



## PSO, CS ve FA Algoritmalarıyla Ortak Emiterli BJT'li Yükselteç Tasarımı

Ali Fuat BOZ, Murat Erhan ÇİMEN\*

*Sakarya Üniversitesi, Teknoloji Fakültesi, Elektrik-Elektronik Mühendisliği, Sakarya*

Received: 31.10.2016; Accepted: 01.12.2016

**Özet.** BJT'lerin, elektronik devrelerde önemli bir yeri olan yükselteçlerde kullanımı hala popülerliğini korumaktadır. Diğer taraftan, BJT'li yükselteçlerin istenilen karakteristiklerde tasarlanması önemli bir problemdir. Özellikle standart eleman değerleri ile istenilen yükselteç özelliklerini sağlamak için klasik tasarım yöntemlerini kullanmak önemli bir işlem yükü ve aşırı zaman kaybına neden olur. Bunun yanı sıra, elde edilen sonuçlarda, istenilen karakteristiklerden bir veya birkaçının elde edilememesi ihtimali de vardır. Yine, değişen bir parametre için yukarıda sayılan işlem yükünün tekrarlanması da gerekecektir. Bu çalışmada, bu probleme çözüm olarak ortak emiterli bir BJT'li yükseltecin standart eleman değerleri kullanarak tasarımı, Cuckoo Search, Parçacık Sürü Optimizasyonu ve Ateşböceği algoritmaları kullanılarak yapılmıştır. Yine elde edilen sonuçlar karşılaştırmalı olarak sunulmuştur. Bu algoritmalar sayesinde geleneksel yöntemlere göre işlem yükleri ortandan kalkmış ve yükseltecin istenilen karakteristikte çalışması için gereken standart eleman değerleri, belirlenen hata oranının altında seçilmiştir.

**Anahtar Kelimeler:** Yükselteç Tasarımı, PSO, Cuckoo Search, FireFly, Optimizasyon

## Common Emitter BJT Amplifier Design using PSO, CS and FA

**Abstract.** Basic purpose in the amplifier circuits design is to solve and select the standard component values for obtaining the desired circuit characteristics. If the classical design methods are selected then the process will take long time and need a lot of mathematical efforts. As a solution of these problems, in this work Cuckoo Search Algorithm, Partical Swarm Optimization Algorithm and FireFly Algorithm are proposed to determine the standard components values, which are in the market, for Common-Emitter Amplifier. Also, comparisons of the obtained results are given. By those algorithms, required mathematical efforts are reduced substantially. In addition to this, proposed methods provide to choose standard component values to obtain desired circuit characteristics under the specified process error.

**Keywords:** Amplifier Design, PSO, Cuckoo Search, FireFly, Optimization

### 1. GİRİŞ

Elektronik alanında kullanılan BJT'li yükselteçler elektronik devre tasarımında yaygın olarak kullanılmaktadır. Yükselteçlerin tasarımı ve analizi karmaşık ve zahmetlidir. Klasik yöntemlerle çözülmeye çalışıldığında, istenilen sonuçları elde edebilmek için pek çok deneme yapmak gerekir. Bunun nedeni, tasarlanan devrenin istenilen kısıtlar ve karakteristikler için seçilen/tasarlanan eleman değerlerinin, standart eleman değerlerinden farklı olmasıdır.

Klasik yöntem ile tasarımda, elemanlar genellikle ideal ve değerleri sınırsız olarak kabul edilir. Gerçekte ise dirençler E12, E24, E48, E96 ve kondansatörlerde E3, E6, E12, E24 gibi uluslararası belli standartlarda üretilmektedir. Böyle olunca klasik yöntem ile hesaplanan eleman değerleri, standart eleman değerlerine en yakın olanları seçilerek devre gerçekleştirilir. Bu da beraberinde istenilen çözümden sapmalar meydana getirmektedir. Literatürde, sapmaları azaltarak, optimum tasarım yapabilmek için farklı yöntemler önerilmiştir. Bu yöntemler doğadan esinlenilerek zor problemlerin çözülmesi için üretilmiş olan Genetik algoritma(GA), Karınca Kolonisi Algoritması(KKA), Tabu Search Algoritması(TS), Benzetiilmiş Tavlama(SA), Arı Kolonisi Algoritması(BC), Firefly Algoritması(FA), Cuckoo Search(CS), Parçacık Sürü Optimizasyonu(PSO), Differential Evolation(DE) v.b. sezgisel

\* Corresponding author. Email address: muratcimen@sakarya.edu.tr

algoritmalarıdır. Bu algoritmaların kendilerine özgü avantajları olduğu gibi, farklı problemler için farklı başarımları vardır.

