mühendislikdergisi
Dicle Üniversitesi Mühendislik Fakültesi
Cilt: 7, Sayı: 3,515-526
Eylül 2016

Melez elektromanyetizma benzeri-parçacık sürü optimizasyon algoritması

Soner KIZILOLUK[*,1], Ahmet Bedri ÖZER[2],

[1] *Tunceli Üniversitesi, Bilgisayar Mühendisliği Bölümü, Tunceli*
[2] *Fırat Üniversitesi, Bilgisayar Mühendisliği Bölümü, Elazığ*

Makale Gönderme Tarihi: 20.06.2016 Makale Kabul Tarihi: 25.07.2016

Öz

Optimizasyon, bir problemin alternatif çözümleri içinden en uygununu seçme işlemidir. Optimizasyon problemlerinin çözümü için, kabul edilebilir sürede optimuma yakın çözümler verebilen birçok sezgisel optimizasyon algoritması önerilmiştir. Literatürde çok başarılı sezgisel optimizasyon algoritmaları bulunsa da; tüm problemlerin çözümü için en optimum çözümü bulan algoritmalar henüz tasarlanmamıştır. Bu yüzden yeni sezgisel optimizasyon algoritmaları önerilmekte ya da var olanların daha etkili çalışması için öneriler sunulmaktadır.

Bu çalışmada, global optimizasyon için, Elektromanyetizma Benzeri (EM) algoritma ile Parçacık Sürü Optimizasyon (PSO) algoritmasının birleşiminden oluşan yeni bir melez yöntem olan EM-PSO önerilmiştir. Önerilen yöntemde, PSO algoritmasının hız denklemindeki sabit katsayılar yerine EM algoritmasındaki yük ve toplam kuvvet değerleri kullanılmış ve EM algoritmasındaki parçacıkların hareketi bu denklem ile gerçekleştirilmiştir. Önerilen yöntemin performansı beş farklı kalite testi fonksiyonu kullanılarak test edilmiştir ve sonuçlar standart EM ve PSO algoritmalarının sonuçları ile karşılaştırılmıştır. Deneysel sonuçlar, önerilen yöntemin standart EM ve PSO algoritmalarına göre daha başarılı olduğunu göstermiştir.

Anahtar Kelimeler: *Elektromanyetizma benzeri algoritma; Parçacık sürü optimizasyon algoritması; Melez EM-PSO;*

[*]Yazışmaların yapılacağı yazar: Soner KIZILOLUK. sonerkiziloluk@*tunceli.edu.tr*; Tel: (428) 213 17 94

Giriş

Sezgisel optimizasyon algoritmaları, klasik algoritmalar gibi ilgilenilen problemler üzerinde değişiklik yapmadan kendilerini o problemlere kolaylıkla uyarlayabilmesi, farklı tipteki karar değişkenleri ve sınırlayıcıların olması durumunda bile çözüm stratejileri sunmaları vb. avantajlarından dolayı birçok farklı alanda kullanılmaktadır. Sezgisel optimizasyon algoritmalarının çoğu doğadan esinlenilerek geliştirilmiştir ve esin kaynağına göre sosyal tabanlı, fizik tabanlı, biyolojik tabanlı ve kimya tabanlı olarak çeşitli gruplara ayrılmaktadır. Parlamenter optimizasyon algoritması sosyal tabanlı, elektromanyetizma benzeri algoritma ve yer çekimsel arama algoritması fizik tabanlı genetik algoritmalar biyolojik tabanlı ve yapay kimyasal reaksiyon algoritması kimya tabanlı algoritmalara örnek olarak verilebilir (Kızıloluk ve Alataş, 2012).

Günümüzde literatürde çok başarılı sezgisel optimizasyon algoritmaları bulunsa da; tüm problemler için optimumum çözümü bulan algoritmalar henüz geliştirilmediğinden, sürekli yeni yöntemler önerilmekte veya var olan yöntemlerin etkili çalışması için öneriler sunulmaktadır (Altunbey ve Alataş, 2015). Literatürde, Elektromanyetizma Benzeri (EM) algoritmanın daha etkili çalışması için; EM' nin diğer algoritmalar ile birleşiminden oluşan melez EM-GA(Genetik Algoritma) (Javadi vd., 2013; Chang ve Lee, 2008; Lee ve Chang, 2010), melez EM-GD(Great Deluge algorithm) (Abdullah vd., 2009), melez EM-COA(Chaos Optimization Algorithm) (Wang vd., 2010), melez EM-RA(Restarted Arnoldi Algorithm) (Taheri vd., 2007), melez EM-SA(Simulated Annealing) (Jamili vd., 2011; Naderi vd., 2010) ve melez EM-BP(Back-Propagation technique) (Lee vd., 2012) gibi yeni yöntemler önerilmiştir.

Bu çalışmada EM ile Parçacık Sürü Optimizasyon (PSO) algoritmasının birleşiminden oluşan yeni bir melez yöntem olan EM-PSO önerilmiştir. Önerilen bu yöntem 5 farklı kalite testi fonksiyonu ile test edilmiştir ve sonuçlar standart EM ve PSO algoritmalarının sonuçları ile karşılaştırılmıştır.

Standart EM ve PSO

Bu bölümde standart EM ve PSO algoritmaları hakkında genel bir bilgi verilecektir.

