2169	85.5116	22.7259	10.4195	0
2.Pop.	1.6653	40.0233	41.0821	55.3889	71.2743	73.9266	22.9183	15.5043	1
3.Pop.	15.8640	36.5136	66.4387	82.5214	82.9632	84.1923	23.0558	19.1375	1
4.Pop.	0.8875	5.2102	31.7581	36.9243	73.1850	80.4285	24.0882	46.4198	2
5.Pop.	12.5002	17.8850	17.8933	18.2489	24.4969	54.3413	23.2501	24.2719	1
6.Pop.	1.3747	37.6785	40.0587	41.9395	67.2107	83.8633	22.2796	0.0000	0
7.Pop.	1.7676	18.2383	47.2637	60.4924	75.4307	76.1599	23.6222	34.1044	2
8.Pop.	34.1533	38.6003	45.2532	61.3149	63.8524	74.8616	23.1834	22.5111	1
9.Pop.	17.0688	17.4088	27.2488	27.4156	48.7506	61.4001	23.7132	36.5096	2
10.Pop.	13.5786	34.0536	53.4207	62.8109	76.8290	77.4010	23.2346	23.8634	1
11.Pop.	44.6897	58.0419	59.4205	73.6177	73.9466	80.9792	22.6629	8.7543	0
12.Pop.	26.0753	27.8361	30.7074	30.7774	48.0671	65.4402	23.3031	25.6733	1
13.Pop.	33.3372	40.0392	49.1914	51.1265	63.2466	75.4646	23.3105	25.8671	1
14.Pop.	47.0331	55.9179	62.5111	71.5339	79.2128	86.1159	23.5354	31.8119	1
15.Pop.	15.5661	22.7096	24.4303	66.3575	78.8168	88.1772	23.2653	24.6728	1

15 popülasyonlu ve her popülasyonda 6 açı değeri rast gele üretilerek başlangıç generasyonu başlatılır. Uygunluk ve ölçeklenmiş uygunluklara göre kopya sayıları hesaplanır. Uygunluğu en kötü olan popülasyonlar üremeyecektir.



Tablo 3.9 Örnek 2 Çaprazlama Sonucundaki Durum

Pop.	Anahtarlama Açıları						Uyg.
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı	6.Açı	
1.Pop.	8.7543	22.5111	23.8634	24.6728	25.1563	25.6733	24.7715
2.Pop.	1.3750	8.7543	17.0000	22.5111	23.3750	23.8634	25.3245
3.Pop.	2.4375	8.7543	10.8942	15.5938	22.5111	23.8634	24.9953
4.Pop.	1.8750	8.7543	18.4688	22.5111	23.8634	24.6728	25.4474
5.Pop.	1.6250	4.5000	8.7543	17.8750	22.5111	23.8634	24.9909
6.Pop.	2.3750	8.7543	13.1875	16.4688	18.0938	22.5111	25.1175
7.Pop.	8.7543	13.4375	22.5111	23.8634	24.6728	25.6733	25.5816
8.Pop.	8.7543	11.1563	22.5111	22.8125	23.8634	24.2813	25.2804
9.Pop.	5.2500	8.7543	22.5111	23.8634	24.6728	25.6733	25.3854
10.Pop.	8.7543	13.5625	22.5111	23.8634	24.6728	25.6733	25.5755
11.Pop.	8.7543	10.4375	22.5111	23.8634	24.5792	24.6728	25.1637
12.Pop.	8.7543	22.5111	23.8634	24.6728	25.6733	25.8671	24.8079
13.Pop.	1.3750	5.0625	8.7543	22.5111	23.8634	24.6728	24.6906
14.Pop.	8.7543	17.0000	17.3750	19.4688	22.5111	23.8634	25.1311
15.Pop.	1.8750	8.7543	19.4688	22.5111	23.8634	24.6728	25.5299

Başlangıçta üretilen rast gele açılar, iyiliklerine göre birbirleriyle çaprazlanarak çoğalır. Çaprazlama ile iyi açılardan daha iyi açılar üretilebilir. Bu şekilde çok değişik açılar test edilebilir. Çaprazlama neticesinde yeni üretilen açı, çaprazlama yapılan her iki açının da özelliklerini alabilir. İkisinin kopyası olur. Yeni bir değer üretilir.

**Tablo 3.10 Örnek 2 Çaprazlama Sonucunda Üretilen Değerler İle Başlangıç Generasyonundaki Değerlerin Kıyaslanıp En Uygunlarının Seçilmesi (2.Generasyon)**

Pop.	Anahtarlama Açılıarı						Uyg.	Ölç.Uy.	Kop.
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı	6.Açı			
1.Pop.	8.7543	13.4375	22.5111	23.8634	24.6728	25.6733	25.5816	50.3724	2
2.Pop.	8.7543	13.5625	22.5111	23.8634	24.6728	25.6733	25.5755	49.9838	2
3.Pop.	1.8750	8.7543	19.4688	22.5111	23.8634	24.6728	25.5299	47.0799	2
4.Pop.	1.8750	8.7543	18.4688	22.5111	23.8634	24.6728	25.4474	41.8232	2
5.Pop.	5.2500	8.7543	22.5111	23.8634	24.6728	25.6733	25.3854	37.8735	2
6.Pop.	1.3750	8.7543	17.0000	22.5111	23.3750	23.8634	25.3245	33.9910	1
7.Pop.	8.7543	11.1563	22.5111	22.8125	23.8634	24.2813	25.2804	31.1843	1
8.Pop.	8.7543	10.4375	22.5111	23.8634	24.5792	24.6728	25.1637	23.7508	1
9.Pop.	8.7543	17.0000	17.3750	19.4688	22.5111	23.8634	25.1311	21.6771	1
10.Pop.	2.3750	8.7543	13.1875	16.4688	18.0938	22.5111	25.1175	20.8115	1
11.Pop.	2.4375	8.7543	10.8942	15.5938	22.5111	23.8634	24.9953	13.0228	0
12.Pop.	1.6250	4.5000	8.7543	17.8750	22.5111	23.8634	24.9909	12.7460	0
13.Pop.	8.7543	22.5111	23.8634	24.6728	25.6733	25.8671	24.8079	1.0864	0
14.Pop.	8.7543	22.5111	23.8634	24.6728	25.1563	25.6733	24.7715	0.0000	0
15.Pop.	1.3750	5.0625	8.7543	22.5111	23.8634	24.6728	24.6906	0.0000	0

Burada Tablo 3.9' daki çaprazlama sonrasındaki açılar ile Tablo 3.8. deki başlangıç generasyonundaki açıların uygunları kıyaslanmış ve bu her 2 durumdaki en iyi popülasyonlar, dolayısıyla en iyi açılar seçilmiştir. Buradaki seçilme kesinlikle rast gele yapılmamaktadır. Seçim işlemi uygunluk fonksiyonunu temel almaktadır. Uygunluk fonksiyonu yüksek olan popülasyonlar, bireyler seçilir. Programımızda sadece rast gelelik başlangıçta üretilen değerlerdir. Onun dışında üretilen ve seçilen hiçbir değer rast gele değildir. Kaliteli olanlar yaşamını sürdürmeye devam edecektir. Seçim işlemi programın çözümü bulması yolunda zamandan tasarruf sağlamak ve en optimum sonucu bulmak için çok önemlidir. Çok geniş bir çözüm havzasında aradığı çözüme, seçme işlemleriyle yaklaşmaktadır.

Yine burada da uygunluk ortalamalarına dikkat edecek olursak;

Başlangıç generasyonu için uygunluk ortalaması=23.2099

2. generasyon için hesaplanan uygunluk ortalaması=25.1862

Görüldüğü gibi uygunluk ortalaması artmıştır. Uygunluk ortalaması arttığı için bu durumda mutasyona geçilmeyecektir. Burada mutasyon işlemine gerek kalmadığından buradaki açılar aynı zamanda 2. generasyonun başlangıç açılarıdır ve yine uygunlukları, ölçeklenmiş uygunlukları, kopya sayıları hesaplanmıştır.

Tablo 3.11 Örnek 2 İkinci Generasyonda Çaprazlama Sonucu Üretilen Değerler

Popülasyon	Anahtarlama Açıları						Uyg.
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı	6.Açı	
1.Pop.	6.5000	15.4688	16.0938	16.3438	22.6250	24.7500	25.0947
2.Pop.	2.4688	8.8750	15.2188	21.8438	24.5313	25.6563	25.2077
3.Pop.	8.7500	13.5938	19.4375	22.5000	22.5313	23.8438	25.3851
4.Pop.	8.7500	17.7188	19.3438	21.9688	24.9063	25.6563	25.1030
5.Pop.	1.9063	8.8125	19.4688	23.8438	24.5000	24.7188	25.4827
6.Pop.	7.9688	8.9063	22.5000	22.5000	24.5000	26.3438	24.9801
7.Pop.	7.7813	8.3438	17.9063	18.9688	22.3438	24.8125	25.0651
8.Pop.	8.7500	11.0313	22.5000	22.5000	23.9688	24.6563	25.2504
9.Pop.	7.4688	9.7813	21.8438	22.4375	22.5000	22.5000	25.2598
10.Pop.	1.4063	8.8750	17.9688	23.8438	24.5313	25.7188	25.3240
11.Pop.	8.5000	12.4688	22.5000	23.9688	25.5313	25.9688	25.5474
12.Pop.	5.4688	8.5313	22.5625	23.7813	24.7813	25.6563	25.3196
13.Pop.	5.8438	8.7813	22.5000	22.5313	24.6250	31.7188	25.0268
14.Pop.	8.7500	9.6563	17.9063	19.3438	22.3438	24.9688	25.1587
15.Pop.	1.9063	8.9375	19.4063	22.5625	23.8438	24.9063	25.5036

3. generasyonda yine 1. generasyonda olduğu gibi kopya sayılarına göre çaprazlama yapılır. Yeni açı değerleri elde edilir.

**Tablo 3.12 Örnek 2 İkinci Generasyonda Çaprazlama Sonucunda Üretilen Değerler İle Başlangıç Generasyonundaki Değerlerin Kıyaslanıp En Uygunlarının Seçilmesi (3.Generasyon)**

Pop.	Anahtarlama Açılıarı						Uyg.	Ölç.Uy.	Kop.
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı	6.Açı			
1.Pop.	8.7543	13.4375	22.5111	23.8634	24.6728	25.6733	25.5816	50.8263	2
2.Pop.	8.7543	13.5625	22.5111	23.8634	24.6728	25.6733	25.5755	49.9059	2
3.Pop.	8.5000	12.4688	22.5000	23.9688	25.5313	25.9688	25.5474	45.6584	2
4.Pop.	1.8750	8.7543	19.4688	22.5111	23.8634	24.6728	25.5299	43.0289	2
5.Pop.	1.9063	8.9375	19.4063	22.5625	23.8438	24.9063	25.5036	36.0610	2
6.Pop.	1.9063	8.8125	19.4688	23.8438	24.5000	24.7188	25.4827	35.9043	2
7.Pop.	1.8750	8.7543	18.4688	22.5111	23.8634	24.6728	25.4474	30.5802	1
8.Pop.	5.2500	8.7543	22.5111	23.8634	24.6728	25.6733	25.3854	21.2265	1
9.Pop.	8.7500	13.5938	19.4375	22.5000	22.5313	23.8438	25.3851	21.1833	1
10.Pop.	1.3750	8.7543	17.0000	22.5111	23.3750	23.8634	25.3245	12.0320	0
11.Pop.	1.4063	8.8750	17.9688	23.8438	24.5313	25.7188	25.3240	11.9668	0
12.Pop.	5.4688	8.5313	22.5625	23.7813	24.7813	25.6563	25.3196	11.2935	0
13.Pop.	8.7543	11.1563	22.5111	22.8125	23.8634	24.2813	25.2804	5.3852	0
14.Pop.	7.4688	9.7813	21.8438	22.4375	22.5000	22.5000	25.2598	2.2767	0
15.Pop.	8.7500	11.0313	22.5000	22.5000	23.9688	24.6563	25.2504	0.8683	0

2. generasyonda uygunluk ortalaması=25.1862

Çaprazlama sonrasında seçilen en uygunların uygunluk ortalaması=25.4301

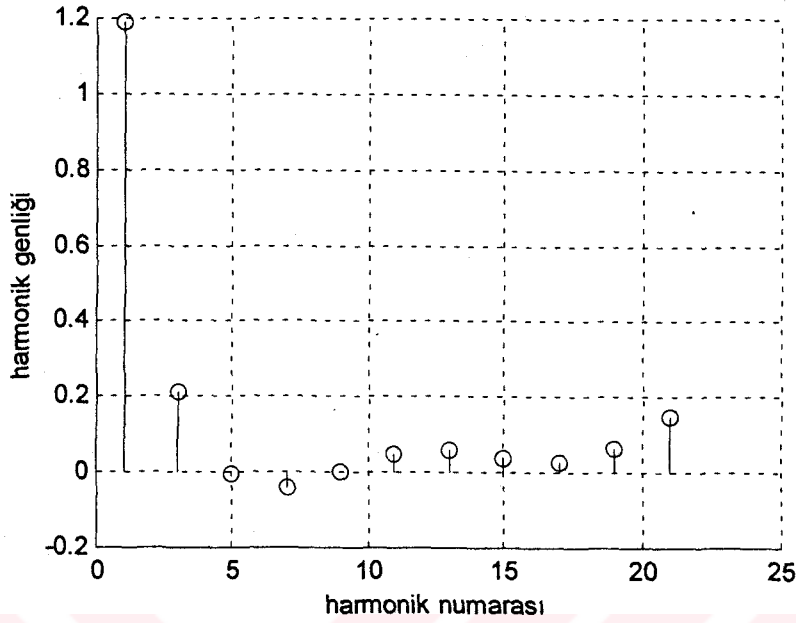
Görüldüğü gibi uygunluk ortalaması önceki generasyonun uygunluk ortalamasından büyüktür ve dolayısıyla yine mutasyona gerek kalmamıştır. Maksimum generasyon sayısına ulaşıldığında, uygunluğu en yüksek olan popülasyon en uygun çözümdür.

