



*T.C*

***SAKARYA ÜNİVERSİTESİ***  
**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

# **BSM 310 – YAPAY ZEKA**

**Grup üyeleri:**

**B1712.10011 BİLGENUR KARA**  
**B1712.10065 BÜŞRA DOĞAN**  
**B1712.10077 GÜL AYIK**  
**B1712.10029 OĞUZ CAN KEÇİLİ**  
**B1712.10107 PINAR ELİGÜL**

**Sakarya**  
**2020**

# ATEŞ BÖCEĞİ ALGORİTMASI

## Özet

*Doğa, birçok meta-sezgisel algoritmanın tanıtımı için bir ilham kaynağı olmuştur. Bunlardan bir tanesi de Ateş Böceği algoritmasıdır. Ateş Böceği algoritması ateş böceklerinin ışık yayma ve ışık kaynağına doğru hareketlerinden esinlenerek çözüm üretmeyi amaçlayan bir algoritmadır. Diğer bazı sezgisel algoritmalara kıyasla gerçek problemlere uygulanması kolaydır.*

2008 Yılında Xin-She Yang tarafından Cambridge Üniversitesi'nde geliştirilmiş sürü tabanlı Meta-Sezgisel Optimizasyon Algoritmasıdır. Ateş böceklerinin parlaklığa duyarlı sosyal davranışlarını ele alarak geliştirilmiştir.

Ateş böcekleri haberleşmek için yaydıkları ışıkları kullanırlar. Genellikle belli bir düzeni bulunan, kısa süreli yanıp sönen bir ışık yayarlar. Işığın yayılımının arkasında biyoluminesans (bazı canlı organizmaların gerçekleştirdiği, kimyasal reaksiyonlar sırasında kimyasal enerjinin ışık enerjisine dönüştürülmesi ile ışık üretilmesi ve yayılması olayına verilen isim) denen biyolojik bir süreç vardır. Işıkla iletişim kendilerine eş çekmek ve avcılara karşı uyarmak için kullanılır. Işığın örüntüsüne göre bu örüntüye uyan bir ateş böceği ya aynı örüntüyü tekrarlayarak ya da kendisine özgü örüntüyü uygulayarak cevap verir.

Algoritma ateş böceklerini cinsiyet ayrımı olmadan ele alır. Yani herhangi bir ateş böceği diğer bütün ateş böceklerinden etkilenebilir.

Daha parlak olan ateş böcekleri daha çekicidir. Daha az parlak olan ateş böcekleri çekici olan ateş böceklerine doğru yönelir. Parlaklık etkisi uzaklık arttıkça azalacağı için daha uzaktaki ateş böcekleri uzaktaki parlak ateş böceklerinden daha az etkilenir. Ayrıca bir ateş böceği eğer kendinden daha parlak bir ateş böceği bulamazsa rastgele hareket gerçekleştirir.

Ateş böceklerinin en parlak ateş böceği arayışı bu algoritmadaki temel prensiptir. Ateş böceklerinin bu içgüdüsel davranışını ele alarak diğer seçenekler içinde en uygun ve iyiyi arayışımız onların bu hareketlerini taklit ederek formülize etmemizi ve gerçek hayatta yararlanmamızı sağlamıştır.

Esinlenen Davranışlar	Algoritmadaki uygulama
1. Ateş böceklerinin rastgele konumu belirlenir. Ardından parlaklıkları hesaplanır.	1. Rastgele çözüm kümeleri oluşturulur. Her çözüm kümesi parametre sayısı kadar eleman içerir. Oluşturulan çözüm kümelerinin uygunluk değerleri bulunur.
2. Ateş böcekleri kendilerinden daha parlak olan ateş böceklerine doğru hareket ederler.	2. Tüm çözüm kümelerine sırasıyla iyileştirilme formülü uygulanır.
3. Belirlenen tekrar sayısına ulaşıncaya kadar ikinci aşamaya dönülür.	3. Maksimum iterasyon sayısına ulaşıncaya kadar ikinci aşamaya geri dönülür.