Standart elektromanyetizma benzeri algoritma

İlk olarak Birbil ve Fang (2003) tarafından önerilen EM, Colomb yasasındaki yüklü parçacıklar arasındaki itme-çekme mekanizmasından esinlenilerek geliştirilmiştir. Bu algoritmaya göre, arama uzayındaki her bir eleman yüklü birer parçacık olarak temsil edilir. Her bir parçacığın yükü, o parçacığın uygunluk değeri ile ilişkilendirilmiştir. Parçacıklar yüklerine göre birbirlerini bir kuvvet ile iterler veya çekerler. Bir parçacığa diğer parçacıklar tarafından etki eden toplam kuvvet vektörel olarak hesaplanır ve parçacık bu kuvvet doğrultusunda hareket eder. Bu hareket ile optimum çözüme ulaşılmaya çalışılır. EM algoritması, başlangıç popülasyonu oluşturma, yerel arama, toplam kuvvet hesaplama ve parçacıkların hareketi olmak üzere dört temel evreden oluşur. Algoritmanın genel yapısının sözde kodu Tablo 1' de verilmiştir:

Tablo 1. EM' nin genel yapısı

```
1: başlangıçpopulasyonu()
2: iterasyon ←1
3: while iterasyon <MAXITER do
4:     yerelarama(LSITER, δ)
5:     toplamkuvvet()
6:     hareket()
7:     iterasyon ←iterasyon + 1
8: end while
```

Kodda, *MAXITER* maksimum iterasyon sayısını, *LSITER* yerel aramadaki maksimum iterasyon sayısını ve δ yerel arama parametresini temsil etmektedir.

Algoritmanın başlangıç popülasyonu oluşturma evresinde, m adet parçacık n boyutlu arama uzayı sınırları dâhilinde gelişigüzel oluşturulur. Her parçacığın $f(x)$ uygunluk fonksiyonu ile uygunluk değeri hesaplanır. Daha sonra en iyi uygunluk değerine sahip olan parçacık x^{best}

olarak saklanır. Tablo 2'de bu evrenin sözde kodu verilmiştir (Birbil ve Fang, 2003).

Tablo 2. Başlangıç popülasyonu oluşturma

```
for i = 1 to m do
    for k = 1 to n do
        λ ← random (0, 1)
        x_k^i ← l_k + λ(u_k − l_k)
    end for
    hesapla f(x^i)
end for
x^best ←min{f(x^i),∀i}
```

Kodda λ 0 ile 1 arasında gelişigüzel bir sayıyı, u_k ve l_k ise sırasıyla arama uzayının üst ve alt sınırını temsil etmektedir. Algoritmanın yerel arama evresi ise x^i noktası için yerel bilgi toplamak amacıyla kullanılır. Bu evrenin sözde kodu Tablo 3' deki gibidir.

Tablo 3. Yerel arama

```
1: sayaç←1
2: Length← δ(max_k {u_k-l_k})
3: for i = 1 to m do
4:    for k = 1 to n do
5:       λ1←random(0,1)
6:       while sayaç <LSITER do
7:          y ← x^i
8:          λ2←random(0,1)
9:          if λ1 > 0.5 then
10:             y_k←y_k+λ2(Lenght)
11:         else
12:             y_k←y_k-λ2(Lenght)
13:         end if
14:         if f (y) < f(x^i) then
15:             x^i← y
16:             sayaç← LSITER − 1
17:         end if
18:         sayaç ←sayaç + 1
19:      end while
20:   end for
21: end for
22: x^best ←min{f(x^i),∀i}
```

Sözde koda göre, ilk başta, maksimum uygun basamak uzunluğu (*Length*), δ parametresine göre hesaplanır (adım 2). Daha sonra x^i parçacığı geçici y değişkenine atanır ve y değişkenine gelişigüzel λ_2 sayısı eklenerek veya çıkarılarak koordinat koordinat arama yapılır. Bu arama belirlenen *LSITER* iterasyon sayısı kadar devam eder (adım 5-19). Bu arama sırasında y noktası x^i' den daha iyi bir konuma ulaşırsa x^i' ye y' nin değeri atanır ve o parçacık için arama sona erer (adım 14-17). En son mevcut en iyi parçacık olan x^{best} güncellenir (adım 22) (Birbil ve Fang, 2003).

Toplam kuvvet hesaplama evresinde ise, ilk başta parçacıkların yükü, uygunluk fonksiyonu değerlerine göre denklem 1' de gösterildiği gibi hesaplanır. $f(x^i)$ *i.* parçacığın uygunluk değerini ve $f(x^{best})$ ise en iyi parçacığın uygunluk değerini temsil eder.

$$q^i = exp\left(-n\frac{f(x^i)-f(x^{best})}{\sum_{k=1}^{m}(f(x^k)-f(x^{best}))}\right), \forall i. \qquad (1)$$

Daha sonra i parçacığına, diğer parçacıklar tarafından etki eden toplam F^i kuvveti denklem 2' de gösterildiği gibi hesaplanır. Denklemde $\|x^j\text{-}x^i\|^2$ iki parçacık arasındaki Öklid uzaklığıdır.

$$F^i = \sum_{j\neq i}^{m}\begin{Bmatrix}(x^j - x^i)\frac{q^i q^j}{\|x^j - x^i\|^2} & eğer\ f(x^j) < f(x^i)\\ (x^i - x^j)\frac{q^i q^j}{\|x^j - x^i\|^2} & eğer\ f(x^j) \geq f(x^i)\end{Bmatrix} \qquad (2)$$

Toplam kuvvet hesaplama evresinin sözde kodu Tablo 4' deki gibidir (Birbil ve Fang, 2003).

Tablo 4. Toplam kuvvet hesaplama

```
for i=1 to m do
   hesapla q^i, F^i←0
end for
for i=1 to m do
   for j=1 to m do
      if f(x^j) < f(x^i) then
         F^i ← F^i + (x^j − x^i) q^i q^j / ||x^j − x^i||^2 {Çekim}
      else
         F^i ← F^i − (x^j − x^i) q^i q^j / ||x^j − x^i||^2 {İtim}
      end if
   end for
end for
```

Algoritmanın hareket evresinde ise i parçacığı denklem 3' te verildiği gibi, F^i kuvveti doğrultusunda arama uzayının alt ve üst

sınırlarını aşmayacak şekilde λ sayısı ile gelişigüzel hareket ettirilir.

$$x^i = x^i + \lambda \frac{F^i}{||F^i||}(u_k - l_k) \quad (3)$$

Hareket evresinin sözde kodu ise Tablo 5' de gösterilmiştir (Birbil ve Fang, 2003).