Tablo 3.13 Örnek 2 15. Generasyon (Çözüm)

Popülasyon	Anahtarlama Açıları						Çözüm
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı	6.Açı	
1.Pop.	8.7543	13.5625	22.5111	23.8634	24.6728	25.6733	*****
2.Pop.	8.7543	13.5625	22.5111	23.8634	24.6728	25.6733	
3.Pop.	8.5000	12.4688	22.5000	23.9688	25.5313	25.9688	
4.Pop.	7.3438	9.6563	14.6563	16.9063	22.8750	23.5000	
5.Pop.	1.8750	8.7543	19.4688	22.5111	23.8634	24.6728	
6.Pop.	1.9063	8.9375	19.4063	22.5625	23.8438	24.9063	
7.Pop.	1.9063	8.8125	19.4688	23.8438	24.5000	24.7188	
8.Pop.	1.9063	8.7813	19.4688	23.9688	25.5938	25.9688	
9.Pop.	8.9688	12.4688	22.9688	23.8438	24.7188	24.9688	
10.Pop.	1.8750	8.7543	18.4688	22.5111	23.8634	24.6728	
11.Pop.	6.9010	6.9347	7.0253	7.2802	9.8448	13.2267	
12.Pop.	8.4560	8.4688	8.9744	11.7640	11.8212	12.5995	
13.Pop.	8.4222	8.4499	8.4824	8.5698	8.5156	12.5938	
14.Pop.	8.4560	8.8438	8.9744	11.7640	11.8212	12.5995	
15.Pop.	8.2188	8.4375	8.5313	8.8125	8.8125	12.4688	

Tablo 3.14 Örnek 2 Çözüm Harmonikleri

1.har.	3.har.	5.har.	7.har.	9.har.	11.har.
1.1906	0.2111	-0.0057	-0.0399	0.0020	0.0472
13.har.	15.har.	17.har.	19.har.	21.har.	
0.0571	0.0383	0.0291	0.0655	0.1510	



Şekil 3.2 Örnek 2 Harmonik Spektrumu

Programda ayrıca bastırılacak harmoniklere ağırlıklar vermekte mümkündür. Burada mantık; üretilen dalga şeklinin süreceği sistemde, harmonik akımlarının sistemin yapısından kaynaklanan nedenlerle kendiliğinden küçük olabilmesi gerçeğidir. Örneğin belli harmonikleri bastırılmış bir dalga şekli, basit olması için bir tek fazlı asenkron motoru sürüyor olsun. Asenkron motor en basit haliyle bir omik-endüktif statik yük olarak ele alınabilir. Omik-endüktif bu yükten zaten yüksek dereceli harmonikler bu frekansta yük büyük empedans göstereceğinden büyük akımlar akıtamayacaktır. Bu nedenle bastırılacak harmonikler içinde yüksek dereceli olanların çok küçük genlikli olması gerekmez. Bu düşünceyle hazırladığımız programda kullanıcıya harmonik eliminasyonlu işareti kullanacağı yere göre istediği bastırılacak harmoniği bastırma oranını bu katsayı(ağırlıkla) seçebilme imkanı tanımıştır.

**Örnek 3.**

Aynı örnekler harmoniklere ağırlıklar vererek yapılmış ve şu sonuçlar elde edilmiştir.(Harmonik ağırlıklarının etkisini göstermek amacıyla Örnek 1. çalıştırılırsa)

Elimine edilecek harmonik sayısı:4

Popülasyon sayısı:7

Maksimum generasyon sayısı:7

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilsin

Elimine edilecek harmonik sayısından kaç fazla açı alınarak eliminasyon yapılacağı:1

Toplam görülmek istenen harmonik sayısı:10

3.harmonikten itibaren sırasıyla harmonik ağırlıklarını giriniz:

1.0000 0.4000 0.3000 0.0800

Değerleri girilerek GAHE programı çalıştırıldığında;

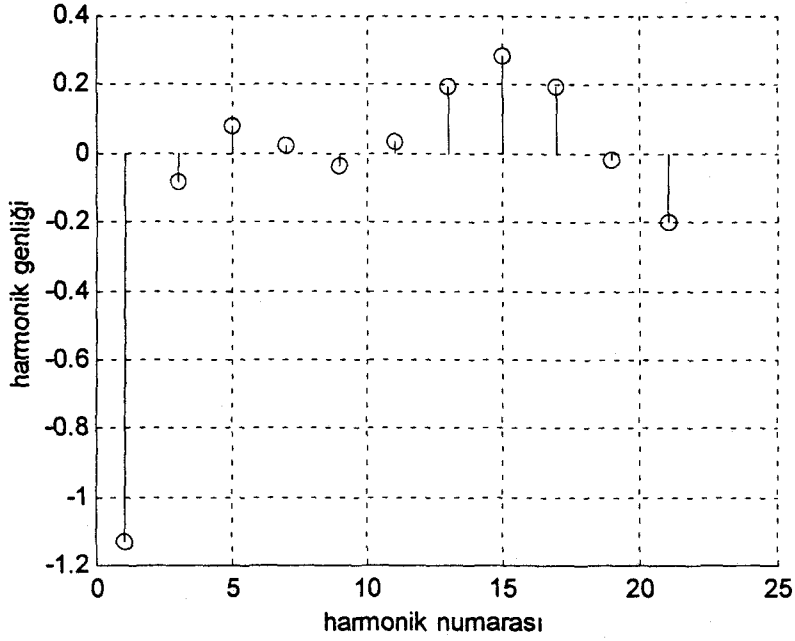
Çözüm açıları:

4.9688 8.2188 13.1250 26.5313 30.9688

**Tablo 3.15 Örnek 3 Çözüm Harmonikleri**

<b>1.har.</b>	<b>3.har.</b>	<b>5.har.</b>	<b>7.har.</b>	<b>9.har.</b>	<b>11.har.</b>
-1.1285	-0.0842	0.0820	0.0197	-0.0376	0.0361
<b>13.har.</b>	<b>15.har.</b>	<b>17.har.</b>	<b>19.har.</b>	<b>21.har.</b>	
0.1953	0.2853	0.1985	-0.0177	-0.1995	





Şekil 3.3 Örnek 3 Harmonik Spektrumu

Görüldüğü gibi, 3. harmoniğin değeri 0.2403' den 0.0842' ye düşürülmüştür. Bastırılacak harmoniklere uygun ağırlıklar vermek suretiyle 3. harmoniğin değeri de veya diğer harmonikler de istenildiği kadar küçültülebilir.

Örnek 4.

Elimine edilecek harmonik sayısı:5

Popülasyon sayısı:15

Maksimum generasyon sayısı:16

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilsin

Elimine edilecek harmonik sayısından kaç fazla açılı alınarak eliminasyon yapılacağı:1

Toplam görülmek istenen harmonik sayısı:8

3.harmonikten itibaren sırasıyla harmonik ağırlıklarını giriniz:

1.0000 0.4000 0.3000 0.1000 0.0400

Değerleri girilerek GAHE programı çalıştırıldığında;

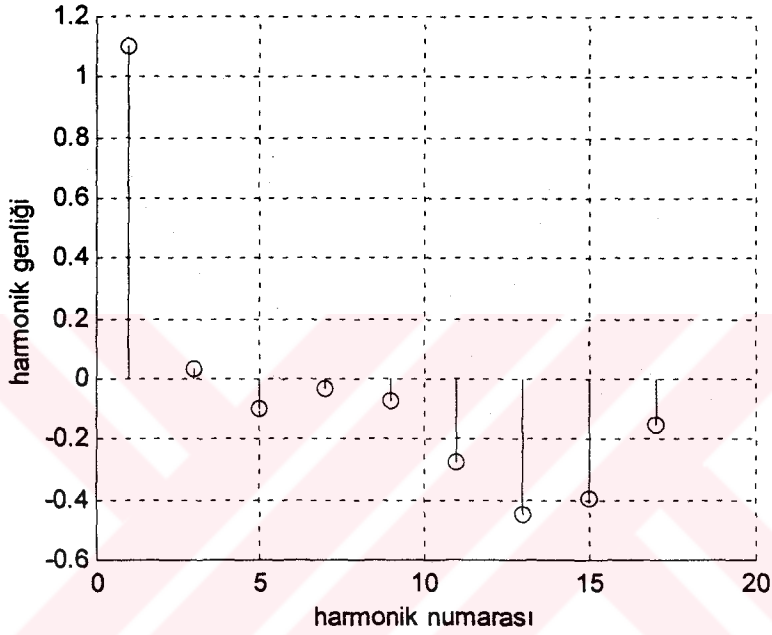
Çözüm açıları:

0.4063 12.5938 16.5313 16.7813 30.0938 34.7188



Tablo 3.16 Örnek 4 Çözüm Harmonikleri

<b>1.har.</b>	<b>3.har.</b>	<b>5.har.</b>	<b>7.har.</b>	<b>9.har.</b>	<b>11.har.</b>
1.0987	0.0347	-0.0969	-0.0317	-0.0717	-0.2756
<b>13.har.</b>	<b>15.har.</b>	<b>17.har.</b>			
-0.4506	-0.3974	-0.1511			



Şekil 3.4 Örnek 4 Harmonik Spektrumu

3. harmoniğin değeri 0.2111' den 0.0347' ye düşürülmüştür.

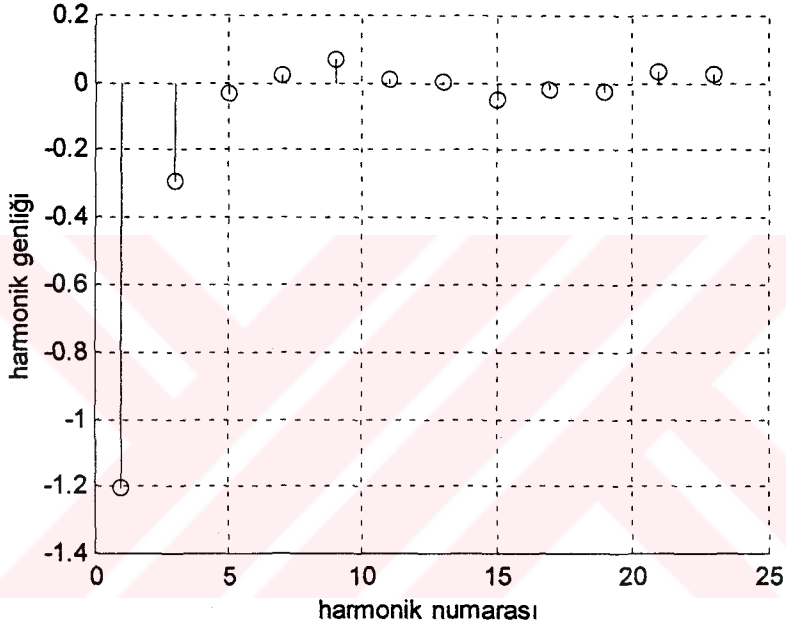
### 3.2 Başka Çözüm Yöntemleriyle Kıyaslama

Çeyrek ve yarı dalga simetrisine sahip işaretin açılı yerleri 'Tatar' tarafından aşağıdaki gibi bulunmuştur. (Lagrange yöntemine göre)

3.5140 7.9990 10.7820 16.8760 18.9840 89.6700

Burada amaç bu 6 açılı ile 5., 7., 11., 13. harmonikleri bastırmaktır. 3 ve 3'ün katları 3 fazlı 3 telli sistemlerde harmonik akım akmadığı düşünülmüştür. Kutup gerilimleri arasında  $120^\circ$  faz farkı olduğunda 3'ün çarpanları olan harmoniklerde kutup gerilimi

360° farklılık gösterir. Kutup gerilimlerinin farkı alındığında, bu harmonikler fazlar arası gerilimde gözükmez. Yüksek dereceli harmonikler filtre ile rahatlıkla giderilebilir. İdeal sonuç, düşük dereceli harmoniklerin tamamen süzülmesi ve bağlı darbe genişliği ile doğrudan orantılı bir temel bileşenin kalmasıdır. Yüksek dereceli harmoniklerin filtreleme elemanı ile bastırılabilmesi daha kolaydır. Yukarıdaki açılara göre şu sonuçlar bulunmuştur.



Şekil 3.5 Klasik Optimizasyon Yöntemiyle 5., 7., 11., 13. Harmoniklerin Elenmesi

Tablo 3.17 Klasik Optimizasyon Yöntemi Çözüm Harmonikleri

<b>1.har.</b>	<b>3.har.</b>	<b>5.har.</b>	<b>7.har.</b>	<b>9.har.</b>	<b>11.har.</b>
-1.2048	-0.2909	-0.0325	0.0247	0.0671	0.0117
<b>13.har.</b>	<b>15.har.</b>	<b>17.har.</b>	<b>19.har.</b>	<b>21.har.</b>	<b>23.har.</b>
0.0046	-0.0487	-0.0198	-0.0276	0.0324	0.0261

$$\text{Toplam Harmonik Distorsiyonu (THD)} = \frac{\sqrt{H_5^2 + H_7^2 + H_{11}^2 + H_{13}^2}}{H_1}$$

$$\text{THD} = \frac{\sqrt{0.0325^2 + 0.0247^2 + 0.0117^2 + 0.0046^2}}{1.2048} = 0.0354$$

Genetik Algoritmalar ile Çözüm:

Elimine edilecek harmonik sayısı:4

Popülasyon sayısı:15

Maksimum generasyon sayısı:16

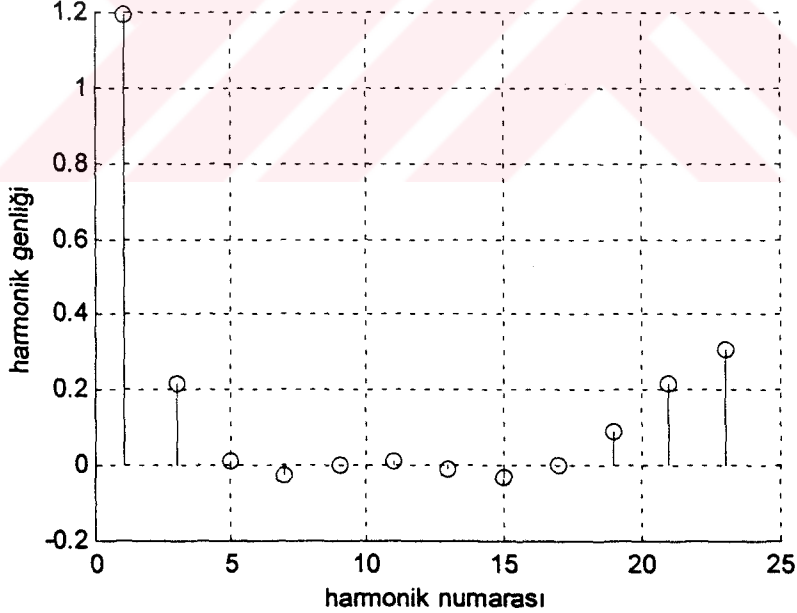
3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilmesin

Elimine edilecek harmonik sayısından kaç fazla açı alınarak eliminasyon yapılacağı:2

Toplam görülmek istenen harmonik sayısı:12

Harmonik ağırlıkları:1

Değerleri girildikten sonra, Bizim programımızda da aynı amaçla çözüm yapılmış ve şu sonuçlar bulunmuştur.