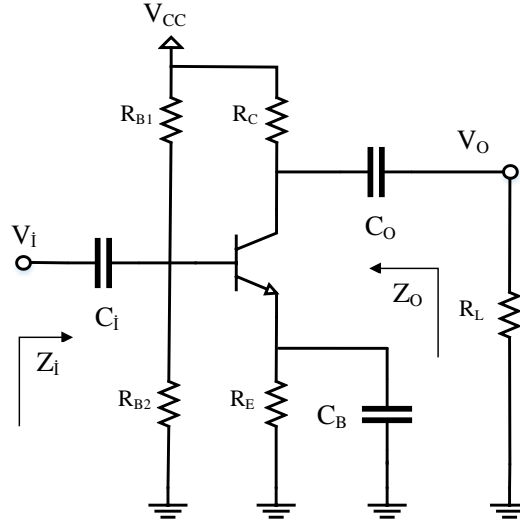
Literatürde David ve ark. GA kullanarak alçak geçiren bir aktif filtre tasarımı gerçekleştirmişlerdir. Önerdikleri yöntemi klasik yöntem ile karşılaştırmışlar ve önerdikleri yöntemin standart devre elemanı seçiminde başarılı olduğunu görmüşlerdir[1]. Bununla birlikte, Turgay ve ark. GA kullanarak Sallen-Key bant geçiren filtre için tasarım yapmışlar, fakat standart eleman değerlerini kullanmamışlardır[2]. Adem Kanalı, KKA kullanarak farklı filtre devreleri için standart eleman değerlerini buldurarak çözüm yapmış ve elde edilen sonuçların, klasik yöntemle göre daha başarılı olduğunu göstermiştir[3]. Mourad ve ark., MOSFET’li yükselteçler için PSO algoritması kullanarak gerilim kazancı ve frekans karakteristiği için analizler yapmışlar ve sonuçlarını Pspice programında gerçekleştirmişlerdir. Kısıtların artması durumunda, PSO’nun diğer algoritmalara nazaran daha iyi sonuç verdiği görülmüştür[4]. Mondal ve Ark., geliştirilmiş parçacık sürü optimizasyonunu yüksek bant geçiren FIR filtresine uygulamışlar ve farklı algoritmalarla karşılaştırarak kendi önerdikleri yöntemin en iyi sonucu verdiğini göstermişlerdir[5]. Ayrıca Gao ve ark., Adaptif IRR filtresini PSO kullanarak FPGA tabanlı bir uygulama olarak yapmışlar ve simülasyonlar ile PSO’nun FPGA’ye başarılı bir şekilde uygulanabileceğini göstermişlerdir[6]. Mehmet A. Belen ve ark. FA kullanarak mikrodalga transistör için performans analizi yapmışlar ve yaptıkları simülasyonlar da başarılı sonuçlar almışlardır[7]. Ayrıca Engin Karaarslan ve ark. ise FA ile haftalık ders programına çözüm getirecek bir yazılım geliştirmişlerdir. Ayrıca FA’nın son zamanlarda farklı alanlara da uygulanabildiğini belirtmişlerdir[8, 9].

Bu çalışmada CS, PSO ve FA algoritmaları Ortak Emiterli BJT’li yükselteç devresi tasarımına uygulanmıştır. Elde edilen sonuçlar ile bu problem için başarımları, çözüm süreleri ve iterasyon sayıları karşılaştırılmıştır.

## 2. ORTAK EMİTERLİ YÜKSELTEÇ DEVRESİ

Bilindiği gibi yükselteçler, girişindeki sinyali akım, genlik veya güç olarak yükselten devrelere verilen genel bir isimdir. İyi bir yükselteçten istenen, yukarıda belirtilen yükseltme işlemi, belirlenen sınırlar içerisinde ve bozulma olmadan gerçekleştirmesidir. Pratikte bu işlemi yapacak farklı devre modelleri kullanılmaktadır. Bu modellerden bazıları yüksek frekans uygulamalarında bazıları ise empedans uygunlaştırma gibi farklı amaçlar ile kullanılmaktadır.

Bu çalışmada, yukarıda sayılan devre çeşitlerinden biri olan ve Şekil 1’de görülen gerilim bölücülü ortak emiterli yükselteç devresi ele alınmıştır. Buradaki amaç, devrede bulunan eleman değerlerinin istenilen giriş empedansı( $Z_i$ ), gerilim kazancı( $A_v$ ) ve maksimum çıkış gerilim( $V_{o(p-p)max}$ ) değerlerine bağlı olarak standart değerlerinin hesaplanmasıdır. Tasarım, devrenin orta frekans bölgesinde çalıştığı farz edilerek yapılmıştır.



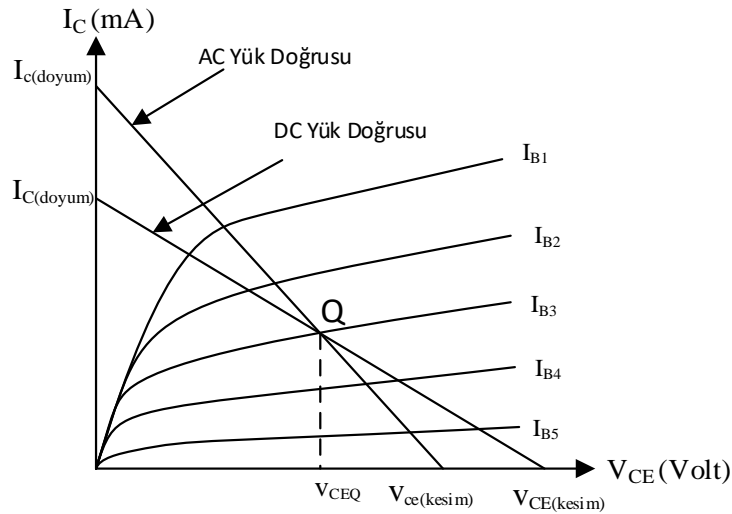
Şekil 1: Ortak Emiterli Yükselteç[10]

### 1.1. DC Analiz, Yük Eğrisi, Çalışma Noktası

Yukarıda belirtilen tasarım kriterlerinden olan ve bir yükseltecin çıkışından bozulma olmadan alınabilecek maksimum tepeden-tepeye gerilim değerini veren  $V_{O(p-p)max}$ , doğrudan devrede kullanılan transistörün çalışma noktasına(Q) bağlı olarak belirlenmektedir. Yük doğrusu üzerinde yanlış bir noktanın seçimi, bu tasarım kriterinin elde edilememesini de beraberinde getirir. Dolayısı ile tasarım açısından çalışma noktası tayini büyük öneme sahiptir. Çalışma noktasını belirleyen devrenin sahip olduğu akım( $I_{CQ}$ ,  $I_{BQ}$ ) ve gerilim( $V_{CEQ}$ ) değerleri ise DC analiz ile bulunabilmektedir.