## Algoritmada Tanımlar

**Problem :** Algoritmada çözüm aranılan problemdir ve genellikle bir matematiksel formülle belirtilir.

**Uygunluk fonksiyonu :** Algoritmada problem için bulunulan çözüm kümelerinin iyilik derecesinin bulunmasına yarayan formül.

**Popülasyon Boyutu:** Popülasyondaki ateş böceği sayısının toplamıdır. Yani bir iterasyondaki çözüm kümesi sayısı.

**Parametre Boyutu:** Uygunluk fonksiyonunda bilinmeyen değişkenlerin sayısını ifade eder.

**Parametre Aralığı:** Parametrelerin alabileceği en büyük ve en küçük değerlerdir.

**Maksimum İterasyon:** Algoritmanın çalışacağı maksimum döngü sayısını belirtir. Her bir iterasyonda tüm işlemler tekrar eder.

**$\alpha$ (alpha):** Rastlantı değişkeni. 0 ve 1 arasında alınabilir.

**$\gamma$ (gama):** Sabit emilim katsayısı 0.01 ve 100 arasında alınabilir.

## • Çekicilik ve Işık Yoğunluğundaki Değişim

Ateş Böceği Algoritmasında iki önemli durum vardır. Bunlar; ışık yoğunluğunu değişimi ve çekiciliğin formülize edilmesidir. Çekicilik,  $i$  ve  $j$  konumundaki ateşböcekleri arasındaki mesafeye göre değişiklik gösterir. Ek olarak ters kare kanununu baz alarak, kaynaktan uzaklaştıkça ışık yoğunluğu düşer ve ışık ortam tarafından emilir.

$$I = I_0 e^{-\gamma r}$$

- $I_0, r=0$  konumundaki ilk ışık yoğunluğudur.
- $\gamma$ , sabit emilim katsayısıdır.

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

- $\beta_0, r=0$  konumundaki çekiciliktir.
- Hesaplanması daha kolay olan  $1/(1+r^2)$  formatı kullanılır

$$\beta = \frac{\beta_0}{1 + \gamma r^2}$$

$$\beta(r) = \beta_0 e^{(-\gamma r^m)}, \quad m \geq 1$$

- Monoton azalan genelleştirilmiş bir denklem elde edilir.

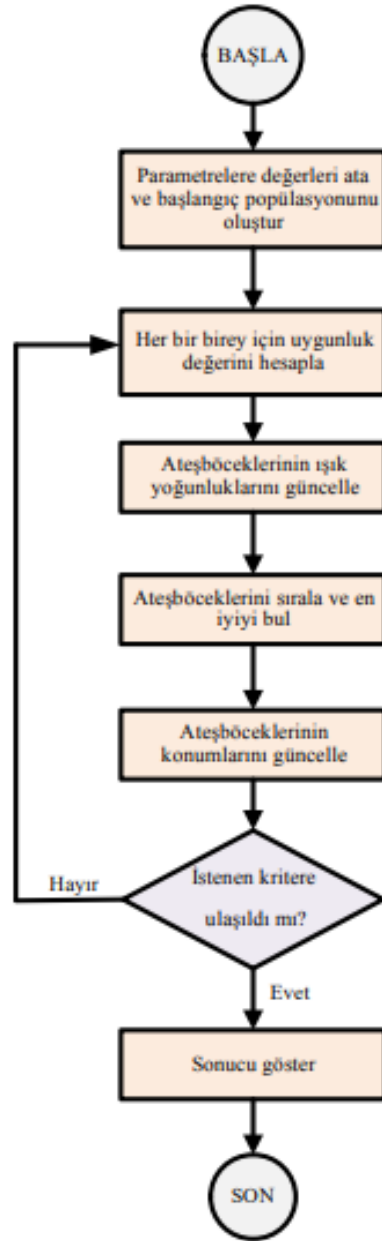
$$r_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

- Herhangi iki ateşböceği arasındaki mesafe( $x_i, x_j$ ) iki boyutlu uzayda bulunur.