Şekil 3.6 Genetik Algoritmalar Yöntemiyle 5., 7., 11., 13. Harmoniklerin Elenmesi

Çözüm açıları:

8.2813 13.2813 21.2813 21.2813 25.2813 27.2813

Tablo 3.18 Genetik Algoritmalar Yöntemi Çözüm Harmonikleri

<b>1.har.</b>	<b>3.har.</b>	<b>5.har.</b>	<b>7.har.</b>	<b>9.har.</b>	<b>11.har.</b>
1.1923	0.2187	0.0099	-0.0238	0.0004	0.0115
<b>13.har.</b>	<b>15.har.</b>	<b>17.har.</b>	<b>19.har.</b>	<b>21.har.</b>	<b>23.har.</b>
-0.0092	-0.0294	-0.0010	0.0932	0.2179	0.3065

$$THD = \frac{\sqrt{0.0099^2 + 0.0238^2 + 0.0115^2 + 0.0092^2}}{1.1923} = 0.0248$$

Görüldüğü gibi genetik algoritmalar ile yapılan çözümde Toplam Harmonik Distorsiyonu daha düşük bulunmuştur. Yani harmonik bozulma daha azdır.

Lagrange yöntemine göre 'Tatar' in çözümü

<b>1.har.</b>	<b>3.har.</b>	<b>5.har.</b>	<b>7.har.</b>	<b>9.har.</b>	<b>11.har.</b>
1.2048	0.2909	0.0325	0.0247	0.0671	0.0117
<b>13.har.</b>	<b>15.har.</b>	<b>17.har.</b>	<b>19.har.</b>	<b>21.har.</b>	<b>23.har.</b>
0.0046	0.0487	0.0198	0.0276	0.0324	0.0261

Görüldüğü gibi GA' lara dayalı yapılan çözümde bastırılması istenen tüm harmonikler daha iyi bastırılmıştır. Ayrıca Lagrange yöntemiyle yapılan çözümde, başlangıç değerlerinin iyi seçilmesi zorunluluğu vardır. Elenmeyen ilk harmonik bileşenin değeri, başlangıç açısı değerinin seçimiyle doğrudan ilgilidir. Çözüm başlangıç değerleri etrafında aranacaktır. Başlangıç değerlerinin çok iyi seçilmesi gerekir. İyi seçilmeyen başlangıç değerleriyle, hem ana harmonik maksimum yapılamaz hem de diğer harmonikler bastırılmaz. Oysa GA' lara dayalı yapmış olduğumuz programda istenilen herhangi rast gele değerlerle en iyi çözüme gidilir. Başlangıç değerlerinin seçilmesi gibi bir dezavantaj yoktur. Program rast gele değerlerden çok geniş bir alanı tarayarak sonuca gider.

Burada dikkat edilmesi gereken önemli bir husus ta genetik algoritmalar ile yapılan çözümde bulunan ateşleme açılarından;

8.2813 13.2813 21.2813 21.2813 25.2813 27.2813

ikisinin birbirinin aynı olmasıdır. Bu 2 açı aynı ise fazladan bir açı almaya gerek kalmaz. Yani bu açılardan biri alınmadan da aynı çözüm bulunur. Öyleyse GA' nın bir diğer avantajının da daha az açı ile daha iyi çözümü bulabilmesi olduğu görülmektedir.

### 3.3 Çeşitli Örnekler

#### 3.3.1 Harmonik Ağırlıklarının Etkisi

Örnek 5.

Elimine edilecek harmonik sayısı:6

Popülasyon sayısı:10

Maksimum generasyon sayısı:11

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilsin

Elimine edilecek harmonik sayısından kaç fazla açı alınarak eliminasyon yapılacağı:2

Toplam görülmek istenen harmonik sayısı:12

3.harmonikten itibaren sırasıyla harmonik ağırlıklarını giriniz:

1 1 1 1 1 1

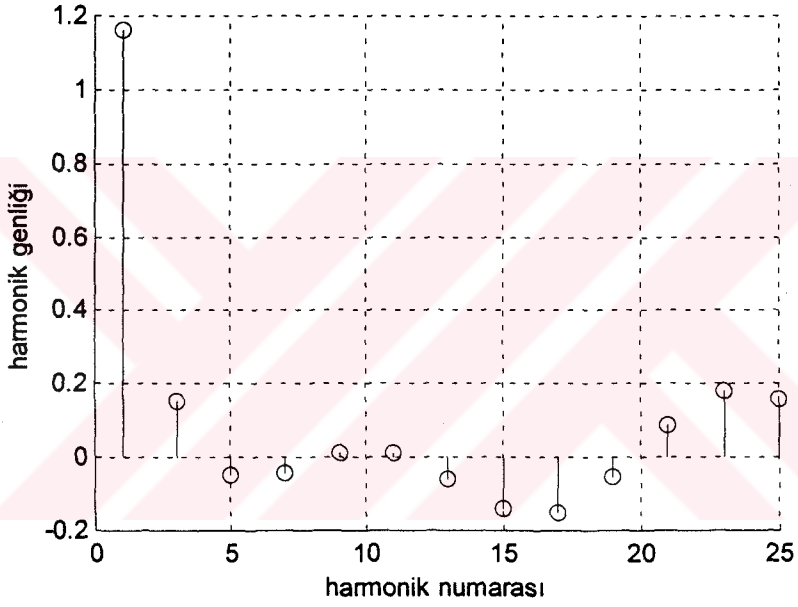
Değerleri girilerek GAHE programı çalıştırıldığında;

Çözüm açıları:

0.7188 4.0938 7.7188 11.0569 13.5097 14.9688 24.9688 28.1563

Tablo 3.19 Örnek 5 Çözüm Harmonikleri

<b>1.har.</b>	<b>3.har.</b>	<b>5.har.</b>	<b>7.har.</b>	<b>9.har.</b>	<b>11.har.</b>
1.1635	0.1525	-0.0495	-0.0448	0.0101	0.0116
<b>13.har.</b>	<b>15.har.</b>	<b>17.har.</b>	<b>19.har.</b>	<b>21.har.</b>	<b>23.har.</b>
-0.0599	-0.1413	-0.1485	-0.0528	0.0901	0.1812
<b>25.har.</b>					
0.1555					



Şekil 3.7 Örnek 5 Harmonik Spektrumu

Örnek 6.

Elimine edilecek harmonik sayısı:6

Popülasyon sayısı:10

Maksimum generasyon sayısı:11

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilsin

Elimine edilecek harmonik sayısından kaç fazla açı alınarak eliminasyon yapılacağı:2

Toplam görülmek istenen harmonik sayısı:12

3.harmonikten itibaren sırasıyla harmonik ağırlıklarını giriniz:

1.0000 0.8000 0.5000 0.4000 0.2000 0.1000

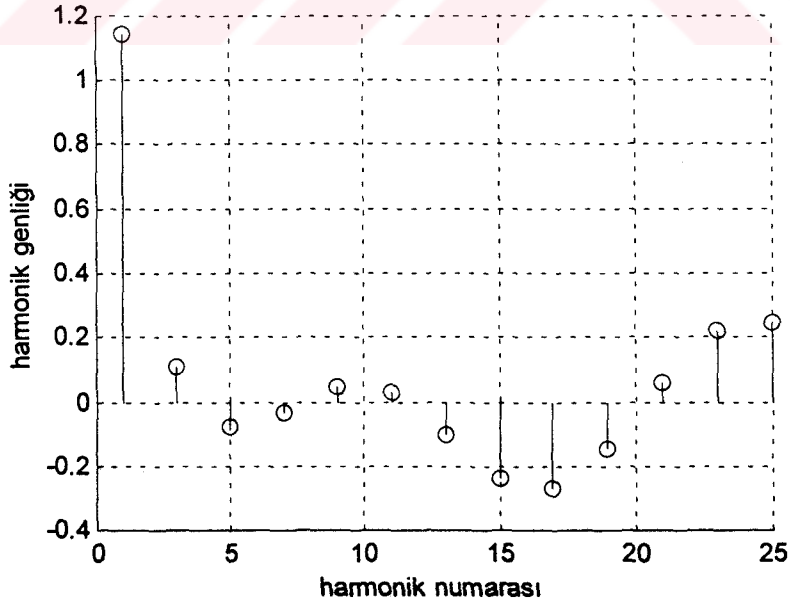
Değerleri girilerek GAHE programı çalıştırıldığında;

Çözüm açıları:

2.8449 3.7188 5.5313 11.3692 21.2188 21.4688 24.4688 28.7813

Tablo 3.20 Örnek 6 Çözüm Harmonikleri

<b>1.har.</b>	<b>3.har.</b>	<b>5.har.</b>	<b>7.har.</b>	<b>9.har.</b>	<b>11.har.</b>
1.1430	0.1088	-0.0786	-0.0359	0.0438	0.0278
<b>13.har.</b>	<b>15.har.</b>	<b>17.har.</b>	<b>19.har.</b>	<b>21.har.</b>	<b>23.har.</b>
-0.0998	-0.2408	-0.2716	-0.1473	0.0605	0.2208
<b>25.har.</b>					
0.2463					



Şekil 3.8 Örnek 6 Harmonik Spektrumu

Görüldüğü gibi 3. harmoniğe en büyük ağırlık verilerek 0.1525' den 0.1088' e düşürülmüştür.

Örnek 7.

Elimine edilecek harmonik sayısı:4

Popülasyon sayısı:10

Maksimum generasyon sayısı:10

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilsin

Elimine edilecek harmonik sayısından kaç fazla açılı alınarak eliminasyon yapılacağı:1

Toplam görülmek istenen harmonik sayısı:10

3.harmonikten itibaren sırasıyla harmonik ağırlıklarını giriniz:

1 1 1 1 1 1

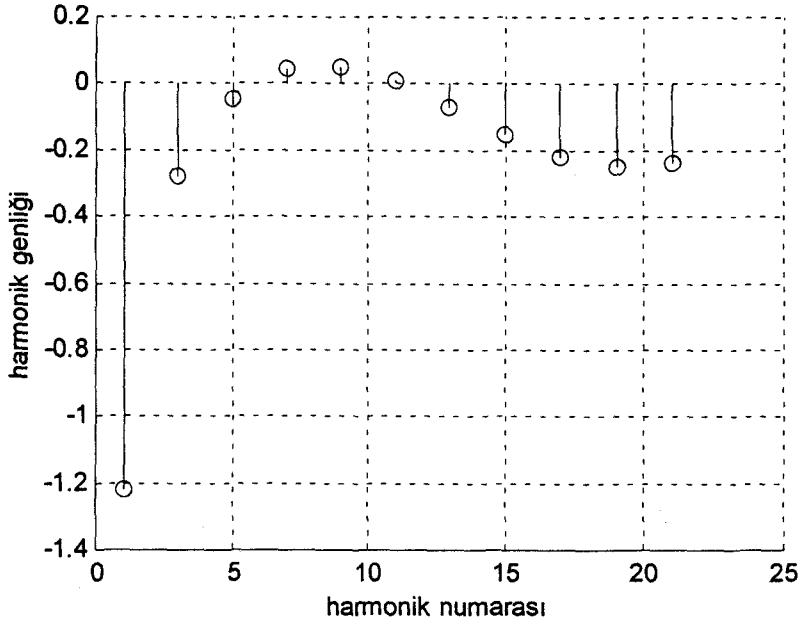
Çözüm açıları:

1.2188 10.9063 13.9688 13.9688 15.9688

Tablo 3.21 Örnek 7 Çözüm Harmonikleri

1.har.	3.har.	5.har.	7.har.	9.har.	11.har.
-1.2204	-0.2775	-0.0461	0.0430	0.0516	0.0058
13.har.	15.har.	17.har.	19.har.	21.har.	
-0.0708	-0.1531	-0.2182	-0.2492	-0.2384	





Şekil 3.9 Örnek 7 Harmonik Spektrumu

Örnek 8.