DC analiz için Şekil 1’de görülen kondansatörler açık devre yapılı ve çevre denklemleri uygulanırsa,  $I_{BQ}$  akımı bulunabilir. Eşitlik-1’den de  $I_{CQ}$  akımı bulunabilir. Bilinen  $I_{CQ}$  akımı ile transistör çıkışından bir çevre denklemi yazıldığında, yük doğrusu denklemi elde edilir. Elde edilen bu denklemlerden  $V_{CEQ}$  gerilimi bulunabilir. Bulunan değerler Şekil 2’de görülen yük doğrusu üzerinde yerine yazılırsa, transistörün Q çalışma noktası belirlenmiş olur[10].

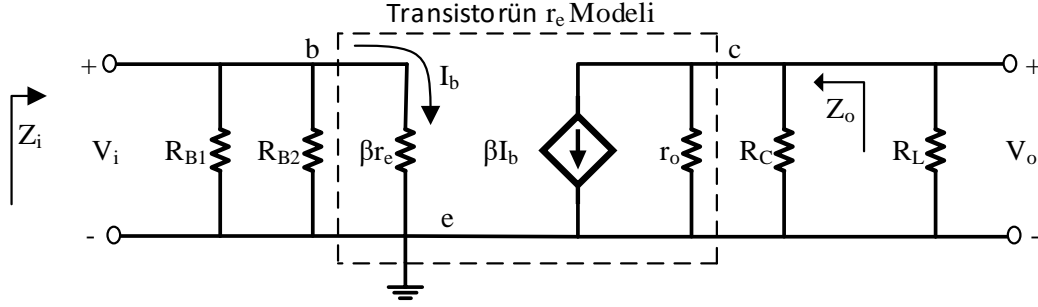
$$I_C = \beta I_B \quad (1)$$



Şekil 2: Transistörün Yük Doğruları ve Q Çalışma Noktası[10].

## 1.2. AC Analiz

Küçük genlikli sinyalleri yükselten devrelere küçük sinyal yükseltici denilmektedir. Küçük sinyal yükselteç devrelerinde giriş empedansı( $Z_i$ ), çıkış empedansı( $Z_o$ ), gerilim( $A_v$ ) ve akım kazanç( $A_i$ ) değerleri AC analiz ile belirlenmektedir. Dolayısı ile istenilen tasarım kriterlerinin sağlanabilmesi için öncelikle bu analizin yapılması gerekmektedir. Bu analiz için öncelikle Şekil 1’de verilen devrenin AC eşdeğeri melez(hibrit) yada  $r_e$  modellerinden biri kullanılarak elde edilir. Bu iki modelden  $r_e$  modeli, dinamik olarak devrenin çalışma noktasına bağlı hesaplandığı için, melez modele göre daha az hatalıdır. Dolayısı ile bu çalışmada da  $r_e$  modeli kullanılmıştır. Şekil 3’de devrenin AC eşdeğer modeli verilmiştir.



Şekil 3. Transistörün AC Analizde  $r_e$  Modeli [10].

Model için gerekli olan  $r_e$  direnç değeri normal oda sıcaklığı için Eşitlik-2 ve 3 kullanılarak yaklaşık olarak elde edilebilir[10].

$$r_e = \frac{26mV}{I_{EQ}} \quad (2)$$

$$I_{EQ} = (\beta + 1)I_{BQ} \quad (3)$$

Tasarım için gerekli olan giriş empedansı( $Z_i$ ), çıkış empedansı( $Z_o$ ), gerilim kazancı( $A_v$ ), akım kazancı( $A_i$ ) ve çıkıştan alınabilecek maksimum tepeden tepeye genlik ( $V_{o(p-p)max}$ ) değerleri AC analiz sonucu elde edilen ve aşağıda verilen (4-8) denklemleri yardımıyla bulunabilir.

$$Z_i = R_{B1} // R_{B2} // \beta r_e \quad (4)$$

$$Z_o = R_C // r_o \quad (5)$$

$$A_v = \frac{V_o}{V_i} = - \frac{R_L // R_C // r_o}{r_e} \quad (6)$$

$$A_i = \frac{i_o}{i_i} = - \frac{Z_i A_v}{R_L} \quad (7)$$

$$V_{o(p-p)max} = \begin{cases} 2 * (V_{CEKsesim} - V_{CEQ}) & \{(V_{CEKsesim} - V_{CEQ}) < V_{CEQ} \text{ ise}\} \\ 2 * V_{CEQ} & \{(V_{CEKsesim} - V_{CEQ}) \geq V_{CEQ} \text{ ise}\} \end{cases} \quad (8)$$

Daha önceden değinildiği gibi, yukarıda verilen denklemler kullanılarak istenilen kısıtlar için klasik veya sayısal yöntemler ile tasarım yapıp, eleman değerleri bulunabilmektedir. Klasik yöntemde işlem yükünün fazla olması ve elde edilen eleman değerlerinin standart değerlerden sapmalar göstermesi dezavantaj olarak görülebilir. Sezgisel yöntemler ise bu problemlere çözüm olarak önerilmektedir. Bu çalışmada da sezgisel yöntemler kullanılmaktadır.

Sezgisel yöntemler ile tasarımda, istenilen eleman değerlerinin bulunması için algoritmaların kullanılacağı bir amaç fonksiyonu tanımlamak gerekmektedir. Bu çalışmada amaç fonksiyonu, istenilen değerler ile algoritmanın bulmuş olduğu değerlerin mutlak farkı şeklinde seçilmiştir. (Eşitlik-9, 10, 11) Ayrıca algoritmalar bu amaç fonksiyonunu yerine getirirken, bazı durumların belirli sınırlar içinde kalması yada bir noktada durması istenebilir. Bu çalışmada, transistörün  $V_{CE}$  gerilimi için aşağıda görülen 3 kısıt eklenmiştir. Bu kısıtlar istenirse arttırılabilir.

