



YMT320 – Optimizasyon Teknikleri

Wolf Colony Search Algorithm

Proje Çalışma Ekibi

Çağlar TELEF

Proje Danışmanları

Doç. Dr. Bilal Alataş

Güncel Sürü Zekâsı Optimizasyon Algoritmaları

Optimizasyon, bir problemde belirli koşullar altında çözüm olabilecek sonuçlar içerisinde en iyi olanı seçme işlemidir. Optimizasyon problemleri için birçok algoritmalar vardır. Genel amaçlı optimizasyon algoritmaları büyük boyutlu optimizasyon problemlerini kısa bir sürede optimuma yakın çözümler veren algoritmalarlardır. Genel amaçlı sezgisel optimizasyon algoritmaları, fizik, biyoloji, sürü, sosyal, müzik ve kimya tabanlı olmak üzere altı farklı gruba ayrılırlar. Sürü tabanlı optimizasyon algoritmalarından “Wolf Colony Search Algorithm” inceleyeceğiz.

Özet

Kurt kolonisi algoritması kurt sürülerinin sıkı bir organizasyon sistemine sahip olmasından esinlenerek geliştirilmiştir. Sürüdeki kurtlar görev dağılımı yapmakta ve avlandıkları zaman tutarlı adımlar atmaktadırlar. Av aralarında bazı kurtlar arama kurtları olarak atanmakta ve avlarını buldukları zaman avın konumunu diğer kurtlara ulayarak bildirmektedirler.

Diğer kurtlar ava yaklaşmakta ve avı kuşatmaktadırlar. Kurt kolonisinin atanma kuralı, yiyeceğin ilk olarak en güçlü kurda ve daha sonra zayıf olan kurtlara verilmesidir.

(Liu Changan, Yan Xiaohu, Liu Chunyang and Wu Hua,)

Anahtar Kelimeler: Kurt Kolonisi Algoritması, Sürü Tabanlı Optimizasyon

1. Davranışların Açıklanması

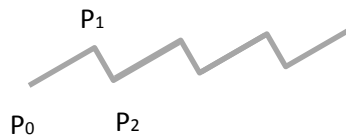
Arama uzayının boyutunun D olduğunu ve kolonideki birey sayısının n olduğunu varsayalım.

(i). yapay kurt pozisyonu X_i olursa;

$$X_i = (X_{i1}, \dots, X_{id}, \dots, X_{iD}) \quad 1 \leq i \leq n, 1 \leq d \leq D \quad (1)$$

1.1 Arama Davranışı

Kurtların av bulma şansını arttırmak için, q yapay kurtları av aralığında aramaya atanır.



Şekil 1. Arama Davranışı

Yapay kurt P0 üzerindeyken, P0 çevresinde h yönde h arama pozisyonları hesaplanmıştır ve optimal arama pozisyonu P1 seçilmiştir. Eğer P1 konumu P0 konumundan daha iyiyse P1 yeni konum olarak atanır ve yapay kurtlar P1 konumuna doğru hareket eder.

Q arama kurtlarını ava en yakın kurtlar olduğunu varsayarsak, maksimum arama sayısı $maxdh$ ve (i).arama yapay kurdunun pozisyonu XXi . H pozisyonları XXi çevresinde oluşturulmuş ve (j).arama pozisyonu Yj olursa ($1 \leq j \leq h$)

$$Yj=XXi+rand(n)*stepa \quad (2)$$

Burada $rand(n)$ [-1,1] aralığında rastgele eşit bir aralıkta dağıtılmış ve ($stepa$) ise adım sayısıdır. Eğer arama numarası $maxdh$ 'dan daha büyükse veya geçerli konum optimal arama pozisyonundan daha iyiyse arama davranışı biter.

1.2 Av Kuşatma

Arama kurtları avlarını buldukları zaman avın konumu diğer kurtlara uluyarak bildirirler. Burada diğer yapay kurtlar da ava yakın olsun ve onu kuşatsınlar. Varsayalım ki avın pozisyonu diğer arama yapay kurtlarından daha yakın olsun. (d). arama uzayında avın konumu (k).koloni iterasyonu G_d^k ve (i).yapay kurt pozisyonu X_{id}^k olursa;

$$X_{id}^{k+1} = X_{id}^k + rand * stepb * (G_d^k - X_{id}^k) \quad (3)$$
 ($rand$) [0,1] aralığında rastgele bir sayı, ($stepb$) kuşatma adımıdır ve (k) iterasyon sayısıdır.

(d). Pozisyon $[X_{mind}, X_{maxd}]$ 'dir. Eğer (3). Formülde hesaplanan değer aralığı aşarsa sınır değer olarak belirlenir.

