**YZM 4032 Meta-Sezgisel Optimizasyon**

**2019 Bahar Dönemi Ödevi**

**Crow Search Algorithm**

**Kaynak Makale Adı:**

**A novel metaheuristic method for solving constrained engineering**

**optimization problems: Crow search algorithm**

**Grup No:8**

**334289- Canan GÖK**

**334311- Oğuz Kağan KIZIL**

**Ders Sorumlusu: Doç. Dr. Hamdi Tolga KAHRAMAN**

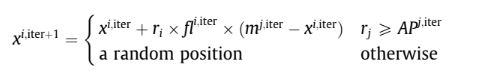
**Nisan 2019**

***Crow Search Algorithm ( CSA )***

Bu yazıda meta-sezgisel arama yöntemlerinden CSA’nın arama sürecinin geliştirilmesi için uygulanan yöntemlerden bahsedilmektedir.

***1- Arama Süreci***

CSA arama süreci; karga i’nin rastgele seçilen bir karga j’yi takip ederek karga j’nin yiyeceklerini sakladığı yere ulaşması üzerine kurulmuştur. Bu sırada, eğer karga j takip edildiğini fark eder ise çevrede rastgele bir konuma giderek takipçisini kandırır. Arama sürecinin formülü *şekil 1*’de gösterilmiştir.



(a)

(b)

***Şekil 1. – Arama Süreci***

Bu formülde kullanılan ***xi,iter*** takip eden karganın mevcut konumunu, ***ri***0 ve 1 arasında rastgele oluşturulan bir sayıyı*,* ***fl i,iter*** algoritmaya özeluçuş uzunluğu değerini ve ***mj,iter*** ise takip etmek için rastgele seçilen kargayı temsil etmektedir. Arama süreci yaşam döngüsünde farkındalık olasılığının (AP) düşük değerleri komşuluk aramasını *(Şekil 1(a))*, büyük değerleri ise çeşitliliği *(Şekil 1 (b))* sağlamaktadır. Asıl algoritmada ***AP*** değeri ***0.1*** olarak kullanılmaktadır.

***2- Uygulanan Geliştirmeler***

Arama yaşam sürecini geliştirmek için FDB seçim yönteminden yararlanılarak çeşitli modifikasyon ve mutasyonlar denenmiştir. FDB seçim yöntemi; topluluk içerisinde uygunluk değeri en yüksek olan çözüm adayını belirten bir yöntemdir. Bu yöntem kullanılarak uygulanan senaryolar aşağıdaki gibidir.

***2.1 – Komşuluk Araması***



***Şekil 2. – Komşuluk Araması***

*Senaryo 1* – FDB Seçim Yöntemi

Bu senaryoda komşuluk araması sırasında karga i’nin takip etmek için rastgele seçtiği karga j (***mj,iter***) yerine, FDB seçim yöntemi uygulanarak seçilmiş, topluluk içerisinde uygunluk değeri en yüksek olan karga j kullanılmıştır.

*Senaryo 2* – Yarı yarıya seçim

Bu senaryoda komşuluk araması işleminde iterasyon sayısı tek ise; karga i’nin takip etmek için rastgele seçtiği karga j yerine, FDB seçim yöntemi uygulanarak seçilmiş, topluluk içerisinde uygunluk değeri en yüksek olan karga j kullanılmıştır. İterasyon sayısı çift ise takip etmek için rastgele seçilen karga j kullanılarak yarı yarıya seçim uygulanmıştır.

*Senaryo 3* – FDB Komşuluk Mutasyonu

Bu senaryoda komşuluk araması sırasında karga i’nin takip etmek için rastgele seçtiği karga j yerine, FDB yöntemi uygulanarak seçilmiş çözüm adayının değişkenlerinde mutasyon yapılarak sağlanmıştır. Mutasyon işlemi, değişkenlerin %30’una denk gelecek sayıda birbirinden farklı olarak seçilen değişkenler yerine, sınırlar içerisinde rastgele oluşturulan sayılar kullanılarak sağlanmıştır.

***2.2 – Çeşitlilik***

*Senaryo 4* – FDB Çeşitlilik Mutasyonu

Bu senaryoda çeşitlilik, sınırlar içerisindeki rastgele bir pozisyona giderek sağlanırken bunun yerine FDB yöntemi uygulanarak seçilmiş çözüm adayının değişkenlerinde mutasyon yapılarak sağlanmıştır. Mutasyon işlemi, değişkenlerin %30’una denk gelecek sayıda birbirinden farklı olarak seçilen değişkenler yerine, sınırlar içerisinde rastgele oluşturulan sayılar kullanılarak sağlanmıştır.