Tablo 5. Parçacıkların hareketi

```
1: for i = 1 to m do
2:     if i ≠ x^best then
3:         λ ← U (0, 1)
4:         F_k^i ← F_k^i / ||F_k^i||
5:         for k = 1 to n do
6:             if F_k^i > 0 then
7:                 x_k^i ← x_k^i + λF_k^i(u_k − x_k^i)
8:             else
9:                 x_k^i ← x_k^i + λF_k^i(x_k^i − l_k)
10:            end if
11:        end for
12:    end if
13: end for
```

Algoritmanın yerel arama, toplam kuvvet hesaplama ve hareket evreleri bitim şartı sağlanana kadar tekrar eder. Algoritma sonunda en iyi uygunluk değerine sahip olan parçacık çözüm olarak seçilir.

Standart parçacık sürü optimizasyon algoritması

İlk olarak Kennedy ve Eberhart (1995) tarafından önerilen PSO, kuş veya balık sürülerinin sosyal davranışlarından esinlenilerek geliştirilmiştir. PSO bireyler arasındaki sosyal bilgi paylaşımını esas alır. Her bireye parçacık adı verilir. Algoritmada başlangıç popülasyonu rastgele üretilmiş parçacıklar ile başlatılır. Her iterasyonda her parçacık *pbest* ve *gbest* değerlerine göre güncellenir. *pbest* o parçacığın o ana kadar bulduğu en iyi değer, *gbest* ise tüm popülâsyonda o ana kadar bulunmuş en iyi değerdir. Bu iki değer dikkate alınarak denklem 4 ve denklem 5' teki gibi parçacıkların hızları ve konumları güncellenir.

$$V_{i(t+1)}=w.V_{i(t)} + c_1.rand1.(pbest_i - X_i) + c_2.rand2.(gbest - X_i) \quad (4)$$

$$X_{i(t+1)}=X_{i(t)} + V_{i(t+1)} \quad (5)$$

Burada $X_{i(t)}$ ve $X_{i(t+1)}$ parçacıkların mevcut ve bir adım sonraki konumlarını, $V_{i(t)}$ ve $Vi_{(t+1)}$ ise parçacıkların mevcut ve bir adım sonraki hızlarını verir. *rand1* ve *rand2* 0 ile 1 arasında gelişigüzel üretilmiş sayılardır. *w* atalet katsayısı, c_1 ve c_2 ise öğrenme katsayılarıdır ve hızlanma katsayıları olarak ta adlandırılır (Özsağlam ve Çunkaş, 2008). c_1 ve c_2 katsayıları, bir iterasyonda bir parçacığın kendi en iyi ve global en iyi pozisyonlarına doğru olan hızlanmanın alabileceği adımın maksimum boyutunu etkiler. Düşük değerlerin seçilmesi parçacıkların hedef bölgeye ulaşmadan önce, uzak yerlerde dolaşmalarına imkân verir. Yüksek değerlerin seçilmesi ise hedefe ulaşmayı hızlandırır fakat beklenmedik hareketlerin oluşmasına ve hedef bölgeye ulaşılmamasına neden olabilir. Algoritma temel olarak aşağıdaki basamaklardan oluşur (Kennedy ve Eberhart, 1995; Özsağlam ve Çunkaş, 2008);

1. Parçacıkların başlangıç pozisyonlarını ve hızlarını gelişigüzel oluştur.
2. Tüm parçacıkların uygunluk değerlerini hesapla.
3. Her bir parçacık için mevcut jenerasyondan yerel en iyiyi (pbest) bul.
4. Mevcut jenerasyondaki yerel en iyiler içerisinden global en iyiyi (gbest) seç.
5. Pozisyonları ve hızları denklem 4 ve denklem 5' teki gibi yenile.
6. Bitim şartı sağlamıyorsa adım 2' ye git.

Melez EM-PSO algoritması

EM algoritmasında, bir parçacığın hareketi, kendi en iyi konumunu ve global en iyi konumu bilmeden, tamamen o parçacık üzerinde diğer parçacıklar tarafında etki eden toplam kuvvet oranında ve yönünde olmaktadır. Bu çalışmada EM algoritmasının hareket evresi, PSO algoritması ile gerçekleştirilerek yeni bir melez EM-PSO algoritması önerilmiştir. EM-PSO algoritmasında, hareket evresinde parçacıkların hareketi, PSO algoritmasının hız ve konum güncelleme denklemleri kullanılarak gerçekleştirilmiştir. PSO' nun hız denklemindeki hızlanma katsayıları olan c_1, c_2

ile $rand_1$ ve $rand_2$ değerleri yerine EM algoritmasındaki q ve F değerleri kullanılmıştır. EM-PSO' nun hız ve konum güncelleme denklemleri, denklem 6 ve denklem 7' deki gibidir:

$$V_k^i = w.V_k^i + q^i.(pbest_k^i - X_k^i) + |F_k^i|.(gbest_k - X_k^i) \quad (6)$$

$$X_k^i = X_k^i + V_k^i \quad (7)$$

Bu sayede, bir parçacık kendi yük değeri olan q^i büyüklüğünde bir adımla kendi kişisel en iyisine ve üzerine etki eden toplam kuvvetin mutlak değeri olan F_k^i büyüklüğünde global en iyiye doğru adım atmış olacaktır. Daha sonra denklem 7' deki gibi parçacığın konumu, hesaplanan hız (maksimum hız:Vmax ve minimum hız:Vmin değerlerini aşmayacak) ile güncellenir. EM-PSO algoritmasının genel yapısının sözde kodu Tablo 6' daki gibidir:

Tablo 6. EM-PSO' nun temel yapısı

```
1: başlangıçpopulasyonu()
2: iterasyon ←1
3: while iterasyon <MAXITER do
4:     yerelarama(LSITER, δ)
5:     pbest, gbest ve w güncelle
6:     toplamkuvvet()
7:     hareket()
8:     iterasyon ←iterasyon + 1
9: end while
```

EM-PSO' da yerel arama ve toplam kuvvet evreleri EM algoritması ile aynıdır. EM' nin başlangıç popülasyonunu oluşturma evresine ek olarak EM-PSO' da başlangıç hızları, maksimum hız (Vmax), minimum hız (Vmin) ve atalet katsayısının (w) başlangıç değerleri de oluşturulur. Durdurma kriteri sağlanıncaya kadar EM-PSO' nun 4, 5, 6 ve 7 numaralı adımları tekrar edilir. Algoritma sonunda en iyi uygunluk değerine sahip olan parçacık çözüm olarak seçilir. EM-PSO algoritmasının hareket evresinin sözde kodu ise Tablo 7' deki gibidir.