(3).denklem iki bölümden oluşmaktadır. İlk bölüm yapay kurtların mevcut pozisyonu ve ikincisi ise Kurt kolonisinin optimal yapay kurda doğru yönelmesidir. Koloni arasında çalışma ve bilgi paylaşımı eğilim gösterir. Kurt kolonisinin avı kuşattığı gibi WCA ile global optimumu arayabilirsiniz.

1.3 Kurt Kolonisinin Güncellenmesi

Kurt sürüsünün atanma kuralı yiyecek önce güçlü olan kurda daha sonra ise zayıf kurtlara atanır. Burada güçlü olan kurda en fazla yiyecek atanır ve zayıf olan kurtlarda bir süre sonra açlıktan ölürlür. Bir süre sonra güçlü olan kurtlarda zayıf olacağından kurt kolonisinin adaptasyonu da arttırılmış olur. WCA algoritmasında koloni içindeki zayıf kurtlar kaldırılır ve yerine yeni kurtlar ürettirilirse bu durum bize çeşitlilik sağlar ve lokal optimumu önlemiş oluruz.

1.4 Kurt Koloni Algoritması Adımları

1.Adım

- Rastgele “ n ” sayısı
- Maksimum iterasyon sayısı “ $maxk$ ”
- Arama yapacak yapay kurt sayısı “ q ”
- Arama yönü “ h ”
- Maksimum arama numarası “ $maxdh$ ”
- Arama adımı “ $stepa$ ”
- Kuşatma Adımı “ $stepb$ ”
- Zayıf yapay kurt sayısı “ m ”
- (i). Yapay kurt için X_i değerleri belirlenir.

2.Adım

→ Yapay kurtlar içinden optimal yapay kurtları seçin ve her arama yapan yapay kurtları formül (2) göre hareket ettirin.

3.Adım

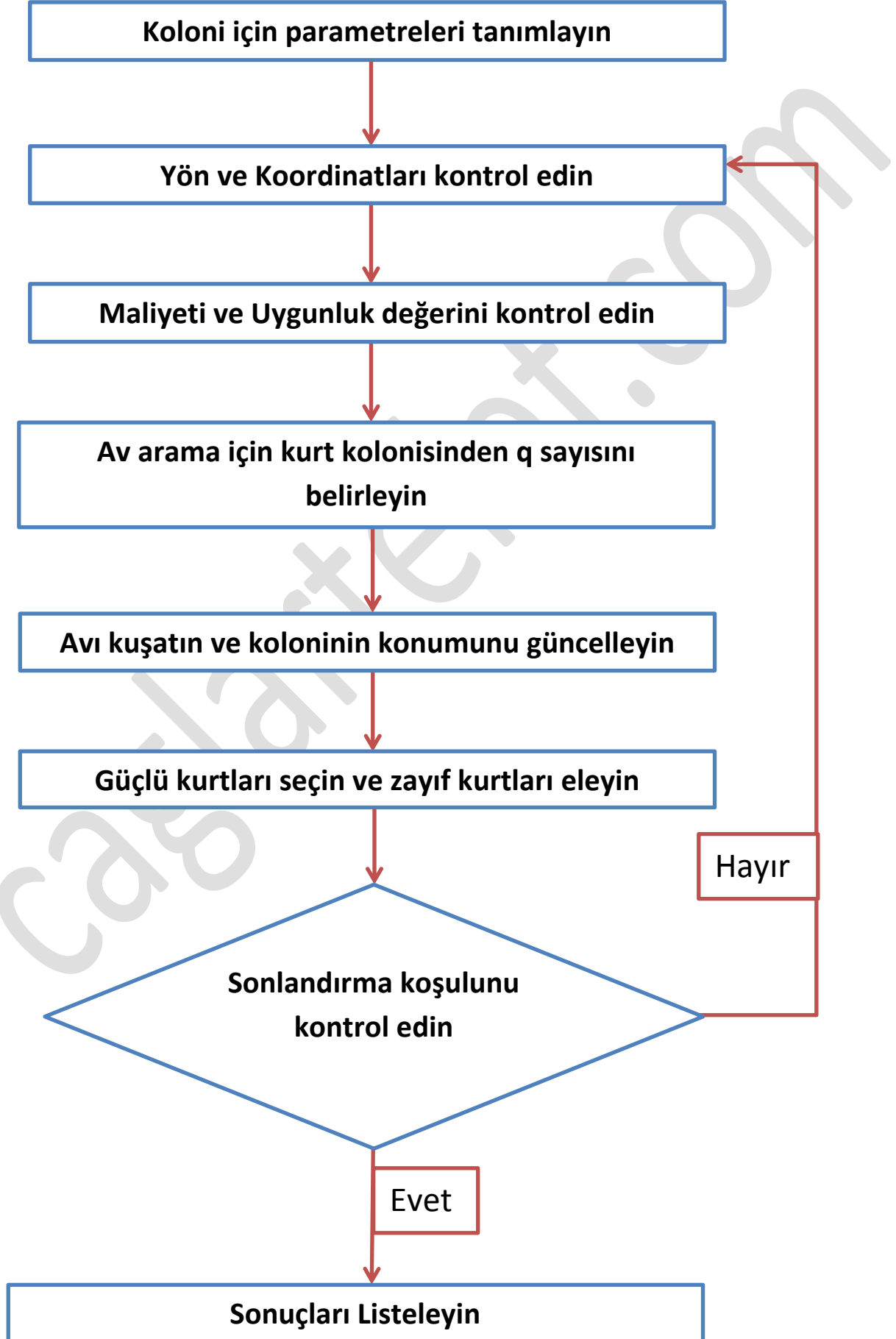
- Arama yapan yapay kurtlar içerisinde ava pozisyonuna göre en iyi pozisyonu seçin
- Tüm yapay kurtların pozisyonunu formül (3) e göre yeniden belirleyin.
- Eğer “ Xid ” değeri “ $Xmind$ ” değerinden küçükse “ Xid ” eşittir “ $Xmind$ ” olarak atanır.
- Eğer “ Xid ” değeri “ $Xmaxd$ ” değerinden büyükse “ Xid ” eşittir “ $Xmaxd$ ” olarak atanır.