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

- $i$  ateş böceğinin daha çekici olan ateş böceklerine hareketi,  $j$  ateş böceği daha çekici olan, yukarıdaki gibi tanımlanır.
- Bu denklemde ikinci ifade çekicilik formülünden gelmektedir. Üçüncü ifade ise; rastlantısal olarak gelir,  $\alpha$  rastlantı parametresi ve  $\varepsilon_i$  „de rastgele sayılar ile çizilmiş bir Gauss dağılım vektörüdür. Örnek olarak, en basit anlamda  $\varepsilon_i$  rand-1/2 ile değişmesi. Burada rand, rastgele sayı üreticidir ve  $[0,1]$  aralığındadır.

## • Ateş Böceği Algoritması Akış Diyagramı



Ateş böceği algoritması oldukça hızlı ve efektif bir algoritmadır. Ateş böceklerini olası çözüm adayları olarak düşünür. Her seferinde gerçek çözüme yakınlığa göre aday çözümler puanlanır bu da ateş böceğinin parlaklığını belirleyen etken olur. Her ateş böceği hem kendinden daha parlak olana doğru hemde rastgele hareket eder. Böylece temsil ettiği çözüm geliştirilir ve daha iyi bir çözüme yaklaşan parlaklığı artmış olur. Bu döngü içerisinde, belli sayıda bir iterasyondan sonra bütün ateş böcekleri optimum çözümlerde toplanmış olur.

# Örnek Çalışma 1: Ateş Böceği Algoritması ile Mikrodalga Transistör Performans Karakterizasyonu

## 1.Mikrodalga Transistör Karakteristikleri:

Bir mikrodalga transistörün, sistem karakteristiklerini optimize edebilmek için birçok nümerik yöntem kullanılmaktadır. Bu yöntemlerin çoğu bir çalışma bandı boyunca (GT) üzerinde yoğunlaşırken, giriş uyumsuzluğu ( $V_i$ ), gürültü faktörü (F), çıkış uyumsuzluğu ( $V_{out}$ ), gibi diğer performans ölçü kriterleri hesaba alınmaz. Diğer taraftan, „optimizasyon“ süreci, sistemin performans model parametrelerine, yüksek dereceli nonlinearer bağımlı bir süreçtir. Aslında, bir optimizasyon sürecinde, aktif eleman fiziksel limitleri ve/ya da NF,  $V_i$ , GT performans ölçüleri ara ilişkileri bilinmeksizin, istenilen F,  $V_i$ , GT değerleri, „referans“ değer takımı olarak verilebilir. Fakat bu (NFreq,  $V_i$ req, GTreq) üçlünün uyumsuzluğu nedeni ile optimizasyon süreci, sık sık başarısızlıkla sonuçlanmaktadır. Bu bitirme çalışmasındaki asıl amaç, ateşböceği algoritması ile belirli frekanslarda transistörün; NF,  $V_i$ ,  $V_{out}$ , GT performans ölçülerinin birlikte verimli çalıştığını araştırmaktır.

## 2.Transistör Karakteristiklerinin Ateş Böceği Algoritmasıyla Optimizasyonu:

Bu bölümde transistör 15,5-17,5 GHz frekanslarında incelenmiştir. Gerekli formüller ve istenilen çıkışlar belirlenip programa girilmiştir ve optimizasyon için ateşböceği algoritması ile kodlanmıştır. Bu işlem adımlarının yer aldığı akış diyagramı şekil 1 de verilmiştir.