Tablo 7. EM-PSO hareket evresi

```
1: for i = 1 to m do
2:     if i ≠ x^best then
3:         λ ← U (0, 1)
4:         F_k^i ← F_k^i / ||F_k^i||
5:         for k = 1 to n do
6:             V_k^i = w. V_k^i + q^i. (pbest_k^i − X_k^i) +
               |F_k^i|. (gbest_k − X_k^i)
7:             if V_k^i>Vmax
8:                 V_k^i=Vmax
9:             else if V_k^i<Vmin
10:                V_k^i=Vmin
11:            end if
12:            X_k^i = X_k^i + V_k^i
13:        end for
14:    end if
15: end for
```

Deneysel çalışma

Kullanılan kalite testi fonksiyonları

Optimizasyon algoritmalarının performansını ölçmek amacıyla kalite testi fonksiyonları kullanılabilir. Bu kalite testi fonksiyonlarının çoğu mühendislik problemlerinin karmaşıklığına sahiptir. Ayrıca fonksiyonların zorluk dereceleri parametrelerinin değiştirilmesi ile ayarlanabilir (Ozer, 2010; Alatas vd. 2009).

Bu çalışmada önerilen EM-PSO algoritmasının performansı, Tablo 8' de özellikleri verilen, birden çok yerel optimuma ve tek bir global optimuma sahip olan çok-modlu Griewank, Ackley ve Rastrigin fonksiyonları ile sadece tek bir global optimuma sahip olan Rosenbrock ve De jong kalite testi fonksiyonları kullanılarak test edilmiştir ve sonuçlar standart EM ve PSO algoritmaları ile karşılaştırılmıştır.

Rastrigin fonksiyonu

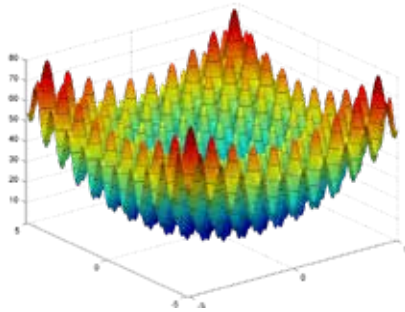
Bu fonksiyonun çok sayıda yerel minimum noktası bulunmaktadır ve geniş arama uzayına sahiptir. Fonksiyon çok-modludur ve doğrusal değildir (Can ve Alataş, 2015; Rastrigin, 1974). Fonksiyonun tanımı denklem 8' de, grafiği ise Şekil 1' de gösterilmiştir.

$$f_1(x) = 10d + \sum_{i=1}^{d}[x_i^2 - 10\cos(2\pi x_i)] \quad (8)$$

Sınırları $-5.12 \leq x \leq 5.12$ olarak belirlenmiştir. Global minimum noktası $x^* = (0,0,..,0)$ ve $f(x^*) = 0$.

Tablo 8. Kalite testi fonksiyonları özellikleri

Fonksiyon adı	Alt sınır	Üst sınır	Optimum	Tip
Rastrigin	-5.12	5.12	0	Çok-modlu
Rosenbrock	-2.048	2.048	0	Tek-modlu
De Jong	-5.12	5.12	0	Tek-modlu
Griewank	-10	10	0	Çok-modlu
Ackley	-32	32	0	Çok-modlu



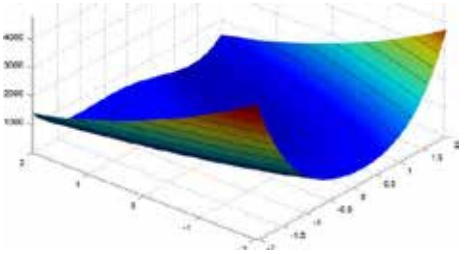
Şekil 1. Rastrigin fonksiyonu

Rosenbrock fonksiyonu

Global minimum noktası uzun ve dar bir vadide yer alan tek-modlu bir fonksiyondur. Vadiye ulaşmak kolaydır fakat global minimuma yakınsamak zordur (Jamil ve Yang, 2013).

$$f_2(x) = \sum_{i=1}^{d} [100(x_i^2 - x_{i+1})^2 + (1 - x_i^2)] \quad (9)$$

Sınırları $-2.048 \leq x \leq 2.048$ olarak belirlenmiştir. Global minimumu $x^* = (1,1,..,1)$ noktasındadır ve $f(x^*) = 0$. Şekil 2' de fonksiyonun grafiği görülmektedir.



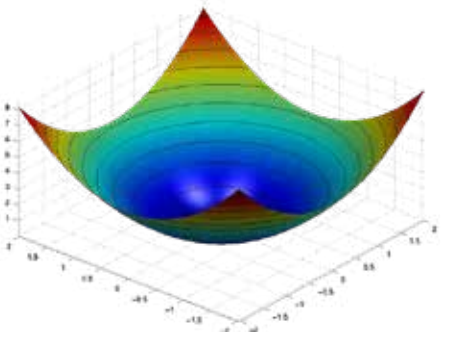
Şekil 2. Rosenbrock fonksiyonu

De jong fonksiyonu

Minimizasyon amacıyla kullanılan bir fonksiyondur. Fonksiyon tek-modludur (Jamil ve Yang, 2013).

$$f_3(x) = \sum_{i=1}^{d} x_i^2 \quad (10)$$

Sınırları $-5.12 \leq x \leq 5.12$ olarak belirlenmiştir. Global minimum noktası $x^*=(0,0,\ldots,0)$ ve $f(x^*) = 0$. Şekil 3' te fonksiyonun grafiği görülmektedir.