4.Adım

→ Kurt kolonisini atanma kuralına göre güncelleyin. Zayıf kurtları koloniden çıkarın ve yeni kurtlar üretin.

5.Adım

→ Sonlandırma koşulunun son duruma ulaşıp ulaşmadığı kontrol edilir. WCA yineleme adım sayısı maksimum yineleme sayısına ulaşırsa, optimal çözüm yapay kurtların içinden probleme en iyi sonucu veren kurdun pozisyonudur. Eğer yineleme sayısı maksimum sayı değerine ulaşmadıysa 2. Adım’a dönülür ve döngü devam eder.



2. Wolf Colony Algorithm Performans Analizi

Parametrelerin seçimi optimizasyon algoritmalarının performansı için çok önemlidir. Wolf Colony Algorithm için önemli olan parametreler aşağıdaki gibidir:

-Kuşatma adımı “*stepb*”

-Zayıf kurt sayısı “*m*”

WCA için önemli parametreleri seçmek için aşağıdaki sürekli fonksiyon kullanılır.

$$f1(x) = -20 \exp \left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + 20 + e, \quad (4)$$
$$S = [-32, 32]^2,$$
$$fmin = 0$$

Fonksiyonun boyutu 30 olduğunda WCA için önemli parametreler farklıdır. Diğer parametreler ise $n=200$, $q=5$, $h=4$, $maxdh=15$, $steпа=1.5$, WCA için ortalama optimal değerler aşağıdaki (tablo 1)'de gösterilmiştir.

Tablo 1) Fonksiyon Sonuçları

stepb	m	Ortalama Değer	Optimal Değer
0.1	5	0.9702	5.8271e-005
5	5	0.0606	0.0086
0.9	50	0.1090	2.5731e-005
0.9	5	0.0373	9.5163e-007

Tablo-1'de en uygun değer birinci satırda bulunmaktadır. Çünkü “*stepb*” değer en küçük ve global optimal değeri buradan elde edilebilir. Burada *stepb* değer çok büyük olsaydı yapay kurtlar tarafından tespit edilecek olan pozisyonlar sınır değerini aşacak ve global optimal değeri arama yeteneği azalır. “*m*” için uygun değerler küçük olmalıdır. Aksi takdirde koloni içindeki değerler rastgele olarak değiştirilecek ve bundan dolayı yakınsama hızı ve arama hızı etkilenecektir.

Arama yapay kurt sayısı “*q*” ve arama yönü “*h*” çok büyük değerler seçilemez. Eğer bu değerler çok büyük seçilirse arama alanı çakışması ve hesaplama maliyeti artar.

Arama yapay kurt sayısı koloni sayısından küçüktür. “ $h*maxdh$ ” değeri “*n*” değerinden küçük olduğu varsayılırsa WCA algoritmasını zaman karmaşıklığı $O(n*maxk)$.

Yakınsama ve global arama yeteneğini teste etmek için aşağıdaki fonksiyonları kullanacağız.