***3- Senaryo Sonuçları***

Yukarıda açıklanan senaryolar 30-50-100 boyutlarında 51 koşum yapılarak test edildi ve elde edilen ilk sonuçlarda, komşuluk aramasında FDB yöntemi kullanan 1 ve 2 numaralı senaryolarda başarılı sonuçlar verince en uygun değeri bulmak için bu senaryolar üzerinden yeni test senaryoları oluşturuldu. Mutasyon yapılan 3 ve 4 numaralı senaryolar herhangi bir problemde daha iyi bir sonuç vermeyi başaramadığı için ilerleyen aşamalarda bu senaryolara yer verilmedi.

Senaryo 1 komşuluk aramasında sadece FDB kullanırken, senaryo 2 %50 FDB kullanarak ilk testlerde en iyi sonuç veren senaryo olmayı başardı. Tek-çift sayı yöntemi kullanılarak sıralı yapılan %50 seçimde, seçim sırasının etkisini görmek için maksimum iterasyon sayısının ilk yarısında FDB ile seçim yapıp sonraki yarısında rastgele seçim yapılarak test edildi. Aynı test, ilk yarının rastgele seçim yöntemi ile ve ikinci yarının FDB seçimi ile yapılması ile tekrar edildi. Bu ön testler 30 boyutta 21 koşum yapılarak oluşturuldu. Test sonuçlarında sıralı seçimin en iyi sonucu verdiği görüldü. (*bkz. Tablo 1*)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Run: 21 | | | | | |
| **Problem Havuzu** | **Boyut** | **Senaryo** | İYİ | EŞİT | KÖTÜ |
| KLASİK | 30 | **%50 sıralı** | **17** | **4** | **9** |
| **ilk %50 FDB** | **10** | **15** | **5** |
| **son %50 FDB** | **7** | **11** | **12** |
| CEC2014 | **%50 sıralı** | **5** | **14** | **11** |
| **ilk %50 FDB** | **3** | **20** | **7** |
| **son %50 FDB** | **3** | **9** | **18** |
| CEC2017 | **%50 sıralı** | **8** | **7** | **15** |
| **ilk %50 FDB** | **4** | **20** | **6** |
| **son %50 FDB** | **2** | **9** | **19** |

***Tablo 1.***

Bu bilgiler ışığında komşuluk aramasını geliştirebilecek, FDB kullanımının en iyi oranını bulmak için %70, %50, %30, %25, %20 ve %10 olduğu yeni senaryolar üretildi. Senaryolar 30-50-100 boyutlarda ve 51 koşum yapılarak çalıştırıldı.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem Havuzu** | **Boyut** | **Senaryo** | İYİ | EŞİT | KÖTÜ |
| KLASİK | 30 | CASE1 (%50) | 18 | 2 | 10 |
| CASE2 (%70) | 17 | 3 | 10 |
| CASE3 (%10) | 15 | 10 | 5 |
| CASE4 (%20) | 15 | 7 | 8 |
| CASE5 (%25) | 17 | 6 | 7 |
| CASE6 (%30) | 17 | 5 | 8 |
| 50 | CASE1 (%50) | 18 | 3 | 9 |
| CASE2 (%70) | 18 | 0 | 12 |
| CASE3 (%10) | 16 | 11 | 3 |
| CASE4 (%20) | 17 | 5 | 8 |
| CASE5 (%25) | 17 | 5 | 8 |
| CASE6 (%30) | 17 | 5 | 8 |
| 100 | CASE1 (%50) | 17 | 2 | 11 |
| CASE2 (%70) | 17 | 0 | 13 |
| CASE3 (%10) | 18 | 10 | 2 |
| CASE4 (%20) | 18 | 3 | 9 |
| CASE5 (%25) | 18 | 4 | 8 |
| CASE6 (%30) | 18 | 4 | 8 |

***Tablo 2. (Klasik Benchmark)***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem Havuzu** | **Boyut** | **Senaryo** | İYİ | EŞİT | KÖTÜ |
| CEC2014 | 30 | CASE1 (%50) | 8 | 10 | 12 |
| CASE2 (%70) | 8 | 9 | 13 |
| CASE3 (%10) | 5 | 23 | 2 |
| CASE4 (%20) | 7 | 15 | 8 |
| CASE5 (%25) | 8 | 12 | 10 |
| CASE6 (%30) | 7 | 13 | 10 |
| 50 | CASE1 (%50) | 13 | 5 | 12 |
| CASE2 (%70) | 12 | 6 | 12 |
| CASE3 (%10) | 12 | 16 | 2 |
| CASE4 (%20) | 14 | 8 | 8 |
| CASE5 (%25) | 13 | 7 | 10 |
| CASE6 (%30) | 14 | 5 | 11 |
| 100 | CASE1 (%50) | 14 | 2 | 14 |
| CASE2 (%70) | 14 | 3 | 13 |
| CASE3 (%10) | 15 | 12 | 3 |
| CASE4 (%20) | 16 | 7 | 7 |
| CASE5 (%25) | 16 | 6 | 8 |
| CASE6 (%30) | 14 | 4 | 12 |