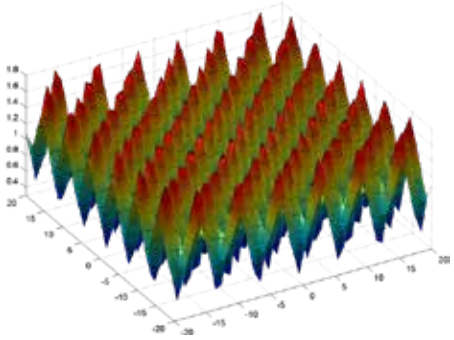
Şekil 3. De jong fonksiyonu

Griewank fonksiyonu

Düzenli dağıtılmış birçok yerel optimum noktasına sahiptir. Bu fonksiyon sürekli, çok-modlu, ölçeklenebilir, konveks ve ikinci dereceden bir fonksiyondur (Can ve Alataş,2015; Jamil ve Yang, 2013).

$$f_4(x) = \sum_{i=1}^{d} \left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{d} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \quad (11)$$

Sınırlar $-10 \leq x \leq 10$ olarak belilenmiştir. Global minimum noktası x*=(0,0,…,0) ve f(x*) = 0. Şekil 4' te fonksiyonun grafiği görülmektedir.



Şekil 4. Griewank fonksiyonu

Ackley fonksiyonu

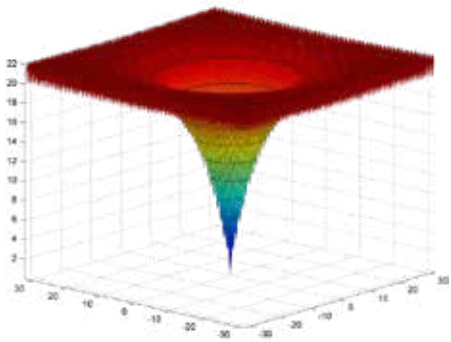
Birçok lokal derin optimum noktası olan çok-modlu bir fonksiyondur. (Can ve Alataş,2015; Jamil ve Yang, 2013).

$$f_5(x) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) \quad (12)$$

Sınırlar $-32 \leq x \leq 32$ olarak belirlenmiştir. Global minimum noktası x*=(0,0,…,0) ve f(x*) = 0. Şekil 5' de fonksiyonun grafiği görülmektedir.

Deneysel sonuçlar

Önerilen EM-PSO, EM ve PSO algoritmaları üst bölümlerde bahsedilen 5 farklı kalite testi fonksiyonu kullanılarak karşılaştırılmıştır. Algoritmalar, Mathworks firmasının, teknik hesaplamalar ve matematiksel problemlerin çözümü ve analizi için geliştirdiği Matlab 7.8 paket programı ile kodlanmıştır. Tüm deneyler için algoritmaların başlangıç popülasyonları 30 olarak alınmıştır. Fonksiyonların boyutları ise 10 olarak alınmıştır. EM-PSO ve EM' de atalet katsayısı olan w, 0.9 dan 0.6 ya her iterasyonda doğrusal olarak azalacak şekilde ayarlanmıştır. Vmax ve Vmin değerleri ise griewank fonksiyonu için; arama uzayı alt ve üst sınırı/30, diğer fonksiyonlar için ise; arama uzayı alt ve üst sınırı/10 olarak ayarlanmıştır. PSO' da öğrenme katsayıları c_1 ve c_2 2 olarak ayarlanmıştır. Algoritmaların sonlandırma şartı 1000 iterasyon olarak seçilmiştir. Her algoritma 10 defa çalıştırılmıştır ve karşılaştırmalar elde edilen ortalama ve en iyi sonuçlar dikkate alınarak yapılmıştır. Her kalite testi fonksiyonu için 10 ar çalıştırmada elde edilen sonuçlar Tablo 9, Tablo 10, Tablo 11, Tablo 12 ve Tablo 13' de gösterilmiştir.



Şekil 5. Ackley fonksiyonu

Tablo 9. Rastrigin fonksiyonu için başarı oranları

	EM-PSO	EM	PSO
1	2.42E-08	0.000274	5.969749
2	1.41E-06	5.89E-05	9.949586
3	1.01E-05	0.000118	4.974795
4	1.05E-06	0.000239	5.969754
5	2.17E-06	9.89E-05	5.969755
6	1.44E-06	0.001061	4.974795
7	2.3E-07	6.56E-05	8.954626
8	6.11E-07	0.000211	5.969754
9	5.13E-06	0.000371	6.964713
10	1.95E-05	0.000154	8.954626
Ortalama	4.16E-06	0.000265	6.865216
Minimum	2.42E-08	5.89E-05	4.974795

Tablo 10. Rosenbrock fonksiyonu için başarı oranları

	EM-PSO	EM	PSO
1	0.28741	4.92568	5.03534
2	4.12845	3.28788	4.71
3	0.22604	5.40072	4.80785
4	4.3145	3.71558	4.6888
5	0.19318	3.06348	4.79752
6	0.48097	3.84481	5.15593
7	0.33836	5.35469	4.78998
8	0.22311	5.33317	4.97406
9	0.25016	4.03205	4.75724
10	0.18943	5.45297	5.18402
Ortalama	1.06316	4.4411	4.89007
Minimum	0.18943	3.06348	4.6888

Tablo 11. De Jong fonksiyonu için başarı oranları

	EM-PSO	EM	PSO
1	4.13E-11	3.71E-07	1.91E-12
2	2.13E-12	1.38E-06	5.25E-13
3	3.92E-11	6.79E-07	5.71E-12
4	2.96E-11	2.69E-06	2.61E-12
5	2.43E-13	1.64E-06	3.40E-11
6	9.11E-17	2.08E-07	4.32E-11
7	1.61E-11	6.46E-07	4.96E-11
8	7.59E-12	2.44E-07	6.95E-12
9	1.11E-11	9.74E-07	4E-12
10	3.07E-14	1.23E-06	3.96E-12
Ortalama	1.47E-11	1.01E-06	1.52E-11
Minimum	9.11E-17	2.08E-07	5.25E-13