➤ Sphere(Küre) Fonksiyonu:

$$f2(x) = \sum_{i=1}^n x_i^2 \quad S = [-100,100]^n, fmin = 0 \quad (5)$$

➤ Schwefel Fonksiyonu:

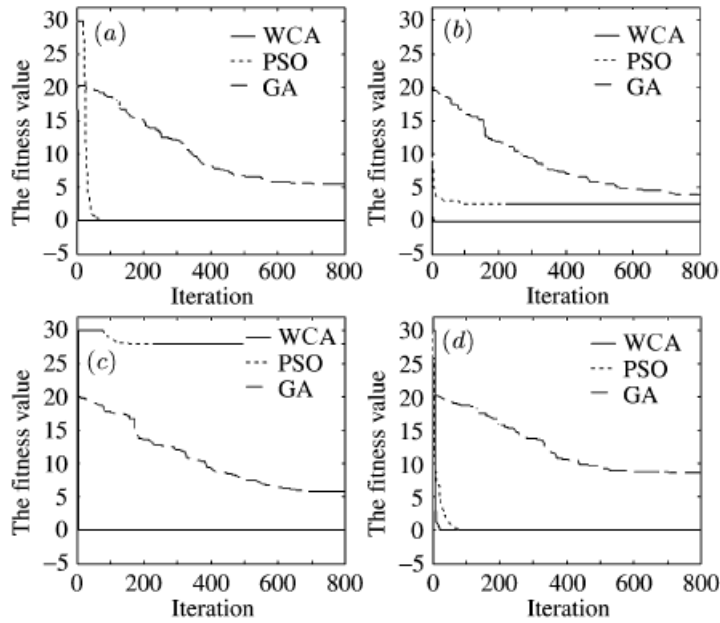
$$f3(x) = \max(|x_i|, 1 \leq i \leq n), S = [-100,100]^n, fmin = 0 \quad (6)$$

➤ Rosenbroc Fonksiyonu:

$$f4(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10] \quad S = [-5.12,5.12]^n, fmin = 0 \quad (7)$$

➤ Rastigri Fonksiyonu:

$$f5(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \quad S = [-600,600]^n, fmin = 0 \quad (8)$$



Şekil-3)Algoritmaların Yakınsama Eğrileri

(a)Sphere Algoritması, (b)Schwefel Algoritması, (c) Rosenbroc Algoritması, (d) Rastigri Algoritması

İlk iki fonksiyonda tek bir yerel optimum varken diğerlerinde iki veya daha fazladır. Üç algoritma karşılaştırılırsa, hepsinin yineleme sayısı 800. WCA için parametre değerleri $n=200$, $q=5$, $h=4$, $maxdh=15$, $stepa=1.5$, $stepb=0.9$, $m=5$. Pso için parçacık sayısı 200 ve etkileşim olmayan ağırlıklı değer ise 0.729 öğrenme katsayılarıysa 1.496. Çaprazlama ve mutasyon değerleri sırasıyla 0.9 ve 0.1'dir. Popülasyon nüfusu 600 ve bir bireyin yaşam uzunluğu 900. Tüm algoritmalar için aşağıdaki tabloda değerleri gösterilmiştir.

Tablo 2) Test Sonuçları

Fonksiyon		Sphere	Schwefel	Rosenbroc	Rastigri
Ortalama Değer	WCA	3.0815e-009	7.8201e-006	2.4866e-006	1.7545e-012
	PSO	0.0628	6.2732	51.6593	0.1408
Optimal Değer	GA	0.4125	0.3632	0.0011	14.3940
	WCA	2.7055e-011	1.8378e-007	5.7362e-008	1.1102e-016
	PSO	0.0024	3.7711	17.9819	0.0022
	GA	0.2443	0.1673	4.2302e-004	6.1172

Tablo-2 de görüldüğü gibi PSO ve GA yerel optimumda sıkışıp kalırlar.

3. Sonuçlar

WCA kurt davranışlarına dayalı bir algoritmadır. Kurt kolonisi birlikte yaşama ve işbirliği yoluyla global optimumu arar. WCA performansı optimizasyon problemlerine göre tartışılmıştır. WCA robotlar için yol planlamasında kullanılan bir algoritmadır.

1. Global arama yeteneği. Arama yapay kurtları ava en yakın konumu arar bu yüzden algoritmanın global arama yeteneği çok güçlüdür.
2. Basitlik. Algoritma basit fonksiyonlar kullanır ve uygulaması çok kolaydır.
3. Genelleme. Algoritmada birkaç değişiklik yapılarak istenilen problemlerde kullanılabilir.

Bu nedenle WCA algoritması optimizasyon problemlerinde kullanmak için uygundur.

Kaynakça

- A Wolf Colony Search Algorithm Based on the Complex Method for Uninhabited Combat Air Vehicle Path Planning, **Qiang Zhou, Yongquan Zhou, Xin Chen**
- The Wolf Colony Algorithm and Its Application, **LIU Changan, YAN Xiaohu, LIU Chunyang and WU Hua**
- Güncel Sürü Zekâsı Optimizasyon Algoritmaları, **Sinem AKYOL ve Bilal ALATAŞ**

CağlarTelef.Com