Elimine edilecek harmonik sayısı:4

Popülasyon sayısı:10

Maksimum generasyon sayısı:10

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilsin

Elimine edilecek harmonik sayısından kaç fazla açı alınarak eliminasyon yapılacağı:1

Toplam görülmek istenen harmonik sayısı:10

3.harmonikten itibaren sırasıyla harmonik ağırlıklarını giriniz:

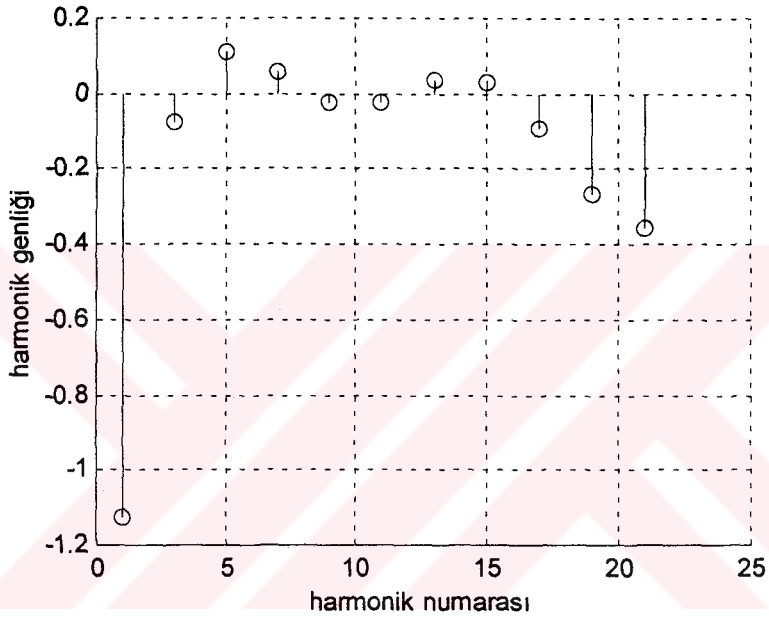
1.0000 0.4000 0.4000 0.1000

Çözüm açıları:

2.9688 9.9688 16.4688 27.4688 30.9688

Tablo 3.22 Örnek 8 Çözüm Harmonikleri

1.har.	3.har.	5.har.	7.har.	9.har.	11.har.
-1.1279	-0.0755	0.1094	0.0588	-0.0218	-0.0208
13.har.	15.har.	17.har.	19.har.	21.har.	
0.0336	0.0308	-0.0917	-0.2690	-0.3581	



Şekil 3.10 Örnek 8 Harmonik Spektrumu

Görüldüğü gibi burada da 3. harmoniğe en büyük ağırlık verilerek 0.2775' den 0.0755' e düşürülmüştür.

### 3.3.2 Eliminasyon İşleminde Açı Sayısının Etkisi

Örnek 9.

Elimine edilecek harmonik sayısı:5

Popülasyon sayısı:9

Maksimum generasyon sayısı:10

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilmesin

Elimine edilecek harmonik sayısından kaç fazla açılış alınarak eliminasyon yapılacağı:1  
 Toplam görülmek istenen harmonik sayısı:13

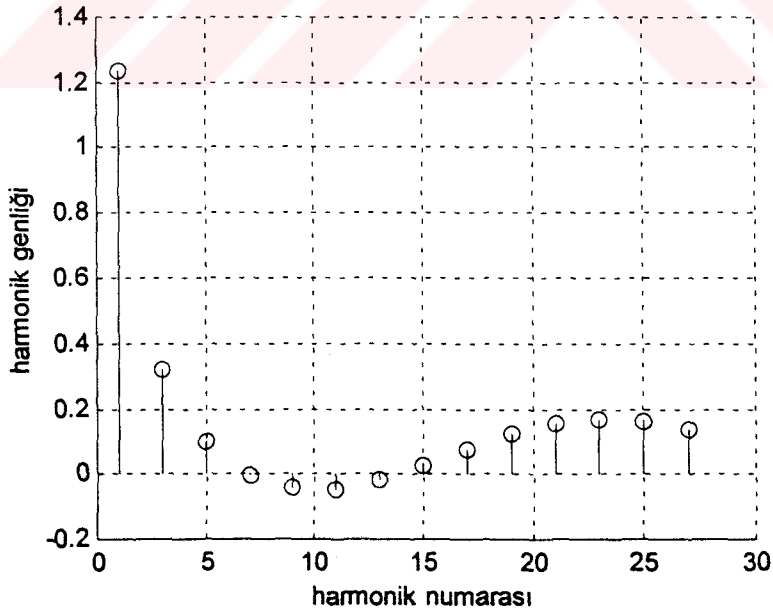
Değerleri girilerek GAHE programı çalıştırıldığında;

Çözüm açıları:

5.1888 5.2433 7.1563 9.0313 11.2475 13.8125

Tablo 3.23 Örnek 9 Çözüm Harmonikleri

<b>1.har.</b>	<b>3.har.</b>	<b>5.har.</b>	<b>7.har.</b>	<b>9.har.</b>	<b>11.har.</b>
1.2366	0.3200	0.0985	-0.0025	-0.0440	-0.0449
<b>13.har.</b>	<b>15.har.</b>	<b>17.har.</b>	<b>19.har.</b>	<b>21.har.</b>	<b>23.har.</b>
-0.0172	0.0276	0.0776	0.1223	0.1532	0.1656
<b>25.har.</b>	<b>27.har.</b>				
0.1587	0.1355				



Şekil 3.11 Örnek 9 Harmonik Spektrumu

Örnek 10.

Elimine edilecek harmonik sayısı:5

Popülasyon sayısı:9

Maksimum generasyon sayısı:10

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilmesin

Elimine edilecek harmonik sayısından kaç fazla açı alınarak eliminasyon yapılacağı:2

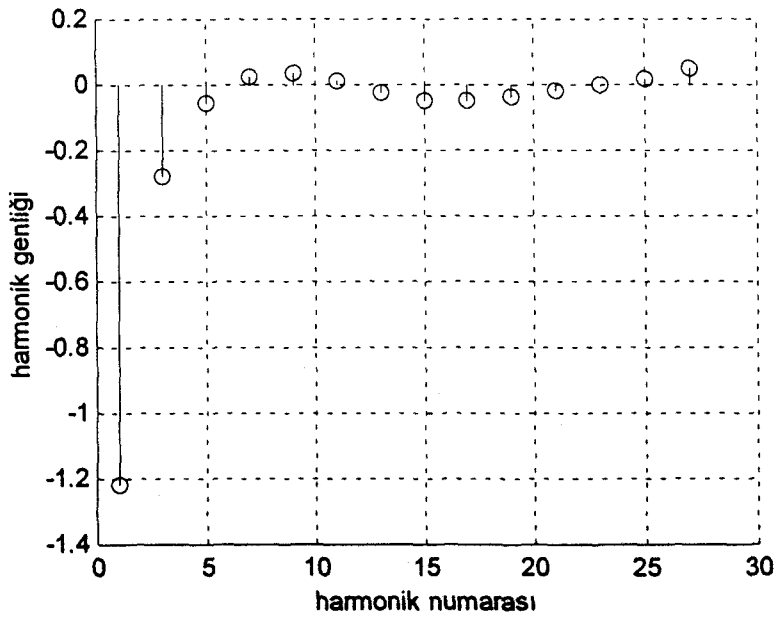
Toplam görülmek istenen harmonik sayısı:13

Çözüm açıları:

1.9062 1.9063 4.3750 11.3750 14.3750 20.4063 21.3750

Tablo 3.24 Örnek 10 Çözüm Harmonikleri

<b>1.har.</b>	<b>3.har.</b>	<b>5.har.</b>	<b>7.har.</b>	<b>9.har.</b>	<b>11.har.</b>
-1.2208	-0.2808	-0.0564	0.0254	0.0356	0.0100
<b>13.har.</b>	<b>15.har.</b>	<b>17.har.</b>	<b>19.har.</b>	<b>21.har.</b>	<b>23.har.</b>
-0.0235	-0.0459	-0.0500	-0.0388	-0.0201	0.0000
<b>25.har.</b>	<b>27.har.</b>				
0.0212	0.0480				



Şekil 3.12 Örnek 10 Harmonik Spektrumu

Görüldüğü gibi 2 fazla açı alarak yaptığımız örnekte GA yine çözüme ulaşmış ancak 1 fazla açı alarak yaptığımız örneğe göre farklı açılar bularak eliminasyon yapmıştır. 2 fazla açı alarak yaptığımız örnekte çözüm açılarına dikkat edecek olursak; Çözüm açıları:

1.9062 1.9063 4.3750 11.3750 14.3750 20.4063 21.3750

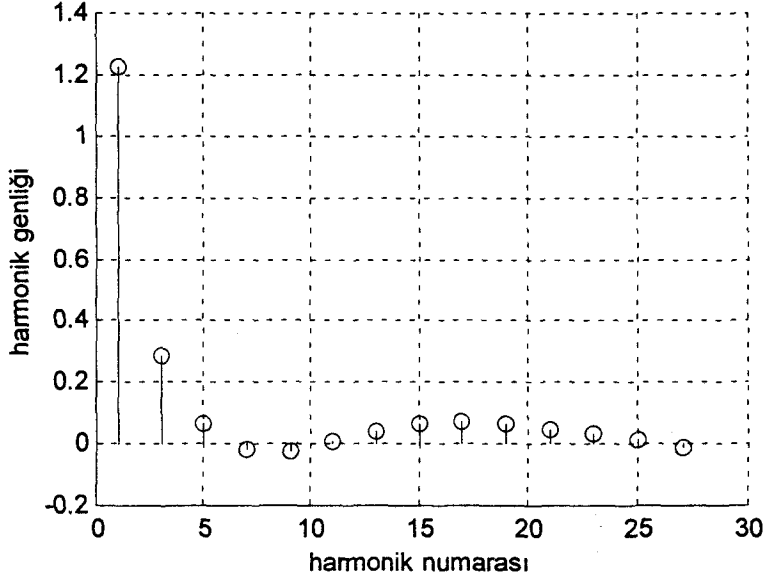
Burada 1. ve 2. açı birbirine çok yakındır. Öyleyse bu açılardan herhangi birini almadan da eliminasyon yapılacaktır. Böylece daha az ateşleme açısıyla çok yakın sonuçlar bulunacaktır. Fazladan ateşleme açısı almaya gerek olmaz. Yani anahtarlama sayısı artmayacak ve dolayısıyla anahtarlama kayıpları olmayacaktır. Yani harmonik kayıplar ile yüksek anahtarlama frekansından kaynaklanan anahtarlama kayıplarını optimum bir noktada dengeleyecek şekilde en az anahtarlama yapılmalıdır. Anahtarlama sayısının az olması kullanılacak mikroişlemcinin de hızının artmasına sebep olacaktır.

2. açıyı almazsak çözüm şu şekilde olacaktır.

1.9062 4.3750 11.3750 14.3750 20.4063 21.3750

**Tablo 3.25 Çözüm Açılarından Aynı Olan Açılardan Birini Almadan Hesaplanan Çözüm Harmonikleri**

<b>1.har.</b>	<b>3.har.</b>	<b>5.har.</b>	<b>7.har.</b>	<b>9.har.</b>	<b>11.har.</b>
1.2222	0.2851	0.0635	-0.0155	-0.0230	0.0053
<b>13.har.</b>	<b>15.har.</b>	<b>17.har.</b>	<b>19.har.</b>	<b>21.har.</b>	<b>23.har.</b>
0.0415	0.0666	0.0734	0.0647	0.0485	0.0308
<b>25.har.</b>	<b>27.har.</b>				
0.0121	-0.0124				



Şekil 3.13 Çözüm Açılarından Aynı Olan Açılardan Birini Almadan Hesaplanan Harmonik Spektrumu

Kendi yaptığım örneklerde elimine edilecek açı sayısından fazla açı alınarak yapılan eliminasyonda, açı sayısının artırılmasının genetik algoritmada çok büyük bir avantaj olmadığını gördüm. Çünkü genetik algoritma 1 fazla açı aldığı zaman da geniş bir alanı tarayarak en iyi çözümü bulabilmektedir. Başka çözüm yöntemlerinde en iyi sonucu bulmak için bu açı sayısının artırılması zorunluluğu vardır. Ancak genetik algoritma herhangi bir yerde sıkışmaz, aradığı alanı geniş tutar. Ve bu alandaki en iyi sonucu her zaman bulur. Bu nedenle açı sayısının artırılması genetik algoritmaya pek bir fayda sağlamamaktadır. Çünkü çözümü zaten her zaman bulmaktadır. Öyleyse fazla açı alınmasına gerek yoktur ve dolayısıyla fazla açı alınarak, daha fazla ortaya çıkacak anahtarlama kayıpları da önlenmiş olunacaktır.

### 3.3.3 Farklı Başlangıç Değerlerinin Etkisi

Örnek 11.

Elimine edilecek harmonik sayısı:4

Popülasyon sayısı:10

Maksimum generasyon sayısı:10

3 ve 3'ün katları harmoniklerin elimine edilip edilmeyeceği:Edilsin

Elimine edilecek harmonik sayısından kaç fazla açı alınarak eliminasyon yapılacağı:1

Toplam görülmek istenen harmonik sayısı:10

3.harmonikten itibaren sırasıyla harmonik ağırlıklarını giriniz:

1.0000 0.4400 0.3400 0.2400

Değerleri girilerek GAHE programı çalıştırıldığında;

Çözüm açıları:

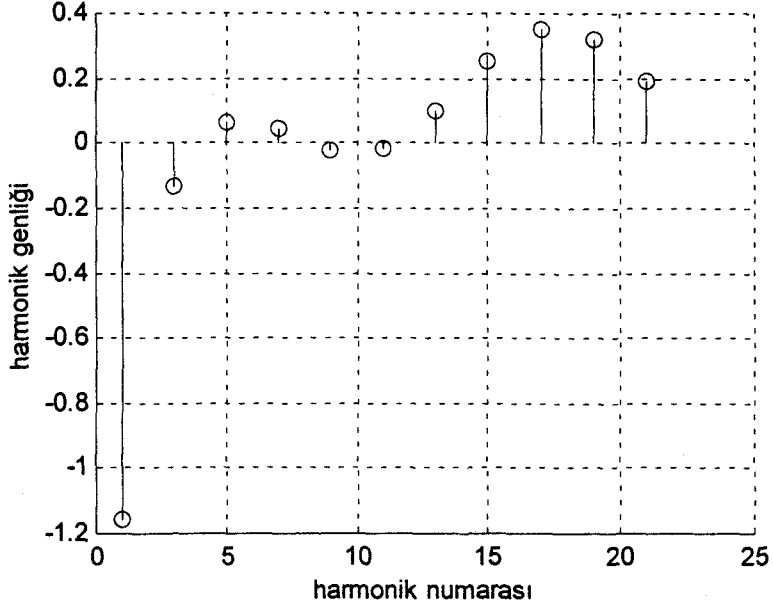
9.2188 18.4688 19.1563 23.8438 27.7188

Tablo 3.26 Örnek 11 Çözüm Harmonikleri

1.har.	3.har.	5.har.	7.har.	9.har.	11.har.
-1.1556	-0.1342	0.0653	0.0469	-0.0212	-0.0159
13.har.	15.har.	17.har.	19.har.	21.har.	
0.0978	0.2559	0.3523	0.3225	0.1921	

Tablo 3.27 Örnek 11 Başlangıç Değerleri

Popülasyon	Anahtarlama Açılı				
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı
1.pop	33.2036	44.7685	67.9165	84.7322	85.6613
2.pop	9.2704	12.7350	30.3262	61.0919	76.9551
3.pop	24.9179	34.0996	55.9959	62.6275	86.3103
4.pop	32.6006	51.2686	51.6696	64.4436	73.6609
5.pop	25.9790	39.2175	57.9546	64.2511	74.2558
6.pop	21.0544	50.6749	58.7699	62.6693	78.5449
7.pop	3.5786	17.6130	32.4534	76.5357	87.3717
8.pop	16.4036	27.1610	69.4590	69.7781	80.7962
9.pop	9.5544	39.7985	81.6625	86.0363	89.7215
10.pop	12.2477	50.5863	62.2798	66.3118	86.3493



Şekil 3.14 Örnek 11 Harmonik Spektrumu

## Örnek 12.

Aynı değerlerle GAHE programı tekrar çalıştırıldığında farklı başlangıç değerleri alacak ve çözümü bulacaktır.