Tablo 12. Griewank fonksiyonu için başarı oranları

	EM-PSO	EM	PSO
1	5.16E-14	5.7E-07	0.058113
2	6.33E-15	4.57E-07	0.031972
3	5.55E-16	4.46E-07	0.032605
4	2.33E-12	1.25E-06	0.019697
5	5.55E-16	6.22E-06	0.024637
6	1.22E-12	5.03E-07	0.012316
7	1.44E-15	2.72E-07	0.031999
8	3.77E-15	2.54E-07	0.02768
9	3.37E-13	1.25E-07	0.041831
10	1.79E-13	1.53E-06	0.036924
Ortalama	4.14E-13	1.16E-06	0.031777
Minimum	5.55E-16	1.25E-07	0.012316

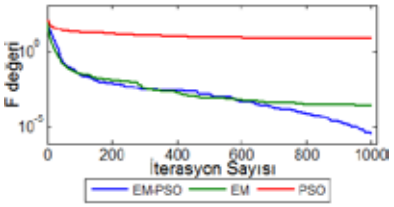
Tablo 13. Ackley fonksiyonu için başarı oranları

	EM-PSO	EM	PSO
1	1.34E-05	0.010026	1.82E-05
2	9.12E-08	0.009104	2.09E-05
3	4.18E-07	0.003248	5.38E-05
4	2.78E-05	0.008508	1.69E-05
5	1.72E-05	0.006142	4.49E-05
6	5.42E-06	0.008232	3.15E-06
7	3.16E-07	0.009472	5.51E-06
8	2.47E-05	0.012756	4.20E-06
9	2.04E-05	0.012943	1.11E-05
10	2.61E-05	0.014948	1.04E-05
Ortalama	1.36E-05	0.009538	1.89E-05
Minimum	9.12E-08	0.003248	3.15E-06

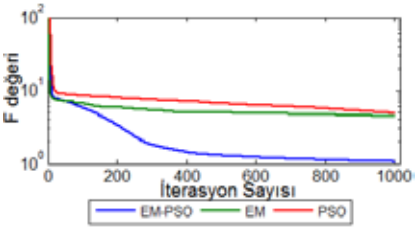
Tablo 14' de ise EM-PSO, EM ve PSO algoritmalarının, tüm kalite testi fonksiyonlarından elde edilen ortalama ve minimum sonuçları görülmektedir. Şekil 6, Şekil 7, Şekil 8, Şekil 9 ve Şekil 10' da tüm kalite testi fonksiyonlarından elde edilen ortalama sonuçlara göre iterasyon-F değeri grafikleri verilmiştir.

Tablo 14. Kalite testi fonksiyonlarında EM-PSO, EM ve PSO sonuçları

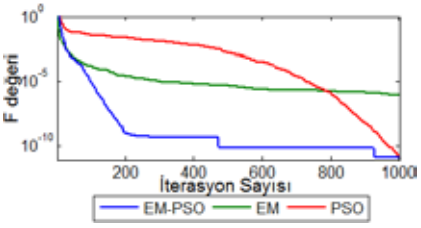
Fonk.		EM-PSO	EM	PSO
f1	Ortalama	**4.16E-06**	0.000265	6.865216
	Minimum	**2.42E-08**	5.89E-05	4.974795
f2	Ortalama	**1.06316**	4.4411	4.89007
	Minimum	**0.18943**	3.06348	4.6888
f3	Ortalama	**1.47E-11**	1.01E-06	1.52E-11
	Minimum	**9.11E-17**	2.08E-07	5.25E-13
f4	Ortalama	**4.14E-13**	1.16E-06	0.031777
	Minimum	**5.55E-16**	1.25E-07	0.012316
f5	Ortalama	**1.36E-05**	0.009538	1.89E-05
	Minimum	**9.12E-08**	0.003248	3.15E-06



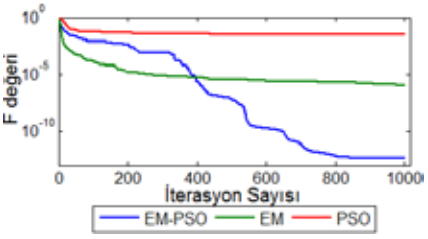
Şekil 6. Rastrigin fonksiyonu için iterasyon-F değeri grafiği



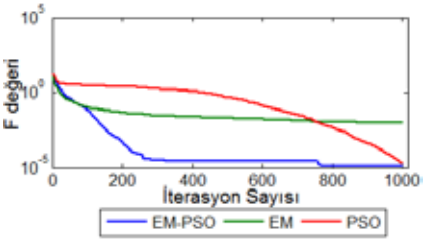
Şekil 7. Rosenbrock fonksiyonu için iterasyon-F değeri grafiği