Amaç Fonksiyonu;

$$a = \frac{|Z_i - Z_{i\_istenilen}|}{|Z_{i\_istenilen}|} \quad (9)$$

$$b = \frac{|A_v - A_{v\_istenilen}|}{|A_{v\_istenilen}|} \quad (10)$$

$$F = a + b + \text{Kısıtlar}; \quad (11)$$

Kısıtlar;

$$V_{CE} < V_{CC} \quad (12)$$

$$V_{CE} > 0 \quad (13)$$

$$V_{o(p-p)\_istenilen} < V_{o(p-p)} \quad (14)$$

### 3. CUCKOO SEARCH ALGORİTMASI

Cuckoo Search Algoritması(CS), guguk kuşunun çoğalma stratejilerinden faydalanılarak Xin-She ve Suash Deb tarafından geliştirilmiş bir algoritmadır. Esasında kuluçka parazitliğinden(brood parasitism) esinlenilerek geliştirilmiştir[11]. Guguk kuşları başka yuvalara uçarak, oradaki yuvaya yumurtlamaktadır. Yumurtladıkları yuvada/konakta, yumurtadan çıkma olasılığı yüksek olan konağın yumurtalarını da yuvadan atarlar. Asıl yuvanın sahibi geldiğinde yabancı yumurtayı tanırsa yabancı yumurtayı ya aşağı atar ya da yuvayı terk eder. Eğer tanımazsa, o yumurtaların üzerine oturarak kuluçka asalaklığına maruz kalır. Bu durumların kurallaştırılması aşağıdaki gibi olmaktadır[11, 12].

- Her kuş bir tane rastgele yuvaya bir tane yumurta bırakabilir.
- Yuvada kaliteli yumurta varsa bir sonraki nesle aktarılmaktadır.
- Konak yuvanın sahibi, bırakılan yumurtayı belirli bir  $p_a \in (0,1)$  olasılığında tanıyabilmektedir. Eğer yumurtayı tanırsa, yumurtayı yuvadan aşağı atabilir veya yuvayı terk ederek kendisine yeni bir yuva inşa edebilir.

Sezgisel algoritmalar global ve lokal aramayı yaparken, doğası gereği olarak en iyi çözüme yaklaşmaktadır. Fakat diğer olası çözümleri bulabilmesi için keşifte yapması gerekmektedir[13, 14]. Böyle olduğu için, CS global rassal yürüyüş ve lokal rassal yürüyüş yaklaşımlarının bir arada dengelenerek kullanıldığı bir algoritma haline gelmektedir. Bu rassal yürüyüşler de manevra parametresi tarafından kontrol edilmektedir [12, 15].

Yerel rassal yürüyüşü ifade eden denklemde (Eşitlik-15)  $x_k^t$  ve  $x_k^{t+1}$  rassal permütasyonla belirlenmektedir.  $H(u)$  Heaviside adım fonksiyonudur ve  $\epsilon$  parametresi ise düzgün dağılımdan gelen rassal bir sayı iken,  $s$  ise adım uzunluğudur[12, 15].

$$x_i^{t+1} = x_i^t + as \otimes H(p_a - \epsilon) \otimes (x_j^t - x_k^t) \quad (15)$$

Global rassal yürüyüş ise aşağıda kuralları verilen Levy flight ile gerçekleştirilmektedir.

$$x_i^{t+1} = x_i^t + \alpha L(s, \lambda) \quad (16)$$

$$L = \frac{\lambda \Gamma(\lambda) \sin(\frac{\pi \lambda}{2})}{\pi} \frac{1}{s^{1+\lambda}} \quad (17)$$

$$(s \gg s_0 > 0)$$

Levy uçuşu, yeni nesiller üretebilmek için gerçekleştirilir. Burada  $\alpha > 0$  olmak üzere problemin büyüklüğüne bağlı olarak Levy uçuşunu ölçeklendirmektedir[11, 12, 15].

Bu denklemlere bakıldığında, bir sonraki durumun şu anki duruma ve geçiş olasılığına bağlı olduğu görülmektedir. Yeni çözüm kümelerinin o anki en iyi çözümden rassal bir şekilde yeterli seviyede uzaklaşması lazım ki lokal bir noktada takılıp kalmasın. Algoritmanın daha anlaşılır olması için adımları aşağıda verilmiştir.

```

Amaç fonksiyonu  $f(x)$ ,  $x=[x_1 \dots x_d]^T$ 
n adet konak yuvası başlangıç popülasyonu üret
while(Durudurma Kriteri)
Levy Uçuşu ile rassal guguk kuşu al
 $F_i$  maliyetini bul
n yuva içinden rassal yuva seç
if( $F_i > F_j$ )
j'yi yeni çözümle değiştir.
End
En kötü yuvaların  $p_a$  oranı kadarını terk et yenilerini
inşa et
İyi çözümleri tut
Çözümleri sırala ve o anki en iyi çözümü bul
End
    
```

Şekil 4: Cuckoo Search Algoritması(CS) [11]

#### 4. PARÇACIK SÜRÜ OPTİMİZASYONU

Parçacık Sürü Optimizasyonu(PSO) kuş, balık ve havyan sürülerinin çevrelerine adapte olma, zengin yiyecek kaynağı bulabilme ve avcılardan kaçabilme yeteneklerinden esinlenilerek Kennedy ve Eberhart tarafından geliştirilmiş bir algoritmadır. PSO, popülasyon temelli olup, en belirgin özelliği bilgi paylaşmaya yönelik olmasıdır. Popülasyondaki her birey, parçacık olarak da adlandırılmaktadır. Bu bireyler, pozisyon ve hız bilgilerine sahiptirler. Her bir iterasyonda hıza göre pozisyonları güncellenmektedir. Her bir birey(parçacık), bir sonraki pozisyonunu ayarlarken, sürünün en iyi pozisyonuna( $g_{best}$ ) ve şu ana kadar bulunduğu en iyi pozisyona( $p_{best}$ ) göre hareket etmektedir. Parçacığın pozisyonu  $p$ , hızı  $v$  olarak tanımlanır ise bir sonraki iterasyonda pozisyon ve hız bilgisi aşağıdaki denklemlerdeki (Eşitlik 18-19) gibi güncellenir[16, 17].

$$v_{t+1} = wv_t + c_1r_1(p_{best} - p_t) + c_2r_2(g_{best} - p_t) \quad (18)$$

$$p_{t+1} = p_t + v_{t+1} \quad (19)$$

Eşitlik-18 de,  $r_1$  ve  $r_2$  programın üretmiş olduğu ve değeri 0-1 arasında değişen rastgele sayılardır.  $c_1$  ve  $c_2$  öğrenme faktörü olup, genelde 1,8-2 değerini almaktadır.  $w$  ise atalet momenti olup, hızın etkisini zamanla azaltmaya yaramaktadır. Fakat belli bir iterasyondan sonra çok küçülmesi, parçacıkların hareket edememesine neden olmaktadır[16, 18]. PSO algoritması aşağıda verilmiştir.