Çözüm açıları:

8.3750 18.4779 18.8495 22.5574 28.1302

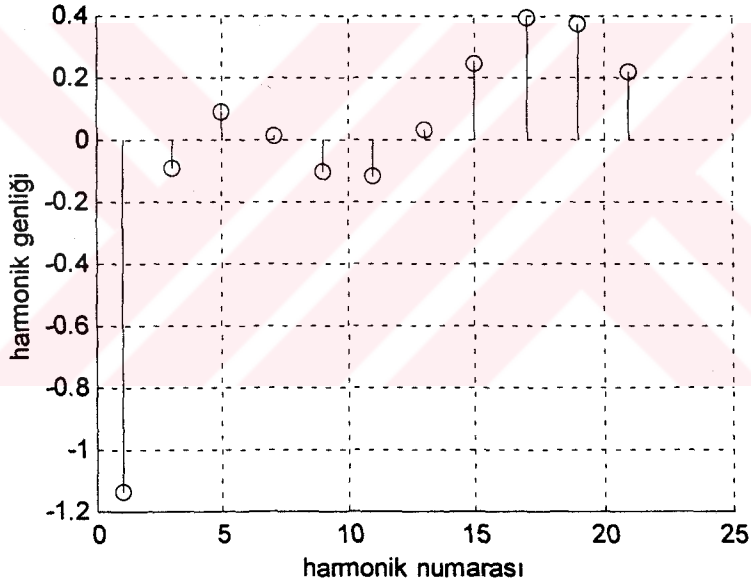
Tablo 3.28 Örnek 12 Çözüm Harmonikleri

1.har.	3.har.	5.har.	7.har.	9.har.	11.har.
-1.1348	-0.0909	0.0885	0.0160	-0.1050	-0.1160
13.har.	15.har.	17.har.	19.har.	21.har.	
0.0290	0.2455	0.3905	0.3730	0.2183	



Tablo 3.29 Örnek 12 Başlangıç Değerleri

Popülasyon	Anahtarlama Açılı				
	1.Açı	2.Açı	3.Açı	4.Açı	5.Açı
1.pop	24.8309	54.0479	55.3082	77.3869	84.5206
2.pop	41.1784	54.0196	57.7314	63.1059	71.2743
3.pop	55.3179	61.9408	70.8401	81.7655	85.6395
4.pop	12.3946	38.7237	55.3325	78.0707	80.3784
5.pop	59.2631	65.4434	67.7861	70.9495	84.9612
6.pop	12.5485	36.7622	36.7970	37.9054	50.4792
7.pop	2.7883	9.3704	19.3849	38.1100	56.4024
8.pop	5.5061	48.0075	49.8366	60.9109	64.0357
9.pop	2.2802	37.4214	48.4355	57.9596	67.7055
10.pop	20.6161	33.7085	52.1420	64.3403	80.6238



Şekil 3.15 Örnek 12 Harmonik Spektrumu

Görüldüğü gibi farklı başlangıç değerlerinde de genetik algoritma en iyi çözümü bulabilmektedir. Çünkü GA hangi başlangıç değerleri olursa olsun popülasyon büyüklüğüne göre çok geniş bir çözüm havzasını tarayacaktır. Bu çözüm havzasının içerinden her zaman en iyi sonuca gider. Ben kendi yaptığım diğer uygulamalarda da hangi başlangıç değerleri olursa olsun GA'nın her zaman en iyi sonuca gittiğini ve

başarıyla eliminasyon yaptığını gördüm. Yukarıdaki örnekte de bunun bir örneği gösterilmiş ve farklı başlangıç değerleriyle yine iyi bir çözüm bulunmuştur.

GA' da çözüme yaklaşmayı sağlayan unsurlar popülasyon büyüklüğü ve maksimum generasyon sayısıdır. Popülasyon büyüklüğünü ve maksimum generasyon sayısı çok küçük verilirse GA girilen dar alan kadarda çaprazlama ve mutasyon yapacak ve belki de yine de uygun iyi bir çözüm bulacaktır. Ancak genel olarak bu 2 faktör çok küçük verilirse, popülasyon büyüklüğünün küçük olmasından dolayı dar bir alanda arama yapmak zorunda kalacak, bu dar alandaki en iyi sonucu alacak, maksimum generasyon sayısının küçük olmasından dolayı da dar alanı genişletemeyecektir. Çaprazlama ve mutasyon işlemleri her iterasyonda olmakta olduğundan ve iterasyon sayısı az olduğunda çaprazlama ve mutasyon sınırlı şekilde yapacağından yeni varyasyonlar, yeni çözümler deneyemeyecektir.

Popülasyon büyüklüğü GA' nın performansını etkileyen bir unsurdur. Popülasyon büyüklüğü seçiminde, yapılan işlemlerin karmaşıklığı ve aramanın derinliği önemlidir. Popülasyonu daha da büyültmek çalışma zamanını uzatır, bunun yanında azaltmak ise kromozomların çeşitliliğinde eksiklik göstereceğinden çözümün hassasiyeti azalır. GA dizaynında bir problemin çözümünü bulmaya gerekli olan generasyonların sayısına karşılık, popülasyon büyüklüğünün dengelenmesi gerekir. Örneğin, 100 kromozomlu bir popülasyon, çözümü sadece 10 generasyonla bulabilir. Buna karşılık 20 generasyonda çözümü bulan 20 kromozomlu bir popülasyondan 4 kat daha uzun zaman alabilir. Böylelikle popülasyon boyutu daha küçük olan, diğerine göre çözümü daha fazla generasyon sayısında bulmasına rağmen süre açısından daha verimlidir.

#### 4. SONUÇ VE TARTIŞMA

Tezde elde edilen sonuçlar, özellikle 3. bölümde, çözülen problemler karşılaştırmalı olarak verilirken anlatılmıştır. Ayrıca bu bölümde özellikle maddeler halinde aşağıdaki sonuçlar vurgulanabilir.

- a- Bölüm 3.2' de önceden çözülmüş bir problem GAHE kullanılarak çözülmüş, her iki sonucun Toplam Harmonik Distorsiyonu (THD) karşılaştırılmış, GAHE ile bulunan sonucun daha iyi olduğu görülmüştür. Yine aynı problem 6 açı yerine 5 açıyla da GAHE ile çözümlenerek, kabul edilebilir iyi bir çözüm bulunmuştur. Bu da DC-AC çeviricinin gerçekleştirilmesinde kolaylık sağlayacaktır.
- b- Uygunluk fonksiyonunda bastırılacak harmoniklere verilen ağırlık katsayıları uygun seçilerek, bastırılacak harmoniklerin içinde istenilen harmoniğin daha da küçük yapılabileceği Bölüm 3.3.1' de gösterilmiştir.
- c- Hazırlanan program çalıştırılmaya başlatıldığında kullanıcıya sorulan seçenekler sayesinde, kullanıcı bastırılacak harmoniklerin hangilerinin özellikle daha çok bastırılması gerektiğini seçebileceği gibi, açı sayısını da belirleyebilir. Böylece elde edilen program esnek bir yapı kazanmıştır.
- d- Program aynı problem için her çalıştırıldığında farklı başlangıç popülasyonlarıyla başladığı ve program içerisinde rast gele bitler seçerek çaprazlama ve gerekiyorsa mutasyon işlemi yaptığı halde, hemen hemen aynı sonuçlara ulaşmaktadır. Bu durum Bölüm 3.3.3' de anlatılmıştır.
- e- Klasik optimizasyon yöntemlerinde başlangıç değerlerinin iyi seçilmesi zorunluluğu varken, Genetik Algoritmalar ile yapılan problem çözümünde bu zorunluluk yoktur. Hangi başlangıç değerleri olursa olsun çözüm bulunur. Bu algoritma özellikle başlangıç değerlerinin iyi seçilmesinin gerektiği, uğraşılan bağıntıların analitik türevlerinin zor olduğu problemlerde üstünlük sağlar.

İleri bir çalışma konusu olarak, sürülecek yükün özelliğine göre girişte kullanıcıya sorulan ağırlıkların sistematik bir şekilde hesaplanma yöntemleri araştırılabilir. Ayrıca programın GAHE' nin sayıları ikilikten ondalığa, ondalıktan ikiliğe dönüştürme,

mutasyon, aprazlama, uygunluk fonksiyonunun belirlenmesi kısımları alt programlar halinde dzenlenebilir. Bylece program sadece bařka bir problem iin yazılması gereken uygunluk fonksiyonunun tanımlanmasıyla, herhangi bir problem iin kullanılacak bir paket program haline getirilebilir.



## KAYNAKLAR

Argın, M., Güç Sistem Harmonik Fitreleri, Master Tezi, YTÜ, Fen Bilimleri Enstitüsü, 2000

Ataman, F., Kaynak, T. ve Yüncü, S., Bilgisayar Ortamında Modellenen Bir sistemin Optimum Konumunu Yapay Sinir Ağları ve Genetik Algoritma Yardımı ile Bulan Bir Program, 5. Bilgisayar-Haberleşme Sempozyumu, pp. 80-83, Bursa, 1998

Aygen, E., Bağrıyanık, M. ve Tarkan, N., Generatörler Arasında Ekonomik Yük Dağıtım Problemine Genetik Algoritma Uygulaması, Elektrik-Elektronik Bilgisayar Mühendisliği 8. Ulusal Kongresi, Cilt 1, pp. 329-332, Gaziantep, 1999

Akdağlı, A., Güney, K., Belirli Doğrultularda Sıfırlara Sahip Anten Diyagram Sentezinin Genetik Algoritma ile Gerçekleştirilmesi, 5. Bilgisayar-Haberleşme Sempozyumu, pp.15-18, Bursa 1998

Bağış, A., Genetik Algoritma Kullanılarak Gerçekleştirilen Sistem Kimliklendirme İşleminde Çaprazlama ve Mutasyon Operatörlerinin Etkisi, 5. Bilgisayar-Haberleşme Sempozyumu, pp. 222-225, Bursa, 1998

Bunday, D.b., Basic Optimisation Methods, Edward Arnold, London, 1984

Davis, L., Handbook of Genetic Algorithms, Van Nostrand, NewYork, 1991

Goldberg, D.E., Genetic Algorithms in Search, Optimisation and Machine Learning, Addison Wesley Publishing Co., Inc. USA, 1989

Gonzales, D.A and McCall, J.C, Design of Filters to Reduce Harmonic Distribution in Industrial Power Systems, IEEE TRANSACTIONS ON POWER INDUSTRY APPLICATIONS, Vol. IA-23, No:3, pp 504-511,1987

Gürdal, O., Güç Elektroniği (Analiz Tasarım, Simülasyon), Ankara, 2000

Holland, J.H., Genetic Algorithms, Scientific American, July 1992, pp. 66-72

IEEE Recommended Practices and Requirements for Harmonic Control in Electrical Power Systems, ANSI/IEEE Standart 519,1992.

İnce, M.C., Mikroişlemci Algoritma ile Optimizasyon İçin Bir Fortran Bilgisayar Programı, 4. Bilgisayar-Haberleşme Sempozyumu, pp.107-110, Bursa, 1996

İnce, M.C., Mikroişlemci Kontrollü Anahtarlanmış Kapasitör Filtrelerinin Tasarımı ve Gerçekleştirilmesi, Doktora Tezi, F.Ü., Elazığ, 1994

Jenkis, W.M., Towards Structural Optimisation via the Genetic Algorithm, Computers Structures, No.5, pp. 321-1327, Great Britain, 1991

Kaynak, O.M., Güç Elektroniği, Boğaziçi Üniv. İstanbul, 1988

Markam, E.B., Subramaniam, E.V., Girgis A.A. and Catoe, R., Harmonic Filter Design Using Actual Recorded data, IEEE TRANSACTIONS ON INDUSTRY APPLICATIONS, Vol.29, no 6, pp 1176-1183,1993

Ortmeyer.A., et al, 'Modeling and Simulation of the Propagation of Harmonics in Electric Power System Equipment and Loads'. IEEE Trans. On Power Apparatus and Systems. Vol.PAS-104,No.9,1985

Rashid, M.H., Power Electronics, Circuits, Devices and Applications, Prentice-Hall, New Jersey, 1993

Tang, K.S., Man, K.F. and Kwang, S., Genetic Algorithms: Concepts and Applications, IEEE Transactions on Industrial Electronics, Vol.43, No.5, October 1996, pp. 519-534

Tatar, Y., Sayısal İşaret İşlemci Temelli Darbe Genişlik Modülasyonlu Frekans Eviricisi ile Sincap kafesli Asenkron Motor Kontrolü İçin İşaret Üretimi, Doktora Tezi, F.Ü., Elazığ 1994.