Ateş Böceği Algoritması
Amaç Fonksiyonu $f(x)$ , $x=(x_1, \dots, x_n)^T$ . N ateş böceğinin başlangıç popülasyonunun oluşturulması $x_i (i=1,2, \dots, n)$ . Işık yoğunluğu $I_i$ , $f(x_i)$ den belirlenir. Işık emilim katsayısı tanımlanır ( $\gamma$ ). <b>while</b> ( $t < \max$ jenerasyon), <b>for</b> $i=1:n$ (tüm ateş böcekleri için) <b>for</b> $j=1:n$ (tüm ateş böcekleri için) (iç halka) <b>if</b> ( $I_i < I_j$ ) (Ateş böcekleri $i$ den $j$ ye doğru hareket ederler.) <b>endif</b> Çekicilik çeşitliliği, uzaklıkla ( $r$ ) $\exp(-\gamma r^2)$ ile değişir. Yeni sonuçlar değerlendirilir ve ışık yoğunluğu güncellenir. <b>endfor j</b> <b>endifor i</b> Ateş böcekleri sıraya konular ve güncel global en iyi bulunur( $g^*$ ) Son işlem sonrası sonuçlar ve simülasyon çıkışı

Şekil 1: Ateş Böceği algoritmasının pseudo kod halinde yazılması

## 3.Hedef Fonksiyonu:

Bu bölümde yer alan sonuçlar 1.5 VSWR altında maksimum kazanç sağlamak için aşağıdaki frekanslar için optimize edilmiştir.

$$\text{Hata} = |V_i - 1.5| + |V_{out}| + e^{-G_{Idb}}$$

Çizelge 1: Frekansa göre Ateşböceği Algoritması optimizasyon sonuçları

Frekans (GHz)	Hata	Reel (ZS)	Im (ZS)	Reel (ZL)	Im (ZL)	GT	NF	$V_i$	$V_{out}$
15,5	0,577	26,28	-8,93	31,202	-9,67	11,613	0,54	1,55	1,51
16	0,573	28,468	-13,6	34,638	-6,87	11,541	0,57	1,53	1,53
16,5	0,543	30,773	-18,2	33,124	-7,61	11,495	0,59	1,55	1,47
17	0,615	33,108	-21,7	34,742	-9,60	11,367	0,62	1,56	1,52
17,5	0,659	35,273	-24,2	36,412	-12,1	11,229	0,64	1,56	1,57

Burada kazancın maksimum olması ön planda tutulup,  $V_0$ ,  $V_i$  kazanca göre optimize edilmiştir.

Çizelge 2: Maksimum kazanç için frekansa göre optimizasyon sonuçları

Frekans (GHz)	Hata	Reel (ZS)	Im (ZS)	Reel (ZL)	Im (ZL)	GT	NF	$V_i$	$V_{out}$
15,5	1,363	26,2	-8,92	32,7	-8,2	11,61	0,54	1,54	1,51
16	1,119	28,4	-13,6	34,2	-7,99	11,53	0,57	1,52	1,54
16,5	1,189	30,7	-18,2	34,4	-7,30	11,46	0,59	1,51	1,52
17	1,315	33,1	-21,6	35,2	-9,65	11,35	0,62	1,55	1,54
17,5	1,364	35,2	-24,2	36,9	-11,8	11,21	0,64	1,55	1,58

Çizelge 1 ve 2 de ki sonuçlar karşılaştırıldığında çok fazla fark olmasa da maksimum kazanç elde edildiği ve sonuçların buna göre nasıl değiştiği gözlemlenebilir. Burada,  $V_i$  değeri artarken  $V_{out}$  değerinin azalmaktadır. Optimizasyonda sadece kazanç önemsendiği ve giriş-çıkış değerleri belli aralıklarda alındığı halde hata payı artmaktadır. Elde edilen sonuçlar AWR ortamında NE3514S02 giriş ve çıkış empedans değerleri olarak verilerek sonuçların doğruluğu sağlanmıştır.

Çizelge 3: AWR „de 15,5-17,5 GHz için alınan sonuçlar:

Frekans(Ghz)	Kazanç(dB)	Gürültü	$V_{in}$	$V_{out}$
15,5	11,61	0,53	1,54	1,51
16	11,54	0,57	1,52	1,54
16,5	11,47	0,59	1,51	1,52
17	11,35	0,62	1,55	1,54
17,5	11,22	0,64	1,55	1,58

**4.Sonuçlar:** Yapılan çalışmalar ve benzetimler sonucunda ateşböceklerinin yaşam tarzından esinlenilerek hazırlanan algoritmanın verilen denklemlerin istenilen çözüm aralıklarında çok düşük hata payında hatalarla çözümler sağladığı ve bu çözümlerle de istenilen mikrodalga devrelerin en uygun şekilde optimize edilebileceği görülmüştür. Çok yeni olmasına rağmen metasezgisel bir algoritma olan ateşböceği algoritmasının diğer doğa-ilham kaynaklı algoritmalara daha iyi sonuçlar elde edebileceği simülasyon sonuçları ile gözlenmiştir.