Begin

```

Amaç fonksiyonu  $f(x)$ ,  $x=[x_1 \dots x_n]^T$ 
Başlangıç pozisyonlarını  $x_i$ , hızlarını  $v_i$  belirle
min  $\{f(x_1) \dots f(x_n)\}$   $g^*$  ata
while(Durdurma Kriteri)
for n parçacık için ve d boyut için
Yeni hızları oluştur ( $v_i^{t+1}$ )
Yeni pozisyonları oluştur ( $x_i^{t+1}$ )
 $f(x_i^{t+1})$ 'leri hesapla ve değerlendir
Her parçacığın en iyi değerini hesapla
end
En iyi çözümü bul  $g^*$  ata
t=t+1 i güncelle
end
xi* ve g* göster

```

End

Şekil 5. Parçacık Sürü Optimizasyonu(PSO) Algoritması[16]

## 5. ATEŞ BÖCEĞİ ALGORİTMASI

Ateş Böceği Algoritması(FA), Ateş böceklerinin parlaklık ve hareket yönlerinden faydalanılarak Yang tarafından geliştirilmiş bir algoritmadır[12, 19]. Bu canlılar, ışığı geceleri avlarını yakalamak veya eşlerini cezbetmek için kullanırlar. Ayrıca bu canlılar, saldırlardan korunmak için esans ve ışık yayarlar. Bununla birlikte, parlaklığı yüksek olan ateşböcekleri doğası gereği diğer ateş böceklerini kendine doğru çekerler. Ateş Böceklerinin davranışları kurallaştırıldığında aşağıdaki sonuçlar çıkar [9, 12].

- Ateş böcekleri cinsiyet ayrımı yapmaksızın birbirlerini etkileyebilirler.
- Ateş böceklerinin çekiciliği, parlaklığı ile orantılıdır. Az parlak olanlar çok parlayanlara doğru hareket ederler.
- Ateş böceğinin parlaklığı, çözülmesi istenen fonksiyonun ürettiği değer ile belirlenir.

Ateş böceklerinin ışık yoğunluğu( $I$ ) değeri Eşitlik-20'de verilmiştir. Bu değer başlangıç ışık yoğunluğuna( $I_0$ ), gammaya( $\gamma$ ) ve uzaklığa( $r$ ) bağlıdır.

$$I = I_0 e^{-\gamma r} \quad (20)$$

## PSO, CS ve FA Algoritmalarıyla

Eşitlik-21’de Ateş böceklerinin çekiciliğini( $\beta$ ) veren denklemdeki (Eşitlik-21) başlangıç değeri( $\beta_0$ ), aslında iki ateşböceğinin arasındaki uzaklığın sıfır olduğu andaki değeri göstermektedir. Çekicilik, uzaklığın karesi, gamma ve başlangıç değerine bağlıdır.

$$\beta = \beta_0 e^{-\gamma r^2} \quad (21)$$

Eşitlik-21’i kullanarak çekiciliği hesaplama bazen zor olabildiğinden, işlemleri kolaylaştırmak için Eşitlik-22 de kullanılabilir[19].

$$\beta = \frac{\beta_0}{1+\gamma r^2} \quad (22)$$

Ateş böceklerinin birbirine olan uzaklıkları önem arz etmektedir, çünkü uzaklıkla hem ışık yoğunluğu hemde çekicilik değişmektedir. Dolayısıyla bu değişimler, ateş böceği için hareketi belirleyecektir. Uzaklık, Eşitlik-23’deki gibi hesaplanabilir.

$$r_{ij} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^d (x_{ik} - x_{jk})^2} \quad (23)$$

Ateş böcekleri daha parlak ve daha çekici olan ateş böceklerine doğru hareket ederler. Bu hareket, Eşitlik-24’deki gibi ifade edilebilir. Eşitlikte bulunan ikinci ifade çekicilik formülünden gelmektedir. Yani, iki ateş böceği arasındaki uzaklık ve çekiciliğin çarpılması ile ifade edilmektedir. Üçüncü ifade de bulunan  $\alpha$  rastlantı parametresi,  $\varepsilon$  ise gaussian dağılımından gelen bir vektörü ifade etmektedir[7, 12].

$$x_i^{t+1} = x_i + \beta_0 e^{-\gamma r_{ij}^2} (x_j - x_i) + \alpha \varepsilon_i^t \quad (24)$$

FA algoritması aşağıda verilmiştir.

```
Amaç Fonksiyonu belirlenir f(x),  $x=(x_1, \dots, x_n)^T$ 
n adet ateş böceği başlangıç popülasyonu
değerleri üret
Işık yoğunluğu(l), f(x) ile belirlenir
Işık emilimi ( $\gamma$ ) belirlenir.
while ( Durdurma kriteri)
  for i=1:n
    for j=1:n
      if ( $l_i < l_j$ )
        Ateş böcekleri i’den j’ye hareket eder
      end if
      uzaklık ve  $[\exp(\gamma r^2)]$  kullanarak çekiciliği değiştir
    end for j
  end for i
  Ateş böcekleri sıraya konular ve global en iyi (g)
  bulunur.
end while
Sonuçlar görüntülenir.
```