Wu, J.C. and Jou, H.L., 1995. A new UPS Scheme Provides Harmonic Suppression and Input Power Factor Correction. IEEE Transactions on Industrial Electronics, Vol.42, No.6, p.629-635, Taiwan

EK-1.

**Genetik Algoritmalar ile Dc-Ac Çeviricilerde Harmonik Eliminasyonunu Gerçekleştiren MATLAB Programı (GAHE)**

```

hs=input('Elimine edilecek harmonik sayısını giriniz:')
pop=input('Popülasyon büyüklüğünü giriniz:')
gensay=input('Maksimum generasyon sayısını giriniz:')
kat=input('3 ve 3 ün katları harmonikleri elimine etmek istiyor musunuz?:\n(Evet için 1, Hayır için 2 yazınız)\n')
kac=input('Elimine edilecek harmonik sayısından kaç fazla açı alınsın:')
harg=input('Toplam kaç tane harmonik görmek istiyorsunuz:')
katsay=0;
for gd=1:1:hs
katsay(gd)=input('3.harmonikten itibaren sırasıyla harmonik ağırlıklarını giriniz:')
end
gsay=1;
for fr=1:1:gensay
for k=((gsay-1)*pop)+1:1:(gsay*pop)
if gsay==1
accd=rand(1,(hs+kac));
accdv=accd*90;
ssfraccdv=sort(accdv);
for ds=1:1:(hs+kac)
ata(k,ds)=ssfraccdv(ds);
end
end
q=1;
gd=1;
if kat~=1
yfr=mod(hs,2);
if yfr==0
snnb=(hs*3)+1;
else
snnb=(hs*3)+2;
end
for n=1:2:snnb
rtfy=mod(n,3);
if rtfy~=0
bcd=0;
for ds=1:1:(hs+kac)
tc=mod(ds,2);
if tc==0
bcd=bcd+8*cos(n*ata(k,ds)*pi/180)/(n*pi);
else
bcd=bcd-8*cos(n*ata(k,ds)*pi/180)/(n*pi);

```

```

end
end
if q==1
har(k,q)=abs((4/(n*pi))+bcd);
end
if q~=1
har(k,q)=(abs((4/(n*pi))+bcd))*katsay(gd);
gd=gd+1;
end
q=q+1;
end
end
if kat==1
for n=1:2:(2*hs)+1
bcd=0;
for ds=1:1:(hs+kac)
tc=mod(ds,2);
if tc==0
bcd=bcd+8*cos(n*ata(k,ds)*pi/180)/(n*pi);
else
bcd=bcd-8*cos(n*ata(k,ds)*pi/180)/(n*pi);
end
end
if q==1
har(k,q)=abs((4/(n*pi))+bcd);
end
if q~=1
har(k,q)=(abs((4/(n*pi))+bcd))*katsay(gd);
gd=gd+1;
end

q=q+1;
end
end
tophar=0;
for n=2:1:(hs+1)
tophar=tophar+har(k,n);
end
amac(k)=har(k,1)-tophar;
end
topharmaks=0;
for n=3:2:(2*hs+1)
topharmaks=topharmaks+(1-((hs+1)*2)*4/(n*pi));
end
uygtop=0;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
x=(1+((hs+1)*2*4/pi))-topharmaks;

```



```

uyg(k)=amac(k)+x;
uygtop=uygtop+uyg(k);
end
uygunluk=sort(uyg);
uygort(gsay)=uygtop/pop;
uygmaks=max(uyg);
tan=uygort(gsay)/(uygmaks-uygort(gsay));
olcuygtop=0;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
olcuyg(k)=(tan*(uyg(k)-uygort(gsay)))+uygort(gsay);
olcuygtop=olcuygtop+olcuyg(k);
if olcuyg(k)<0
olcuyg(k)=0;
end
end
sn=1;
ssff=0;
fpt=0;
snk=1;
olceklenmsfsuyg=sort(olcuyg);
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
skop=pop*olcuyg(k)/olcuygtop;
kopsa(k)=round(skop);
if kopsa(k)==1 | kopsa(k)==-2
kopsa(k)=0;
end
end
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
if kopsa(k)==0
ssff=ssff+1;
g(sn)=k;
sn=sn+1;
end
if kopsa(k)==2
fpt=fpt+1;
gk(snk)=k;
snk=snk+1;
end
end
for a=(((gsay-1)*pop)+1):1:(gsay*pop)-1
for b=(gsay*pop):-1:(a+1)
if olcuyg(a)==olcuyg(b)
olcuyg(a)=olcuyg(a)-0.0001;
end
end
end
h=1;
for f=(((gsay-1)*pop)+1):1:(gsay*pop)

```

```

a(h)=olcuyg(f);
h=h+1;
end
hh=sort(a);
if ssff==0
abx=hh(1);
for y=((gsay-1)*pop)+1:1:(gsay*pop)
if olcuyg(y)==abx
kopsa(y)=0;
ssff=ssff+1;
end
end
end
if fpt==0
abp=hh(h-1);
for y=((gsay-1)*pop)+1:1:(gsay*pop)
if olcuyg(y)==abp
kopsa(y)=2;
fpt=fpt+1;
end
end
end
if ssff>fpt
if (2*ssff)<=pop
h=1;
for f=(((gsay-1)*pop)+1):1:(gsay*pop)
a(h)=olcuyg(f);
h=h+1;
end
hh=sort(a);
n=1;
for r=pop:-1:pop-ssff+1
rk(n)=hh(r);
n=n+1;
end
for f=1:1:ssff
for y=((gsay-1)*pop)+1:1:(gsay*pop)
if olcuyg(y)==rk(f)
kopsa(y)=2;
end
end
end
n=1;
kr=pop-(2*ssff);
if kr>0
for tv=1:1:kr
krt(n)=hh(tv+ssff);
n=n+1;

```

```

end
for f=1:1:kr
for y=((gsay-1)*pop)+1:1:(gsay*pop)
if olcuyg(y)==krt(f)
kopsa(y)=1;
end
end
end
end
end
if (2*ssff)>pop
h=1;
for f=(((gsay-1)*pop)+1):1:(gsay*pop)
a(h)=olcuyg(f);
h=h+1;
end
hh=sort(a);
n=1;
for f=1:1:fpt
rh(n)=(hh(f));
n=n+1;
end
for f=1:1:fpt
for y=((gsay-1)*pop)+1:1:(gsay*pop)
if olcuyg(y)==rh(f)
kopsa(y)=0;
end
end
end
n=1;
kr=pop-(2*fpt);
if kr>0
for tv=1:1:kr
krt(n)=hh(tv+fpt);
n=n+1;
end
for f=1:1:kr
for y=((gsay-1)*pop)+1:1:(gsay*pop)
if olcuyg(y)==krt(f)
kopsa(y)=1;
end
end
end
end
end
if ssff<fpt
if (2*fpt)<=pop

```

```

h=1;
for f=((gsay-1)*pop)+1:1:(gsay*pop)
a(h)=olcuyg(f);
h=h+1;
end
hh=sort(a);
n=1;
for r=1:1:fpt
rk(n)=(hh(r));
n=n+1;
end
for f=1:1:fpt
for y=((gsay-1)*pop)+1:1:(gsay*pop)
if olcuyg(y)==rk(f)
kopsa(y)=0;
end
end
end
for a=((gsay-1)*pop)+1:1:(gsay*pop)
if kopsa(a)~=2 & kopsa(a)~=0
kopsa(a)=1;
end
end
n=1;
kr=pop-(2*fpt);
if kr>0
for tv=1:1:kr
krt(n)=hh(tv+fpt);
n=n+1;
end
for f=1:1:kr
for y=((gsay-1)*pop)+1:1:(gsay*pop)
if olcuyg(y)==krt(f)
kopsa(y)=1;
end
end
end
end
if (2*fpt)>pop
h=1;
for f=((gsay-1)*pop)+1:1:(gsay*pop)
a(h)=olcuyg(f);
h=h+1;
end
hh=sort(a);
n=1;
for r=pop:-1:pop-ssff+1

```

```

rk(n)=hh(r);
n=n+1;
end
for f=1:1:ssff
for y=((gsay-1)*pop)+1:1:(gsay*pop)
if olcuyg(y)==rk(f)
kopsa(y)=2;
end
end
end
for a=((gsay-1)*pop)+1:1:(gsay*pop)
if kopsa(a)~=2 & kopsa(a)~=0
kopsa(a)=1;
end
end
n=1;
kr=pop-(2*ssff);
if kr>0
for tv=1:1:kr
krt(n)=hh(tv+ssff);
n=n+1;
end
for f=1:1:kr
for y=((gsay-1)*pop)+1:1:(gsay*pop)
if olcuyg(y)==krt(f)
kopsa(y)=1;
end
end
end
end
end
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
if kopsa(k)==-1|kopsa(k)==-2
kopsa(k)=0;
end
end
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
kopya(k)=kopsa(k);
end
%%%%%
ksay=0;
bsay=0;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
if kopya(k)==2
ksay=ksay+1;
end
if kopya(k)==1

```

```

bsay=bsay+1;
end
end

%%%%%
xm=1;
xn=1;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
for ds=1:1:(hs+kac)
if kopya(k)==2
cdvkcdv(xm)=ata(k,ds);
xm=xm+1;
end
if kopya(k)==1
bcdvr(xn)=ata(k,ds);
xn=xn+1;
end
end
end
end
f=1;
de=mod((xm+xn),2);
if xm>xn
cdv=1;
for ds=(xn-1):-1:1
bcdvttam=round(rand(1,1)*6)+1;
tamcdvkcdv=dec2bin(cdvkcdv(cdv),7);
tambcdvr=dec2bin(bcdvr(ds),7);
s=tamcdvkcdv(bcdvttam);
tamcdvkcdv(bcdvttam)=tambcdvr(bcdvttam);
cev(f)=bin2dec(tamcdvkcdv);
tambcdvr(bcdvttam)=s;
cev(f+1)=bin2dec(tambcdvr);
cdv=cdv+1;
f=f+2;
end
if de==0
ds=xn;
for cdv=(xm-1):-1:(((xm-1)+(xn-1))/2)+1
bcdvttam=round(rand(1,1)*6)+1;
tamcdvk=dec2bin(cdvkcdv(cdv),7);
tambcdv=dec2bin(cdvkcdv(ds),7);
s=tamcdvk(bcdvttam);
tamcdvk(bcdvttam)=tambcdv(bcdvttam);
cev(f)=bin2dec(tamcdvk);
tambcdv(bcdvttam)=s;
cev(f+1)=bin2dec(tambcdv);
ds=ds+1;
f=f+2;

```

```

end
end
if de~=0
ds=xm-1;
for cdv=xn:1:(((xm-1)+(xn-1))/2)-0.5
rr=round(rand(1,1)*6)+1;
rrr=dec2bin(cdvkcdv(cdv),7);
rrrr=dec2bin(cdvkcdv(ds),7);
s=rrr(rr);
rrr(rr)=rrrr(rr);
cev(f)=bin2dec(rrr);
rrrr(rr)=s;
cev(f+1)=bin2dec(rrrr);
ds=ds-1;
f=f+2;
end
kpptt=round(rand(1,1)*6)+1;
ktpptt=round(rand(1,1)*6)+1;
kppmcdv=dec2bin(cdvkcdv((((xm-1)+(xn-1))/2)+0.5),7);
s=kppmcdv(kpptt);
kppmcdv(kpptt)=kppmcdv(ktpptt);
cev(f)=bin2dec(kppmcdv);
f=f+1;
end
zz=mod((xm-1),2);
if zz==0
ds=xm-1;
for cdv=1:1:((xm-1)/2)
bcdvtt=round(rand(1,1)*6)+1;
tamcdv=dec2bin(cdvkcdv(cdv),7);
tamb=dec2bin(cdvkcdv(ds),7);
s=tamcdv(bcdvtt);
tamcdv(bcdvtt)=tamb(bcdvtt);
cev(f)=bin2dec(tamcdv);
tamb(bcdvtt)=s;
cev(f+1)=bin2dec(tamb);
ds=ds-1;
f=f+2;
end
end
if zz~=0
ds=xm-1;
for cdv=1:1:((xm-1)/2)-0.5
rr=round(rand(1,1)*6)+1;
rrcdv=dec2bin(cdvkcdv(cdv),7);
rrds=dec2bin(cdvkcdv(ds),7);
s=rrcdv(rr);
rrcdv(rr)=rrds(rr);

```

```

cev(f)=bin2dec(rrcdv);
rrds(rr)=s;
cev(f+1)=bin2dec(rrds);
ds=ds-1;
f=f+2;
end
cdvr=round(rand(1,1)*6)+1;
cdvrr=round(rand(1,1)*6)+1;
cdvrrr=dec2bin(cdvkcdv(xm/2),7);
s=cdvrrr(cdvr);
cdvrrr(cdvr)=cdvrrr(cdvrr);
cev(f)=bin2dec(cdvrrr);

end
end
if xn>xm
cdv=1;
for ds=(xm-1):-1:1
bcdvt=round(rand(1,1)*6)+1;
tam=dec2bin(bcdvr(cdv),7);
tamb=dec2bin(cdvkcdv(ds),7);
s=tam(bcdvt);
tam(bcdvt)=tamb(bcdvt);
cev(f)=bin2dec(tam);
tamb(bcdvt)=s;
cev(f+1)=bin2dec(tamb);
cdv=cdv+1;
f=f+2;
end
if de==0
ds=xm;
for cdv=(xn-1):-1:(((xm-1)+(xn-1))/2)+1
bcdv=round(rand(1,1)*6)+1;
ta=dec2bin(bcdvr(cdv),7);
tm=dec2bin(bcdvr(ds),7);
s=ta(bcdv);
ta(bcdv)=tm(bcdv);
cev(f)=bin2dec(ta);
tm(bcdv)=s;
cev(f+1)=bin2dec(tm);
ds=ds+1;
f=f+2;
end
end
if de~=0
ds=xn-1;
for cdv=xm:1:(((xm-1)+(xn-1))/2)-0.5
hrr=round(rand(1,1)*6)+1;