## Örnek Çalışma 2: Uzay Kafes Yapılarının Ateş Böceği Algoritması Yöntemiyle Optimizasyonu

Uzay kafes yapıların minimum ağırlıklı tasarımı için geliştirilen FFA yöntemi şu adımlardan oluşmaktadır:

1. Ateşböceği sayısı ( $nf$ ),  $r=0$  için çekicilik ( $\beta\theta$ ) değeri, ışık emme katsayısı ( $\gamma$ ), rasgele seçim parametresi ( $\alpha$ ) ve maksimum iterasyon sayısı (maksimum kafes yapı analiz sayısı= $mit$ ) parametrelerini atanır. İterasyon değerini sıfırlanır  $it=0$ .
2. Her biri bir kafes yapı tasarımı temsil eden ateşböceklerinin oluşturduğu popülasyonu rasgele oluştur. Bu popülasyondaki her bir kafes yapıya ait bir tasarım değişkeni, maksimum ( $x_{k,maks}$ ) ve minimum ( $x_{k,min}$ ) sınır değerleri arasında aşağıdaki şekilde rasgele üretilir:

$$x_k^j = x_{k,min}^j + rand(0,1)(x_{k,maks}^j - x_{k,min}^j)$$

$$k = 1,2, \dots, ng \quad j = 1,2, \dots, nf$$

Burada  $x_{k,min}$  ve  $x_{k,maks}$  sırasıyla tasarım değişkeninin alabileceği en küçük ve en büyük değerler,  $rand(0,1)$  ise  $[0,1]$  aralığında rasgele üretilen bir reel sayıdır. Bu şekilde eleman grup sayısı ( $ng$ ) kadar tasarım değişkeni üretilerek bir kafes yapı tasarımı elde edilir. Ateşböceği sayısı (kafes yapı sayısı) kadar bu işlem tekrarlanarak tasarım popülasyonu oluşturulur.

3. Popülasyondaki her kafes yapı tasarımı için analiz yapılarak düğüm deplasman ve eleman eksenel kuvvetleri elde edilir. Her kafes yapının  $W(X^1)$ ,  $W(X^2)$ , ...,  $W(X^{nf})$  ile gösterilen modifiye edilmiş amaç fonksiyonu değeri hesaplanır. Tasarım popülasyonundaki kafes yapıların modifiye edilmiş amaç fonksiyonu değerleri küçükten büyüğe sıralanır ( $W(X^1) < W(X^2) < \dots < W(X^{nf})$ ). Buna göre popülasyonun ilk sırasındaki tasarım en iyi tasarım  $W(X^1) = W(X^{en-iyi})$ , son sıradaki tasarım en kötü tasarım olarak  $W(X^{nf}) = W(X^{en-kötü})$  tanımlanır. Bu şekilde düzenlenen tasarımları içeren popülasyon aşağıda verilen matris formunda gösterilir:

$$X^j = \begin{bmatrix} x_1^1 & \dots & x_{ng}^1 \\ \vdots & \ddots & \vdots \\ x_1^{nf} & \dots & x_{ng}^{nf} \end{bmatrix} \begin{matrix} \longrightarrow W(X^1) \\ \\ \longrightarrow W(X^{nf}) \end{matrix}$$

4. İterasyon sayısı artırılır;  $it=it+1$ .
5. Popülasyondaki herhangi bir  $i$ 'nci ateşböceği ( $i$ 'nci kafes yapı tasarımı,  $X^i$ ), kendinden daha çekici olan  $j$ 'nci ateşböceğine ( $j$ 'nci kafes yapı tasarımı,  $X^{nj}$ ) doğru hareket ettirilerek, yeni bir tasarım elde edilir ( $X^{yeni}$ ).