Şekil 6. Ateş Böceği Algoritması(FA) [12]



## 6. PROBLEMİN ALGORİTMALARA UYARLANMASI

Her üç algortmada da devrenin istenilen çıkışları verebilmesi için giriş empedansı  $Z_i$ , gerilim kazancı  $A_v$  ve çıkış genliği olan  $V_{o(p-p)max}$  değerleri tanımlanmıştır. Ardından, dirençlerin alabileceği maksimum ve minimum değerleri tanımlanarak algoritma çalıştırılmıştır. Durdurma kriteri ise; bulunan giriş empedansı ( $Z_i$ ), gerilim kazancı  $A_v$ , ve çıkış geriliminin ( $V_{o(p-p)max}$ ) istenilen değerler ile arasındaki farkın %10'un altına düşmesi veya 1000 iterasyon olarak tanımlanmıştır. Aksi takdirde istenilen değerler için algoritma başarısız sayılmıştır.

### 6.1. Problemin Cuckoo Search Algoritmasına Uyarlanması

Tanımlamalar yapıldıktan sonra amaç fonksiyonu belirlenmiş ve ardından popülasyon büyüklüğü 40 olarak seçilmiştir. Adım uzunluğu 0.05,  $\lambda$  parametresi 1.5 ve  $p_a$  parametresi ise 0.25 olarak alınmıştır[12]. Konak yuvaların başlangıç değerleri, direnç değerlerinin minimum ve maximum değerleri arasında rastgele değerler atanarak belirlenmiştir. Popülasyonun büyüklüğü problemde kullanılan dirençler ile doğru orantılıdır. Çünkü her bir direnç değerini bulmak için popülasyona, o direnç değerleri içinde değer atanarak işlemler yapılmaktadır. Ardından her bir yuva için uygunluk değerleri hesaplanmıştır. Yani yuvada bulunan yumurtalar(direnç kümeleri) için. Ardından global arama yapabilmek için Levy uçuşu (Eşitlik 16-17) gerçekleştirilmiştir. Buradan gelen yeni çözümlerin uygunluğu ölçülerek, bulunan çözümlerin kötü olanlarının yerleri değiştirilmiştir. Ardından lokal arama yapabilmek için Eşitlik-15 gerçekleştirilmiştir. Tekrar lokal arama sonucu elde edilen çözümlerin uygunluğu bulunarak, iyi olanları için o direnç değerlerinin yerleri değiştirilmiştir. Sonrasında ise durdurma kriteri kontrol edilerek eğer kriter sağlanıyorsa bir sonraki iterasyona geçilmiştir.

### 6.2. Parçacık Sürü Optimizasyonun Probleme Uyarlanması

CS algoritmasında olduğu gibi, ilk olarak tanımlamalar yapılarak amaç fonksiyonu belirlenmiştir. Ardından parametreler belirlenerek, popülasyon büyüklüğü 40, momentum katsayısı 0.9,  $c_1$  ve  $c_2$  2 olarak belirlenmiştir[17]. Parçacıkların hızlarına ilk olarak 0 değerleri atanmıştır. Ardından parçacıklara başlangıç değeri olarak, belirlenen kısıtlar arasındaki direnç değerlerinden rastgele reel sayılar verilmiştir. Sonra programı bir döngüye alarak, istenen durdurma kriteri sağlanana kadar çalıştırılmıştır. Döngü içinde hızlar hesaplanıp, ardından bu hızlar ile parçacıklar hareket ettirilme ve bulunan parçacıkların yeni konumlarının uygunlukları belirlenmektedir. İyi olanların yerleri değiştirilerek  $p_{best}$  ve  $g_{best}$  değerleri belirlenmektedir. Her bir parçacık için hız değerleri hesaplanarak döngü tekrar başa gelmekte ve işlemler tekrar edilmektedir.

### 6.3. Problemin Ateş Böceği Algoritmasına Uyarlanması

Diğer algoritmalarda olduğu gibi ilk olarak tanımlamalar yapılmıştır. Amaç fonksiyonu belirlenerek diğer algoritmalarla aynı olması için popülasyon büyüklüğü(n) 40 olarak belirlenmiştir. Ardından kontrol parametreleri belirlenerek, rastgele sayının parametresi olan alfa( $\alpha$ ) 0.2 olarak seçilmiştir. Çekiciliğin başlangıç değeri( $\beta_0$ ) 0.8 olarak seçilmiştir. Ardından dirençlerin sınırları belirlenerek, sınır değerleri dahilinde rastgele değerler atanmıştır. Işık yoğunluğu( $\gamma$ ) bu problem için 0.9 olarak belirlenmiştir. Sonra döngü oluşturulmuştur. Durdurma kriteri olarak seçilen hatanın %10'un altına düşmesi için 1000 iterasyon sağlanana kadar döngü çalıştırılmıştır. Döngü içinde ilk olarak ateş böceklerinin birbirine olan çekiciliğine bakılmış daha parlak ve çekici olanlara doğru diğerleri hareket ettirilmiştir. Hareket sonrası tekrardan çekicilik ve ışık yoğunluğu hesaplanmıştır. Tüm ateş böcekleri hareket ettirildikten sonra, sonuçlar belirlenen amaç fonksiyonuna göre sıraya konularak, o iterasyon için en iyi sonuç belirlenmiştir. Akabinde durdurma kriteri kontrol edilip, şart sağlanıyorsa döngüye devam edilmiş, sağlanmıyorsa döngü kırılarak sonuçlar görüntülenmiştir.