```



```

hrrr=dec2bin(bcdvr(cdv),7);
hrrrr=dec2bin(bcdvr(ds),7);
s=hrrr(hrr);
hrrr(hrr)=hrrrr(hrr);
cev(f)=bin2dec(hrrr);
hrrrr(hrr)=s;
cev(f+1)=bin2dec(hrrrr);
ds=ds-1;
f=f+2;
end
zpptt=round(rand(1,1)*6)+1;
ztpptt=round(rand(1,1)*6)+1;
zppmcdv=dec2bin(bcdvr((((xm-1)+(xn-1))/2)+0.5),7);
s=zppmcdv(zpptt);
zppmcdv(zpptt)=zppmcdv(ztpptt);
cev(f)=bin2dec(zppmcdv);
f=f+1;
end
zzz=mod((xm-1),2);
if zzz==0
ds=xm-1;
for cdv=1:1:((xm-1)/2)
bcdvttt=round(rand(1,1)*6)+1;
tamcdvcdv=dec2bin(cdvkcdv(cdv),7);
tambb=dec2bin(cdvkcdv(ds),7);
s=tamcdvcdv(bcdvttt);
tamcdvcdv(bcdvttt)=tambb(bcdvttt);
cev(f)=bin2dec(tamcdvcdv);
tambb(bcdvttt)=s;
cev(f+1)=bin2dec(tambb);
ds=ds-1;
f=f+2;
end
end
if zzz~=0
ds=xm-1;
for cdv=1:1:((xm-1)/2)-0.5
rrh=round(rand(1,1)*6)+1;
rrcdvh=dec2bin(cdvkcdv(cdv),7);
rrdsh=dec2bin(cdvkcdv(ds),7);
s=rrcdvh(rrh);
rrcdvh(rrh)=rrdsh(rrh);
cev(f)=bin2dec(rrcdvh);
rrdsh(rrh)=s;
cev(f+1)=bin2dec(rrdsh);
ds=ds-1;
f=f+2;
end

```

```

cdvrl=round(rand(1,1)*6)+1;
cdvrrl=round(rand(1,1)*6)+1;
cdvrrrl=dec2bin(cdvkcdv(xm/2),7);
s=cdvrrrl(cdvrl);
cdvrrrl(cdvrl)=cdvrrrl(cdvrrl);
cev(f)=bin2dec(cdvrrrl);
end
end
if xm==xn
cdv=1;
for ds=(xn-1):-1:1
tt=round(rand(1,1)*6)+1;
mcdv=dec2bin(cdvkcdv(cdv),7);
mb=dec2bin(bcdvvr(ds),7);
s=mcdv(tt);
mcdv(tt)=mb(tt);
cev(f)=bin2dec(mcdv);
mb(tt)=s;
cev(f+1)=bin2dec(mb);
cdv=cdv+1;
f=f+2;
end
den=mod((xm-1),2);
if den==0
ds=xm-1;
for cdv=1:1:(xm-1)/2
btt=round(rand(1,1)*6)+1;
tmcdv=dec2bin(cdvkcdv(cdv),7);
tmb=dec2bin(cdvkcdv(ds),7);
s=tmcdv(btt);
tmcdv(btt)=tmb(btt);
cev(f)=bin2dec(tmcdv);
tmb(btt)=s;
cev(f+1)=bin2dec(tmb);
ds=ds-1;
f=f+2;
end
end
if den~=0
ds=xm-1;
for cdv=1:1:((xm-1)/2)-0.5
ptt=round(rand(1,1)*6)+1;
pmcdv=dec2bin(cdvkcdv(cdv),7);
pmb=dec2bin(cdvkcdv(ds),7);
s=pmcdv(ptt);
pmcdv(ptt)=pmb(ptt);
cev(f)=bin2dec(pmcdv);
pmb(ptt)=s;

```

```

cev(f+1)=bin2dec(pmb);
ds=ds-1;
f=f+2;
end
pptt=round(rand(1,1)*6)+1;
tpptt=round(rand(1,1)*6)+1;
ppmcdv=dec2bin(cdvkcdv(xm/2),7);
s=ppmcdv(pptt);
ppmcdv(pptt)=ppmcdv(tpptt);
cev(f)=bin2dec(ppmcdv);
f=f+1;
end
end
ym=((hs+kac)*(pop)*(gsay-1))+1;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
for ds=1:1:(hs+kac)
if(ata(k,ds)-round(ata(k,ds)))>=0
ondalsfk=ata(k,ds)-round(ata(k,ds));
end
if(ata(k,ds)-round(ata(k,ds)))<0
ondalsfk=ata(k,ds)-round(ata(k,ds))-1;
end
for n=1:1:5
ondalsfk=ondalsfk*2;
if ondalsfk<1
tam(ym,n)=round(abs(ondalsfk-0.50000));
end
if ondalsfk>1
tam(ym,n)=round(abs(ondalsfk-0.50000));
ondalsfk=ondalsfk-1;
end
if ondalsfk==1
tam(ym,n)=1;
ondalsfk=ondalsfk-1;
end
end
ym=ym+1;
end
end
xmn=1;
xnm=1;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
for ds=1:1:(hs+kac)
if kopya(k)==1
vbcdivr(xmn)=k;
xmn=xmn+1;
end
if kopya(k)==2

```

```

vcdvkcdrv(xnm)=k;
xnm=xnm+1;
end
end
end
t=hs+kac;
vf=1;
if xnm>xmn
cdv=1;
for ds=(xmn-1):-1:1
ab=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vcdvkcdrv(cdv)-1)*(hs+kac);
sat=((vcdvkcdrv(ds)-1)*(hs+kac))+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
su=tam(sa,ab);
tam(sa,ab)=tam(sat,ab);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,ab)=su;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
cdv=cdv+1;
vf=vf+2;
end
t=hs+kac;
if (xnm-xmn)>(hs+kac)
kbsay=mod((ksay-bsay),2);
if kbsay==0
cdv=xmn;
for ds=(xnm-1):-1:((xmn-1+xnm-1)/2)+1
xy=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vcdvkcdrv(cdv)-1)*(hs+kac);
sat=(vcdvkcdrv(ds)-1)*(hs+kac)+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
s=tam(sa,xy);
tam(sa,xy)=tam(sat,xy);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);

```

```

tam(sat,xy)=s;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
ds=ds+1;
vf=vf+2;
end
end
t=hs+kac;
if kbsay~=0
tkb=((ksay-bsay)/2)-0.5;
cdv=xmn;
for ds=(xnm-1):-1:(xnm-1)-(((tkb)*(hs+kac))+1)
kl=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vcdvkcdv(cdv)-1)*(hs+kac);
sat=(vcdvkcdv(ds)-1)*(hs+kac)+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
s=tam(sa,kl);
tam(sa,kl)=tam(sat,kl);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,kl)=s;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
ds=ds-1;
vf=vf+2;
end

t=hs+kac;
for ad=1:1:(hs+kac)
st=round(rand(1,1)*4)+1;
stv=round(rand(1,1)*4)+1;
if t==hs+kac
cdv=(((ksay/2)-0.5)+1)*(hs+kac))+1;
sa=(vcdvkcdv(cdv)-1)*(hs+kac);
t=0;
end
sa=sa-1;
t=t+1;
s=tam(sa,st);
tam(sa,st)=tam(sa,stv);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
vf=vf+1;

```

```

end
end
end
t=hs+kac;
if (xnm-xmn)<=(hs+kac)
for ad=1:1:(hs+kac)
stp=round(rand(1,1)*4)+1;
stvp=round(rand(1,1)*4)+1;
if t==hs+kac
cdv=xmn;
sa=(vcdvkcdiv(cdv)-1)*(hs+kac);
t=0;
end
sa=sa-1;
t=t+1;
s=tam(sa,stp);
tam(sa,stp)=tam(sa,stvp);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
vf=vf+1;
end
end

zzp=mod((ksay),2);
t=hs+kac;
if zzp==0
ds=xnm-1;
for cdv=1:1:((ksay)/2)*(hs+kac)
fg=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vcdvkcdiv(cdv)-1)*(hs+kac);
sat=(vcdvkcdiv(ds)-1)*(hs+kac)+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
s=tam(sa,fg);
tam(sa,fg)=tam(sat,fg);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,fg)=s;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
ds=ds-1;
vf=vf+2;
end
end

```

```

t=hs+kac;
if zzp~=0
ds=xnm-1;
for cdv=1:1:((ksay/2)-0.5)*(hs+kac)
mmm=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vcdvkcdv(cdv)-1)*(hs+kac);
sat=(vcdvkcdv(ds)-1)*(hs+kac)+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
s=tam(sa,mmm);
tam(sa,mmm)=tam(sat,mmm);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,mmm)=s;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
ds=ds-1;
vf=vf+2;
end

t=hs+kac;
for af=1:1:(hs+kac)
pr=round(rand(1,1)*4)+1;
prs=round(rand(1,1)*4)+1;
if t==hs+kac
cdv=((ksay/2)-0.5)*(hs+kac)+1;
sa=(vcdvkcdv(cdv)-1)*(hs+kac);
t=0;
end
sa=sa++1;
t=t+1;
s=tam(sa,pr);
tam(sa,pr)=tam(sa,prs);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
vf=vf+1;
end
end
end
%%
if xnm<xmn
t=hs+kac;
cdv=1;
for ds=(xnm-1):-1:1

```

```

ab=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vbcdivr(cdv)-1)*(hs+kac);
sat=((vcdvkcdiv(ds)-1)*(hs+kac))+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
sul=tam(sa,ab);
tam(sa,ab)=tam(sat,ab);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,ab)=sul;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
cdv=cdv+1;
vf=vf+2;
end
t=hs+kac;
if (xmn-xnm)>(hs+kac)
bksay=mod((bsay-ksay),2);
if bksay==0
cdv=xmn-1;
for ds=(xnm):1:((xmn-1+xnm-1)/2)
xy=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vbcdivr(ds)-1)*(hs+kac);
sat=((vbcdivr(cdv)-1)*(hs+kac))+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
s=tam(sa,xy);
tam(sa,xy)=tam(sat,xy);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,xy)=s;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
vf=vf+2;
cdv=cdv-1;
end
end
t=hs+kac;
if bksay~=0
cdv=xnm;

```



```

tbk=((bsay-ksay)/2)-0.5;
for ds=(xmn-1):-1:(xmn-1)-(((tbk)*(hs+kac))+1)
kl=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vbcdvr(cdv)-1)*(hs+kac);
sat=(vbcdvr(ds)-1)*(hs+kac)+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
s=tam(sa,kl);
tam(sa,kl)=tam(sat,kl);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,kl)=s;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
cdv=cdv+1;
vf=vf+2;
end

t=hs+kac;
for ad=1:1:(hs+kac)
st=round(rand(1,1)*4)+1;
stv=round(rand(1,1)*4)+1;
if t==hs+kac
cdv=(((bksay/2)-0.5)+1)*(hs+kac))+1;
sa=(vbcdvr(cdv)-1)*(hs+kac);
t=0;
end
sa=sa+1;
t=t+1;
s=tam(sa,st);
tam(sa,st)=tam(sa,stv);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
vf=vf+1;
end
end
end
t=hs+kac;
if (xmn-xnm)<=(hs+kac)
for ad=1:1:(hs+kac)
stp=round(rand(1,1)*4)+1;
stvp=round(rand(1,1)*4)+1;
if t==hs+kac
cdv=xnm;

```

```

sa=(vbcdivr(cdv)-1)*(hs+kac);
t=0;
end
sa=sa-1;
t=t+1;
s=tam(sa,stp);
tam(sa,stp)=tam(sa,stv);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
vf=vf+1;
end
end
if ksay==1
t=hs+kac;
for r=1:1:(hs+kac)
pr=round(rand(1,1)*4)+1;
prs=round(rand(1,1)*4)+1;
if t==hs+kac
cdv=1;
sa=(vcdvkcdiv(cdv)-1)*(hs+kac);
t=0;
end
sa=sa+1;
t=t+1;
s=tam(sa,pr);
tam(sa,pr)=tam(sa,prs);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
vf=vf+1;
end
end
if ksay~=1
zzp=mod((ksay),2);
t=hs+kac;
if zzp==0
ds=xnm-1;
for cdv=1:1:((ksay)/2)*(hs+kac)
fg=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vcdvkcdiv(cdv)-1)*(hs+kac);
sat=(vcdvkcdiv(ds)-1)*(hs+kac)+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
s=tam(sa,fg);
tam(sa,fg)=tam(sat,fg);

```

```

vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,fg)=s;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
ds=ds-1;
vf=vf+2;
end
end
t=hs+kac;
if zzp~=0
ds=xnm-1;
for cdv=1:1:((ksay/2)-0.5)*(hs+kac)
mmmnn=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vcdvkcdiv(cdv)-1)*(hs+kac);
sat=(vcdvkcdiv(ds)-1)*(hs+kac)+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
s=tam(sa,mmmnn);
tam(sa,mmmnn)=tam(sat,mmmnn);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,mmmnn)=s;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
ds=ds-1;
vf=vf+2;
end

t=hs+kac;
for af=1:1:(hs+kac)
pr=round(rand(1,1)*4)+1;
prs=round(rand(1,1)*4)+1;
if t==hs+kac
cdv=(((ksay/2)-0.5)*(hs+kac))+1;
sa=(vcdvkcdiv(cdv)-1)*(hs+kac);
t=0;
end
sa=sa+1;
t=t+1;
s=tam(sa,pr);
tam(sa,pr)=tam(sa,prs);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);

```

```

vf=vf+1;
end
end
end
end

t=hs+kac;
if xnm==xmn
cdv=1;
for ds=(xmn-1):-1:1
zsf=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vcdvkcdv(cdv)-1)*(hs+kac);
sat=(vcdvvr(ds)-1)*(hs+kac)+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
s=tam(sa,zsf);
tam(sa,zsf)=tam(sat,zsf);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,zsf)=s;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
cdv=cdv+1;
vf=vf+2;
end
denp=mod((ksay),2);
t=hs+kac;
if denp==0
ds=xnm-1;
for cdv=1:1:(ksay/2)*(hs+kac)
xsf=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vcdvkcdv(cdv)-1)*(hs+kac);
sat=(vcdvkcdv(ds)-1)*(hs+kac)+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
s=tam(sa,xsf);
tam(sa,xsf)=tam(sat,xsf);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,xsf)=s;