6. Bu tasarım için analiz yapılarak düğüm deplasman ve eleman eksenel kuvvetleri elde edilir. Kafes yapının  $W(X^{yeni})$  ile temsil edilen modifiye edilmiş amaç fonksiyonu hesaplanır.
7. Eğer yeni tasarım ait modifiye edilmiş amaç fonksiyonu değeri popülasyondaki en kötü tasarımdan daha iyi ise  $W(X^{yeni}) < W(X^{en-kötü})$  en kötü tasarım elenir, yerine yeni tasarım popülasyona dahil edilir. Popülasyondaki tasarımlar modifiye edilmiş amaç fonksiyonu değerlerine göre tekrar sıralanır, en iyi modifiye edilmiş amaç fonksiyonu değerine sahip olan tasarım mevcut optimum olarak atanır. Eğer iterasyon sayısı, maksimum iterasyon sayısından küçükse  $it < mit$ , 4. adıma gidilir. Değilse 7. adıma gidilir.
8. Optimizasyon işlemi bitirilir, mevcut optimum tasarım nihai optimum tasarım olarak kabul edilir. Bu tasarıma ait ağırlık ise optimum kafes yapı ağırlığıdır.

### Analiz ve Sonuçlar:

Bu çalışmada sunulan ateşböceği algoritması yönteminden elde edilen sonuçları kıyaslayabilmek için literatürde daha önce armoni arama (HS), hibrid parçacık sürü optimizasyonu (PSO), hibrid parçacık sürü karınca koloni algoritması (HPSACO), self-adaptif armoni arama (SAHS) ve ateşböceği algoritması (FFA) yöntemleriyle optimize edilmiş 25 elemanlı uzay kafes yapı kullanılmıştır.

Elde edilen sonuçlar ile; bu çalışmada öne sürülen FFA yöntemiyle elde edilen tasarımın literatürde mevcut olan armoni arama, hibrid parçacık sürü karınca koloni algoritması ve ateşböceği algoritması yöntemlerinden elde edilen tasarımlardan daha iyi olduğu tespit edilmiştir. FFA'nın farklı başlangıç tasarımları için yapılan icraları sonucunda elde edilen 20 farklı kafes yapı tasarımının standart sapma değerinin, ortalama ağırlığa göre %0.17 gibi oldukça küçük bir değerde olması ateşböceği algoritmasının farklı başlangıç tasarımları için global optimuma yakın tasarımları bulabildiğini göstermektedir. Optimum tasarımı elde etmek için gereken kafes yapı analiz sayıları bakımından ise FFA'nın diğer yöntemlere kıyasla daha fazla kafes yapı analizi gerektirdiği tespit edilmiştir.

Mevcut sonuçlar; FFA yönteminin kafes yapı optimizasyonunda kullanılabilecek güçlü bir sezgisel optimizasyon yöntemi olduğunu göstermektedir. FFA yöntemiyle ilgili ileride yapılacak çalışmalar; FFA'nın optimum tasarıma daha hızlı ulaşmasını sağlayacak stratejileri içermesi konusunda olabilir.

### **Kaynakça :**

1. [https://polen.itu.edu.tr/xmlui/bitstream/handle/11527/14623/045\\_d egertekin.pdf?sequence=1&isAllowed=y](https://polen.itu.edu.tr/xmlui/bitstream/handle/11527/14623/045_d egertekin.pdf?sequence=1&isAllowed=y)
2. [http://www.emo.org.tr/ekler/784ced5ad0f6222\\_ek.pdf](http://www.emo.org.tr/ekler/784ced5ad0f6222_ek.pdf)
3. [https://muhendislik.cu.edu.tr/tr/Dergi/\(33\\_1\\_2018\)/09.pdf](https://muhendislik.cu.edu.tr/tr/Dergi/(33_1_2018)/09.pdf)