## 7. SİMÜLASYON SONUÇLARI

Bu çalışmada, tasarımı yapılmak istenilen ortak emiterli yükselteç devresinden istenilenler gerilim kazancı( $A_v$ ), giriş empedansı( $Z_i$ ) ve maksimum çıkış gerilimi değeri( $V_{o(p-p)max}$ ) olarak belirlenmiştir. Kısıt olarak ise standart direnç değerlerinin kullanılması seçilmiştir. Gerekli olan standart direnç değerlerinin hesaplanması için geleneksel yöntemden ziyade PSO, CS ve FA algoritmaları kullanılmış ve elde edilen sonuçlar kıyaslanmıştır. Bu yükselteç devresi için 20 farklı çalışma noktası için analizler yapılmıştır. Bu sonuçlar Intel(R) Core(TM) i7-6700 HQ CPU @ 2.60 Ghz, 64 Bit, 8GB RAM'e sahip bir bilgisayardan elde edilmiştir. Yapılan çalışma Matlab 2013a programı kullanılarak gerçekleştirilmiştir.

**Tablo 1.** CS, PSO ve FA için Örnek Uygulamalar

	Devreden İstenilen Karakteristikler			Algoritmanın E24 Standardına Göre Bulduğu Direnç Değerleri				Algoritmanın Bulduğu Sonuçlar			Yüzde Hata
	$Z_i(\Omega)$	$A_{v_n}$	$V_{o(p-p)}(V)$	$R_{B1}(k\Omega)$	$R_{B2}(k\Omega)$	$R_C(\Omega)$	$R_E(k\Omega)$	$Z_i(\Omega)$	$A_{v_n}$	$V_{o(p-p)}(V)$	
CS	2000	-10	3,6	24	82	160	1.3	1983,502	-10,158	5,132	%2.4
PSO	2000	-10	3,6	20	10	150	1.8	2015,376	-9,722	6,989	%3.54
FA	2000	-10	3,6	130	47	120	0.910	2123,100	-10,06	4,706	%6.75
CS	2700	-80	5	33	10	4700	1.5	2553,055	-78,339	7,209	%7.51
PSO	2700	-80	5	27	10	5600	2	2650,947	-75,873	5,085	%7,02
FA	2700	-80	5	39	10	5100	1.3	2654.505	-77,180	7,354	%5.21
CS	3000	-10	3	27	5.6	470	2	2898,429	10,0229	3,212	%3.61
PSO	3000	-10	3	20	6.8	390	3	2889,249	-9,8757	5,163	%4.93
FA	3000	-10	3	27	5.6	470	2	2894,400	-10,022	3,212	%3.74
CS	5000	-20	2	56	7.5	9100	2.4	4810,216	-20,091	2,460	%4.25
PSO	5000	-20	2	62	7.5	5100	1.8	4719,800	19,184	2,165	%9.68
FA	5000	-20	2	390	91	750	1.3	5136,300	20,263	3,502	%4.04

Tablo 1'de verilen deney sonuçlarından seçilen bir kaç örnek uygulamanın sonuçlarına bakıldığında,  $Z_i$  ve  $A_{v_n}$ 'ye bağlı olan amaç fonksiyonu(Eşitlik 9-10), %10'un altında değerler vermiş ve kısıt olarak seçilen  $V_{o(p-p)}$ 'nin(Eşitlik 11) istenen değeri sağladığı görülmektedir. Yapılan uygulamalarda  $V_{CC}$  gerilimi 12 Volt,  $\beta$  değeri 200 ve  $R_L$  yük direnci 2.2 k $\Omega$  olarak alınmıştır.

**Tablo 2.** CS, PSO ve FA algoritmalarının sonuçlarının istenilen sonuçlara benzerlikleri.

	$Z_i$	$A_{v_n}$
<b>CS</b>	0,993	0,999
<b>PSO</b>	0,99	0,999
<b>FA</b>	0,992	0,999

Tablo 2'de Ortak Emiterli Yükselteç devresinin CS, PSO ve FA ile çözümlerinde herbirinin 20 farklı çalışma noktası için yapılan deneylerde elde edilen çıkışların, istenilen çıkışlarla algoritmaların bulmuş olduğu çıkışlar arasında anlamlı bir ilişkinin olduğu ve sonuçların ne derece benzer olduğu korelasyon katsayısı ile ifade edilmektedir.

## BOZ, ÇİMEN

Belirlenen hata oranlarına bakıldığında, aslında tüm algoritmaların istenilen tolerans değerleri içerisinde sonuçlar ürettiği görülmektedir. Fakat benzerliğine bakıldığında, 0,993 ile giriş empedansına( $Z_i$ ) en yakın sonucu bulan CS olmuştur. Çıkış kazancına( $A_{Vn}$ ) bakıldığında ise tüm algoritmalar bu 20 farklı örnek içinde benzer sonucu vermişlerdir.  $V_{o(p-p)}$ , tüm algoritmalarda belirlenen şart olan “*istenen çıkış geriliminin üstünde*” (Eşitlik 14) değerlerini sağlamıştır. Eğer sağlamamış olsaydı, kısıtlardan dolayı algoritmaların başarıma kriterini sağlamamış olurdu.

**Tablo 3.** CS, PSO ve FA Hesaplama Süreleri (20 Örnek için).

Süre (sn)	CS	PSO	FA
Ortalama	1,007	0,601	0,515
Standart Sapma	0,269	2,157	0,829
Toplam Süre(sn)	20,130	12,020	10,300

Hesaplama sürelerinin sonuçları Tablo 3’de verilmiştir. Sonuçlara bakıldığında Ateş Böceği algoritması hem ortalama hem de 20 örneğin toplamı 10,3 sn olduğundan, diğer algoritmalara göre daha hızlı işlem yapmıştır. CS ise yaptığı işlemler nedeniyle 20,13 sn ile diğer algoritmalara nazaran yavaş kalmıştır. Fakat standart sapması 0,269 olduğu için, diğer algoritmalara göre kararlı bir yapısı olduğu görülmektedir. Bu bakımdan PSO, CS’den daha hızlı olmasına rağmen, standart sapması yüksek olduğu için 20 örnek içinde bazılarını daha uzun sürede bulmuştur. Bunun nedeni başlangıçta değişkenlere rasgele atanan değerlerden kaynaklanmaktadır.