```

```

vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
ds=ds-1;
vf=vf+2;
end
end
t=hs+kac;
if denp~=0
ds=xnm-1;
for cdv=1:1:(((ksay)/2)-0.5)*(hs+kac)
psf=round(rand(1,1)*4)+1;
if t==hs+kac
sa=(vcdvkcdv(cdv)-1)*(hs+kac);
sat=(vcdvkcdv(ds)-1)*(hs+kac)+hs+kac+1;
t=0;
end
sa=sa+1;
sat=sat-1;
t=t+1;
s=tam(sa,psf);
tam(sa,psf)=tam(sat,psf);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
tam(sat,psf)=s;
vcev(vf+1)=(tam(sat,1)*0.5+tam(sat,2)*0.25+tam(sat,3)*0.125+tam(sat,4)*0.0625+tam
(sat,5)*0.03125);
ds=ds-1;
vf=vf+2;
end
t=hs+kac;
for tr=1:1:(hs+kac)
lsf=round(rand(1,1)*4)+1;
lsfl=round(rand(1,1)*4)+1;
if t==hs+kac
cdv=(((ksay)/2)-0.5)+1)*(hs+kac);
sa=(vcdvkcdv(cdv)-1)*(hs+kac);
t=0;
end
sa=sa+1;
t=t+1;
s=tam(sa,lsf);
tam(sa,lsf)=tam(sa,lsfl);
vcev(vf)=(tam(sa,1)*0.5+tam(sa,2)*0.25+tam(sa,3)*0.125+tam(sa,4)*0.0625+tam(sa,5)
*0.03125);
vf=vf+1;
end
end
end

```

```

if gsay<=gensay
gsay=gsay+1;
f=1;
w=1;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
for ds=1:1:(hs+kac)
ata(k,ds)=cev(f)+vcev(f);
f=f+1;
if ata(k,ds)>90 | ata(k,ds)<0
ata(k,ds)=rand(1,1)*90;
end
a(w)=ata(k,ds);
w=w+1;
if w==hs+kac+1
w=1;
sty=sort(a);
for ds=1:1:(hs+kac)
ata(k,ds)=sty(ds);
end
end
end
end
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
q=1;
gd=1;
if kat~=1
yfr=mod(hs,2);
if yfr==0
snnb=(hs*3)+1;
else
snnb=(hs*3)+2;
end

for n=1:2:snnb
rtfy=mod(n,3);
if rtfy~=0

b=0;
for ds=1:1:(hs+kac)
tc=mod(ds,2);
if tc==0
b=b+8*cos(n*ata(k,ds)*pi/180)/(n*pi);
else
b=b-8*cos(n*ata(k,ds)*pi/180)/(n*pi);
end
end
if q==1
har(k,q)=abs((4/(n*pi))+bcd);

```

```

end
if q~=1
har(k,q)=(abs((4/(n*pi))+bcd))*katsay(gd);
gd=gd+1;
end

q=q+1;
end
end
end
if kat==1
for n=1:2:(2*hs)+1

bcd=0;
for ds=1:1:(hs+kac)
tc=mod(ds,2);
if tc==0
bcd=bcd+8*cos(n*ata(k,ds)*pi/180)/(n*pi);
else
bcd=bcd-8*cos(n*ata(k,ds)*pi/180)/(n*pi);
end
end
if q==1
har(k,q)=abs((4/(n*pi))+bcd);
end
if q~=1
har(k,q)=(abs((4/(n*pi))+bcd))*katsay(gd);
gd=gd+1;
end

q=q+1;
end
end
tophar=0;
for n=2:1:(hs+1)
tophar=tophar+har(k,n);
end
amac(k)=har(k,1)-tophar;
end
topharmaks=0;
for n=3:2:(2*hs+1)
topharmaks=topharmaks+(1-((hs+1)*2)*4/(n*pi));
end
uygtop=0;
for k=((gsay-1)*pop)+1:1:(gsay*pop)
x=(1+((hs+1)*2*4/pi))-topharmaks;
uyg(k)=amac(k)+x;
end

```

```

h=1;
for f=(((gsay-2)*pop)+1):1:(gsay*pop)
ya(h)=uyg(f);
h=h+1;
end
yhh=sort(ya);
n=1;
for r=2*pop:-1:pop+1
yv(n)=yhh(r);
n=n+1;
end
uygtop=0;
for f=1:1:pop
for y=(((gsay-2)*pop)+1):1:(gsay*pop)
if uyg(y)==yv(f)
t(f)=y;
uygtop=uygtop+uyg(y);
end
end
end
ft=1;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
for h=1:1:hs+kac
gata(k,h)=ata(t(ft),h);
if h==hs+kac
ft=ft+1;
end
end
end
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
for h=1:1:hs+kac
ata(k,h)=gata(k,h);
end
end
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
q=1;
gd=1;
if kat~=1
yft=mod(hs,2);
if yft==0
snnb=(hs*3)+1;
else
snnb=(hs*3)+2;
end

for n=1:2:snnb
rtfy=mod(n,3);
if rtfy~=0

```



```

bcd=0;
for ds=1:1:(hs+kac)
tc=mod(ds,2);
if tc==0
bcd=bcd+8*cos(n*ata(k,ds)*pi/180)/(n*pi);
else
bcd=bcd-8*cos(n*ata(k,ds)*pi/180)/(n*pi);
end
end
if q==1
har(k,q)=abs((4/(n*pi))+bcd);
end
if q~=1
har(k,q)=(abs((4/(n*pi))+bcd))*katsay(gd);
gd=gd+1;
end

q=q+1;
end
end
if kat==1
for n=1:2:(2*hs)+1

bcd=0;
for ds=1:1:(hs+kac)
tc=mod(ds,2);
if tc==0
bcd=bcd+8*cos(n*ata(k,ds)*pi/180)/(n*pi);
else
bcd=bcd-8*cos(n*ata(k,ds)*pi/180)/(n*pi);
end
end
if q==1
har(k,q)=abs((4/(n*pi))+bcd);
end
if q~=1
har(k,q)=(abs((4/(n*pi))+bcd))*katsay(gd);
gd=gd+1;
end

q=q+1;
end
end
tophar=0;
for n=2:1:(hs+1)
tophar=tophar+har(k,n);
end

```

```

amac(k)=har(k,1)-tophar;
end
topharmaks=0;
for n=3:2:(2*hs+1)
topharmaks=topharmaks+(1-((hs+1)*2)*4/(n*pi));
end
uygtop=0;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
x=(1+((hs+1)*2*4/pi))-topharmaks;
uyg(k)=amac(k)+x;
uygtop=uygtop+uyg(k);
end
ym=((hs+1)*(pop)*(gsay-1))+1;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
for ds=1:1:(hs+kac)
if(ata(k,ds)-round(ata(k,ds)))>=0
ondalsfk=ata(k,ds)-round(ata(k,ds));
end
if(ata(k,ds)-round(ata(k,ds)))<0
ondalsfk=ata(k,ds)-round(ata(k,ds))-1;
end
for n=1:1:5
ondalsfk=ondalsfk*2;
if ondalsfk<1
tam(ym,n)=round(abs(ondalsfk-0.50000));
end
if ondalsfk>1
tam(ym,n)=round(abs(ondalsfk-0.50000));
ondalsfk=ondalsfk-1;
end
if ondalsfk==1
tam(ym,n)=1;
ondalsfk=ondalsfk-1;
end
end
end
ym=ym+1;
end
end
uygort(gsay)=uygtop/pop;
if uygort(gsay)<=uygort(gsay-1)
bcdvmut=round(rand(1,1)*6)+1;
mutvcdvr=round(rand(1,1)*4)+1;
hym=(gsay-1)*pop*(hs+1);
for k=(((gsay-1)*pop)+1):1:(gsay*pop)

for ds=1:1:(hs+kac)
mutasyon=dec2bin(ata(k,ds),7);
aw(1)=mutasyon(1,1);

```

```

aw(2)=mutasyon(1,2);
aw(3)=mutasyon(1,3);
aw(4)=mutasyon(1,4);
aw(5)=mutasyon(1,5);
aw(6)=mutasyon(1,6);
aw(7)=mutasyon(1,7);
if aw(bcdvmut)==1
mutcev(k,ds)=ata(k,ds)-power(2,(7-bcdvmut));
else
mutcev(k,ds)=ata(k,ds)+power(2,(bcdvmut-1));
end
hym=hym+1;
if tam(hym,mutvcdvr)==1
tam(hym,mutvcdvr)=0;
mutvcdvrcev(k,ds)=tam(hym,1)*0.5+tam(hym,2)*0.25+tam(hym,3)*0.125+tam(hym,4)
*0.0625+tam(hym,5)*0.03125;
else
tam(hym,mutvcdvr)=1;
mutvcdvrcev(k,ds)=tam(hym,1)*0.5+tam(hym,2)*0.25+tam(hym,3)*0.125+tam(hym,4)
*0.0625+tam(hym,5)*0.03125;
end
mutasyonata(k,ds)=mutcev(k,ds)+mutvcdvrcev(k,ds);

end
end
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
q=1;
gd=1;
if kat~=1
yfr=mod(hs,2);
if yfr==0
snnb=(hs*3)+1;
else
snnb=(hs*3)+2;
end

for n=1:2:snnb
rtfy=mod(n,3);
if rtfy==0

b=0;
for ds=1:1:(hs+kac)
tc=mod(ds,2);
if tc==0
b=b+8*cos(n*mutasyonata(k,ds)*pi/180)/(n*pi);
else
b=b-8*cos(n*mutasyonata(k,ds)*pi/180)/(n*pi);
end

```

```

end
if q==1
yhar(k,q)=abs((4/(n*pi))+bcd);
end
if q~=1
yhar(k,q)=(abs((4/(n*pi))+bcd))*katsay(gd);
gd=gd+1;
end

q=q+1;
end
end
end
if kat==1
for n=1:2:(2*hs)+1

bcd=0;
for ds=1:1:(hs+kac)
tc=mod(ds,2);
if tc==0
bcd=bcd+8*cos(n*ata(k,ds)*pi/180)/(n*pi);
else
bcd=bcd-8*cos(n*ata(k,ds)*pi/180)/(n*pi);
end
end
if q==1
yhar(k,q)=abs((4/(n*pi))+bcd);
end
if q~=1
yhar(k,q)=(abs((4/(n*pi))+bcd))*katsay(gd);
gd=gd+1;
end

q=q+1;
end
end
ytophar=0;
for n=2:1:(hs+1)
ytophar=ytophar+yhar(k,n);
end
yamac(k)=yhar(k,1)-ytophar;
end
topharmaks=0;
for n=3:2:(2*hs+1)
topharmaks=topharmaks+(1-((hs+1)*2)*4/(n*pi));
end
uygtop=0;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)

```

```

x=(1+((hs+1)*2*4/pi))-topharmaks;
yuyg(k)=yamac(k)+x;
end
for k=(((gsay-2)*pop)+1):1:(gsay*pop)
for ds=1:1:(hs+kac)
if yuyg(k)>uyg(k)
ata(k,ds)=mutasyonata(k,ds);
else
ata(k,ds)=ata(k,ds);
end
end
end
end
end

if gsay>gensay

for k=(((gsay-1)*pop)+1):1:(gsay*pop)
q=1;
gd=1;
if kat~=1
yft=mod(hs,2);
if yft==0
snnb=(hs*3)+1;
else
snnb=(hs*3)+2;
end

for n=1:2:snnb
rtfy=mod(n,3);
if rtfy~=0

b=0;
for ds=1:1:(hs+kac)
tc=mod(ds,2);
if tc==0
b=b+8*cos(n*ata(k,ds)*pi/180)/(n*pi);
else
b=b-8*cos(n*ata(k,ds)*pi/180)/(n*pi);
end
end
if q==1
har(k,q)=abs((4/(n*pi))+bcd);
end
if q~=1
har(k,q)=(abs((4/(n*pi))+bcd))*katsay(gd);
gd=gd+1;
end

```

```

q=q+1;
end
end
end
if kat==1
for n=1:2:(2*hs)+1

bcd=0;
for ds=1:1:(hs+kac)
tc=mod(ds,2);
if tc==0
bcd=bcd+8*cos(n*ata(k,ds)*pi/180)/(n*pi);
else
bcd=bcd-8*cos(n*ata(k,ds)*pi/180)/(n*pi);
end
end
if q==1
har(k,q)=abs((4/(n*pi))+bcd);
end
if q~1
har(k,q)=(abs((4/(n*pi))+bcd))*katsay(gd);
gd=gd+1;
end

q=q+1;
end
end
tophar=0;
for n=2:1:(hs+1)
tophar=tophar+har(k,n);
end
amac(k)=har(k,1)-tophar;
end
topharmaks=0;
for n=3:2:(2*hs+1)
topharmaks=topharmaks+(1-((hs+1)*2)*4/(n*pi));
end
uygtop=0;
for k=(((gsay-1)*pop)+1):1:(gsay*pop)
x=(1+((hs+1)*2*4/pi))-topharmaks;
uyg(k)=amac(k)+x;
end
uygmaks=max(uyg);
for k=1:1:(gsay*pop)
for ds=1:1:(hs+kac)
if uyg(k)==uygmaks
b(ds)=ata(k,ds);

```

```

end
end
end
sonuc=b
pause(2)
q=1;
for n=1:2:((harg-1)*2)+3
bcd=0;
for ds=1:1:(hs+kac)
tc=mod(ds,2);
if tc==0
bcd=bcd+8*cos(n*b(ds)*pi/180)/(n*pi);
else
bcd=bcd-8*cos(n*b(ds)*pi/180)/(n*pi);
end
end
harmonk(q)=((4/(n*pi))+bcd);
q=q+1;
end
q=1;
figure
hold
for a=1:2:((harg-1)*2)+3
stem(a,harmonk(q))
q=q+1;
end
grid
zoom
xlabel('harmonik numarası')
ylabel('harmonik genliği')
end
end

```