Şekil 8. De Jong fonksiyonu için iterasyon-F değeri grafiği



Şekil 9. Griewank fonksiyonu için iterasyon-F değeri grafiği



Şekil 6. Ackley fonksiyonu için iterasyon-F değeri grafiği

Tablo 14' deki sayısal sonuçlara bakıldığında, önerilen EM-PSO algoritmasının, standart EM ve standart PSO algoritmasına göre tüm kalite testi fonksiyonlarında daha iyi sonuçlar elde ettiği görülmüştür. Rastrigin fonksiyonunda ortalama sonuçlar EM-PSO' da 4.16E-06 EM' de ise 0.0002 olarak elde edilmiştir. PSO ise 6.86' da kalmıştır. Rosenbrock fonksiyonunda ortalama sonuçlar EM-PSO' da 1.06, EM' de 4.44 ve PSO' da ise 4.89 olarak elde edilmiştir. De jong fonksiyonunda ortalama sonuçlar EM-PSO' da 1.47E-11 ve PSO' da 1.52E-11 olarak birbirlerine yakın değerler elde edilmiştir. EM ise 1.01E-06' da kalmıştır. Fakat minimum değer EM-PSO' da 9.11E-17 ve PSO' da 5.25E-13 şeklinde olmuştur. Griewank fonksiyonunda EM-PSO' da ortalama sonuç 4.14E-13 ve minimum sonuç 5.55E-16 olmuştur. EM 1.16E-06' da ve PSO 0.03' de kalmıştır. Ackley fonksiyonunda EM-PSO ve PSO' nun ortalama sonuçları sırasıyla 1.36E-05 ve 1.89E-05 olarak birbirine yakın olmuştur. EM ise 0.009' da kalmıştır. Minimum sonuçlar dikkate alındığında EM-PSO 9.12E-08 ve PSO 3.15E-06 olmuştur.

İterasyon sayısı-F değeri grafiklerine bakıldığında, rosenbrock, de jong ve ackley fonksiyonlarında EM-PSO, EM ve PSO algoritmalarına göre çok daha az iterasyonda hızlı bir şekilde minimuma yaklaşmış ve geri kalan iterasyonlarda ise daha yavaş bir biçimde minimuma yaklaşmaya devam etmiştir. Rastrigin fonksiyonunda EM-PSO ve EM yaklaşık 650. iterasyona kadar benzer bir karakterde minimuma yaklaşırken, geri kalan iterasyonlarda ise EM yerel optimum noktalarına takılmış, EM-PSO ise minimuma yaklaşmaya devam etmiştir. PSO' nun ise çok az iterasyon sonucunda yerel optimum noktalarına takıldığı görülmüştür. Griewank fonksiyonunda ise PSO çok az bir iterasyonda, EM ise yaklaşık 300 iterasyon sonunda yerel optimum noktalarına takılırken, EM-PSO' nun yaklaşık 850 iterasyon boyunca minimuma yaklaştığı görülmüştür.

Sonuçlar

Sezgisel optimizasyon algoritmaları, klasik algoritmalar gibi ilgilenilen problemler üzerinde değişiklik yapmadan kendilerini o problemlere kolaylıkla uyarlayabilmesi, farklı tipteki karar değişkenleri ve sınırlayıcıların olması durumunda bile çözüm stratejileri sunmaları vb. avantajlarından dolayı birçok farklı alanda kullanılmaktadır. Literatürde çok başarılı algoritmalar bulunsa da; tüm problemler için en iyi sonucu veren algoritmalar henüz tasarlanmamıştır. Bu yüzden yeni yöntemler önerilmekte ya da var olan yöntemlerin daha etkili çalışması için öneriler sunulmaktadır.

Bu çalışmada EM ile PSO algoritmasının birleşiminden oluşan yeni bir melez yöntem olan EM-PSO algoritması önerilmiştir. EM algoritmasında parçacıklar, kendi en iyi konumunu ve global en iyi konumu bilmeden, diğer parçacıklardan etki eden toplam kuvvet oranında ve yönünde hareket etmektedir. EM-PSO algoritmasında ise parçacıklar yükleri büyüklüğünde bir adım ile kişisel en iyisine, üzerine etki eden toplam kuvvet büyüklüğünde ise global en iyiye doğru hareket etmesi sağlanmıştır.

EM-PSO algoritması 5 farklı kalite testi fonksiyonu ile test edilmiştir ve sonuçlar standart EM ve standart PSO algoritmalarının sonuçları ile karşılaştırılmıştır. Deneysel sonuçlara bakıldığında EM-PSO' nun tüm kalite testi fonksiyonlarında daha başarılı sonuçlar elde ettiği görülmüş, ayrıca grafikler incelendiğinde EM-PSO' nun zor kabul edilen çok-modlu kalite testi fonksiyonlarında bile yerel optimum noktalarına takılmadan, sürekli bir biçimde global minimuma doğru ilerleyerek iyi çözümler üretebildiği görülmüştür. Böylece önerilen algoritmanın global optimizasyon için etkili bir yöntem olduğu ortaya konmuştur. İleriki çalışmalarda önerilen algoritmanın veri madenciliği tekniklerinden birliktelik kural keşfi ve sınıflandırma alanlarında başarısının test edilmesi planlanmaktadır. Ayrıca algoritmanın çok amaçlı halinin, paralel ve dağıtık versiyonlarının da geliştirilmesi ileride yapılabilecek çalışmalar arasındadır.

Kaynaklar

Abdullah, S., Turabieh, H. ve Mccollum, B., (2009). A hybridization of electromagnetic-like mechanism and great deluge for examination timetabling problems, *Hybrid Metaheuristics*, LNCS 5818, 60–72, Berlin: Springer.

Alatas, B., Akin, E. ve Ozer, A. B., (2009). Chaos embedded particle swarm optimization algorithms, *Chaos, Solitons and Fractals*, 40(4), 1715–1734.

Altunbey, F. ve Alataş, B., (2015). Sosyal ağ analizi için sosyal tabanlı yapay zekâ optimizasyon algoritmalarının incelenmesi, *International Journal of Pure and Applied Sciences*, **1**, 1, 33-52.

Birbil, Ş. İ., ve Fang, S. C., (2003). An electromagnetism-like mechanism for global optimization, *Journal of global optimization*, ***25***, 3, 263-282.

Can, Ü. ve Alataş, B., (2015). Bitki zekâsında yeni bir alan: kök kütlesi optimizasyonu, *Türk Doğa Ve Fen Dergisi*, 8.

Chang, F.K. ve Lee, C.H., (2008). Design of fractional PID control via hybrid of electromagnetism-like and genetic algorithms, *Eighth International Conference on Intelligent Systems Design and Applications (ISDA)*, 525–530, IEEE.