**Tablo 4.** CS, PSO ve FA İterasyon Sonuçları (20 Örnek için).

İterasyon	CS	PSO	FA
Ortalama	19,950	188,350	60,600
Standart Sapma	25,370	675,000	130,000
Toplamı	399	3767	1212

Algoritmaların iterasyon sayıları Tablo 4’de verilmiştir. Sonuçlara bakıldığında 399 iterasyon yaparak en az iterasyon ile sonuç bulan algoritma CS iken, 1212 iterasyon ile en fazla iterasyon yapan algoritma PSO algoritması olmuştur. Bunun nedeni algoritmaların yapısından kaynaklanmaktadır. CS bir adımda çok fazla işlem yaparken, PSO ve FA algoritmaları bu kadar çok işlem yapmamaktadır.

Bu problem için, giriş empedans( $Z_i$ ) ve gerilim kazancı( $A_{Vn}$ ) için en iyi sonuçları CS bulmasına rağmen, süre bakımından FA’nın daha avantajlı olduğu görülmüştür.

## 8. SONUÇ

Bu çalışmada Ortak Emiterli bir yükselteç devresinin giriş empedansının( $Z_i$ ) ve gerilim kazancının( $A_{Vn}$ ) istenilen değerlerde olması ve maksimum çıkış geriliminin( $V_{o(p-p)}$ ) istenilen değerden yüksek olmasını sağlayacak standart elemanların seçimi için CS, PSO ve FA algoritmalarına dayalı bir tasarım yaklaşımı önerilmiştir. Bu problem için, CS’nin doğruluk ve kararlılık bakımından iyi sonuçlar verdiği, fakat süre bakımından FA’nın daha avantajlı olduğu görülmüştür. Sonuçların, belirlenen %10 hata oranının altında olması, başarı bakımından PSO, CS ve FA’nın istenilen sonuçları sağladığı görülmüştür.

**KAYNAKLAR**

- [1]. David H. Horrocks and Mark C. Spittle, "Component Value Selection For Active Filters Using Genetic Algorithms," *IEE/IEEE Workshop on Natural Algorithms in Signal Processing*, vol. 1, pp. 13/1-13/6, 1993.
- [2]. T. Kaya and M. C. İnce, "Genetik Algoritmaların Aktif Filtrelerde Kullanımı," *ELECO-2008 Elektrik-Elektronik-Bilgisayar Mühendisliği Sempozyumu, Bursa, Turkey*, pp. 512-515, 26-30 Kasım 2008.
- [3]. A. Kalinli, "Karınca Koloni Algoritması Kullanarak Aktif Filtre Tasarımı," *Elektrik-Elektronik-Bilgisayar Mühendisliği 10. Ulusal Kongresi*, pp. 267-270, 2003.
- [4]. M. Fakhfakh, Y. Cooren, A. Sallem, M. Loulou, and P. Siarry, "Analog circuit design optimization through the particle swarm optimization technique," *Analog Integrated Circuits and Signal Processing*, vol. 63, pp. 71-82, 2009.
- [5]. S. Mondal, Dishari Chakraborty, R. Kar, D. Mandal, and S. P. Ghoshal, "Novel Particle Swarm Optimization for High Pass FIR Filter Design."
- [6]. Z. Gao, X. Zeng, J. Wang, and J. Liu, "FPGA implementation of adaptive IIR filters with particle swarm optimization algorithm."
- [7]. M. A. Belen, M. Alıcı, A. Çor, and F. Güneş, "Ateşböceği Algoritması ile Mikrodalga Transistör Performansının Karakterizasyonu," *ELECO-2014 Elektrik-Elektronik-Bilgisayar ve Biyomedikal Mühendisliği Sempozyumu*, pp. 491-494, 2014.
- [8]. X. S. Yang and X. He, "Firefly algorithm: recent advances and applications," *International Journal of Swarm Intelligence*, vol. 1, p. 36, 2013.
- [9]. E. KARAARSLAN and K. ZENGİN, "Ateş Böceği Algoritması ile Haftalık Ders Programı Hazırlama," *EEB 2016 Elektrik-Elektronik ve Bilgisayar Sempozyumu*, 11-13 Mayıs 2016.
- [10]. R. BOYLESTAD and L. NASHELSKY, "ELECTRONIC DEVICES AND CIRCUIT THEORY."
- [11]. X.-S. Yang and S. Deb, "Cuckoo Search via Lévy Flights," *Nature & Biologically Inspired Computing*, pp. 210 - 214, 9-11 Dec. 2009.
- [12]. X.-S. Yang, "Nature-Inspired Optimization Algorithms, 1st Edition," *Elsevier*, 20 Feb 2014.
- [13]. Y. X-S, "Harmony Search as a metaheuristic algorithm Music-inspired harmony search: theory and application," 2009.
- [14]. A. H. Gandomi, X.-S. Yang, and A. H. Alavi, "Cuckoo Search algorithm: a metaheuristic approach to solve structural optimization problems," *Springer-Verlag*, vol. 29, p. 18, 2013-01.
- [15]. K. Karagül, "Guguk Kuşu Algoritması: Bir Plastik Atık Toplama Uygulaması," *15th International Symposium on Econometrics, Operations Research and Statistic, Isparta, Turkey*, vol. 15, pp. 775-784, 22-25 May 2014.
- [16]. A. T. F. E. S. O. İnce, "Sensör Yerleştirme Probleminin Parçacık Sürü Optimizasyonu ile Çözümü."
- [17]. Y. S. a. R. Eberhart, "A Modified Particle Swarm Optimizer," 1998.
- [18]. P. Civicioglu and E. Besdok, "A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms," *Artificial Intelligence Review*, vol. 39, pp. 315-346, 2011.
- [19]. X.-S. Yang, "Firefly Algorithms for Multimodal Optimization," 7 Mar 2010.