Jamil, M. ve Yang, X. S., (2013). A literature survey of benchmark functions for global optimisation problems, *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2), 150-194.

Jamili, A., Shafia, M. A. ve Tavakkoli-Moghaddam, R., (2011). A hybridization of simulated annealing and electromagnetism-like mechanism for a periodic job shop scheduling problem, *Expert Systems with Applications*, 38, 5895–5901.

Javadi, B., Jolai, F., Slomp, J., Rabbani, M. ve Tavakkoli-Moghaddam, R., (2013). A hybrid electromagnetism-like algorithm for dynamic inter/intra cell layout problem, *International Journal of Computer Integrated Manufacturing*, **27**, 6, 501–518.

Kennedy, J. ve Eberhart, R., (1995). Particle swarm optimisation, *In Proceedings of the IEEE international conference on neural networks*, pp. 1942–1948, Washington DC: IEEE Computer Society.

Kızıloluk, S. ve Alataş, B., (2012). Sosyal tabanlı güncel sezgisel optimizasyon algoritmaları, *Cumhuriyet Üniversitesi İktisadi ve İdari Bilimler Dergisi*, **13**, 2, 39-56.

Lee, C.H. ve Chang, F.K., (2010). Fractional-order PID controller optimization via improved electromagnetism-like algorithm*, Expert Systems with Applications*, 37, 8871–8878.

Lee, C.-H., Chang, F.K., Kuo, C.T., ve Chang, H. H., (2012). A hybrid of electromagnetismlike mechanism and back-propagation algorithms for recurrent neural fuzzy systems design, *International Journal of Systems Science*, 43, 231–247.

Naderi, B., Tavakkoli-Moghaddam, R. ve Khalili, M., (2010). Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan, *Knowledge-Based Systems*, 23, 77–85.

Ozer, A. B., (2010). CIDE: chaotically initialized differential evolution, *Expert Systems with Applications*, 37(6), 4632-4641.

Özsağlam, M. Y. ve Çunkaş, M., (2008). Optimizasyon problemlerinin çözümü için parçaçık sürü optimizasyonu algoritması, *Politeknik Dergisi*, **11**, 4.

Rastrigin, L. A., (1974). Extremal control systems, *In Theoretical foundations of engineering cybernetics series*, Moscow, Nauka, Russian.

Taheri, S. H., Ghazvini, H., Saberi-Nadjafi, J. ve Biazar, J., (2007). A hybrid of the restarted Arnoldi and electromagnetism meta-heuristic methods for calculating eigenvalues and eigenvectors of a non-symmetric matrix, *Applied Mathematics and Computation*, 191, 79–88.

Wang, Q., Zeng, J. ve Song, W., (2010). A new electromagnetism-like algorithm with chaos optimization, *International Conference on Computational Aspects of Social Networks (CASoN)*, 535–538, IEEE.

Hybrid electromagnetism like–particle swarm optimization algorithm

Extended abstract

Optimization is the process of selecting the most appropriate solution in all alternative solutions of a problem. A lot of meta-heuristic optimization algorithms are proposed to find approximate solutions for solving optimization problems in acceptable time. Although there are many successful meta-heuristic optimization algorithms in the literature, an algorithm hasn't been designed to find optimum solutions for solving all optimization problems yet. Therefore, new meta-heuristic algorithms are proposed or the existing algorithms are modified for better results. In this paper, a novel hybrid optimization algorithm EM-PSO which is consist of the Electromagnetism-like (EM) algorithm and the Particle Swarm Optimization (PSO) algorithm has been proposed for global optimization.

The EM was firstly proposed by Birbil and Fang in 2003. The EM is inspired by attraction-repulsion mechanism of electromagnetism theory. Each member in the search space is considered as a charged particle. The general algorithm consists of four phases. These are initialization of algorithm, local search, calculation of the total force, and movement. In initialization phase, the particles are initialized with random positions between the corresponding upper bound and lower bound of the search space. After, the objective function values of the each particle are calculated. Finally, the best objective function value is stored as x^{best}. In the local search phase, local information about each particle is gathered. In the calculation of the total force phase, the charges of the each particle are calculated according to their objective function values. Then, the total force exerted on a particle from other particles is calculated. In movement phase, each particle is moved in the direction of the total force by a random step length. If maximum number of iterations is reached then algorithm terminates. At the end of the algorithm, the best particle is selected as the solution of the optimization problem.

The PSO is proposed by Kennedy and Eberhart in 1995 which is an evolutionary computation technique. The PSO was inspired from social behaviors of bird and fish swarms. Each member in the search space is considered as a particle. The population is considered as a swarm. The PSO algorithm focused on initializing and particle movement. The particles are initialized with random positions throughout the search space. Each particle should consider the current position and its best position (pbest). Moreover, each particle should know the best position of the swarm (gbest). With this information, the velocities of the each particle toward its pbest and gbest are calculated. Then each particle is moved by its velocity. If maximum number of iterations is reached then algorithm terminates. At the end of the algorithm, the best particle is selected as the solution of the optimization problem.

In EM algorithm, each particle is only moved in the direction of the total force. In EM-PSO algorithm, the movement of particles in movement phase of EM has been executed with PSO. The values q and F in the EM have been used instead of the values c_1, c_2, $rand_1$ and $rand_2$ in the velocity equation that is in PSO. In this way, the movement of the each particle is influenced by its best position (pbest) and the best position of the swarm (gbest).

The performance of proposed hybrid EM-PSO algorithm has been tested with five benchmark functions which are Rastrigin, Rosenbrock, De Jong, Griewang and Ackley. The obtained test results have been compared with standard EM and standard PSO algorithms' test results. According to the all benchmark test results we have showed that the new proposed hybrid algorithm is successful than standard EM and standard PSO.

In the future works, parallel and distributed versions of EM-PSO algorithm can be developed. Generalization of EM-PSO algorithm for multi-objective optimization problems can also be one of the further works.

Keywords: *Electromagnetism-like algorithm; Particle swarm optimization algorithm; Hybrid EM-PSO;*