***Tablo 3. (CEC 2014 Benchmark)***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem Havuzu** | **Boyut** | **Senaryo** | İYİ | EŞİT | KÖTÜ |
| CEC2017 | 30 | CASE1 (%50) | 11 | 5 | 14 |
| CASE2 (%70) | 10 | 5 | 15 |
| CASE3 (%10) | 7 | 18 | 5 |
| CASE4 (%20) | 8 | 8 | 14 |
| CASE5 (%25) | 9 | 9 | 12 |
| CASE6 (%30) | 10 | 6 | 14 |
| 50 | CASE1 (%50) | 13 | 2 | 15 |
| CASE2 (%70) | 13 | 1 | 16 |
| CASE3 (%10) | 9 | 16 | 5 |
| CASE4 (%20) | 11 | 8 | 11 |
| CASE5 (%25) | 11 | 8 | 11 |
| CASE6 (%30) | 12 | 7 | 11 |
| 100 | CASE1 (%50) | 12 | 7 | 11 |
| CASE2 (%70) | 11 | 5 | 14 |
| CASE3 (%10) | 8 | 19 | 3 |
| CASE4 (%20) | 12 | 15 | 3 |
| CASE5 (%25) | 12 | 14 | 4 |
| CASE6 (%30) | 12 | 8 | 10 |

***Tablo 4. (CEC 2017 Benchmark)***

Test sonuçlarında görüldüğü üzere algoritma, problem boyutu büyüdükçe daha iyi sonuç verme eğiliminde. İyi sonuç-kötü sonuç farkına bakarak bir seçim yaparsak en başarılı oranlar *Tablo 5*’de gösterildiği gibidir.

|  |  |  |
| --- | --- | --- |
| **AP = 0.1** | | |
| **Problem Havuzu** | **Boyut** | **En Başarılı Senaryo** |
| **Klasik** | **30** | **CASE5 (25%)** |
| **50** | **CASE1 (10%)** |
| **100** | **CASE3 (10%)** |
| **CEC14** | **30** | **CASE3 (10%)** |
| **50** | **CASE3 (10%)** |
| **100** | **CASE3 (10%)** |
| **CEC17** | **30** | **CASE3 (10%)** |
| **50** | **CASE3 (10%)** |
| **100** | **CASE4 (20%)** |

***Tablo 5. (İyi-Kötü farkına göre başarılı senaryolar)***

Alınan sonuç üzerinde 10% - 25% oranları aralığında FDB kullanımının algoritmayı büyük ölçüde ileriye taşıdığı görüldü. Yine de CEC problemlerindeki başarı klasik havuzuna göre geride kalınca bu sonuçları geliştirmek adına, 0.1 olan ve %90 komşuluk aramasına yönlendiren farkındalık olasılığı (***AP***) değerinin algoritmaya etkisini ölçmek için 30 boyutta ve 21 koşumda bir ön test yapıldı. Bu ön testte CSA’da ve tüm oluşturduğumuz senaryolarda *AP=0.5* yapıldı. Ek olarak CSA’nın kendi içindeki performansı için *AP=0.1* ve *AP=0.2* değerli temel CSA ile karşılaştırıldı. (*bkz. Tablo 6*)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Run: 21 AP = 0.5 | | | | | |
| **Problem Havuzu** | **Boyut** | **Senaryo** | İYİ | EŞİT | KÖTÜ |
| KLASİK | 30 | **csa\_ap=0.1** | **27** | **2** | **1** |
| **csa\_ap=0.2** | **27** | **3** | **0** |
| CEC 2014 | **csa\_ap=0.1** | **28** | **0** | **2** |
| **csa\_ap=0.2** | **25** | **4** | **1** |
| CEC 2017 | **csa\_ap=0.1** | **18** | **9** | **3** |
| **csa\_ap=0.2** | **20** | **10** | **0** |

***Tablo 6. (CSA’da AP değeri etkisi)***

Orijinal CSA’da AP değeri 0.5 yapıldığında, algoritmanın AP değerleri 0.1 ve 0.2 olduğundaki sonuçlarından daha kötü olduğu gözlemlense de (*bkz. Tablo 6*) yapılan FDB modifikasyonları AP=0.5 değerinde çeşitliliğe katkı sağlayarak çok daha başarılı sonuçlar üretmeyi başarmıştır. Algoritmanın yüksek boyutta daha iyi performans sergilediği bilindiği için AP=0.5 yapılarak tüm testler 30-50-100 boyutlar için 51 koşumda tekrarlandı. Testlerin detaylı sonuçları aşağıda bulunmaktadır.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem Havuzu** | **Boyut** | **Senaryo** | İYİ | EŞİT | KÖTÜ |
| KLASİK | 30 | CASE1 (%50) | 21 | 4 | 5 |
| CASE2 (%70) | 21 | 3 | 6 |
| CASE3 (%10) | 20 | 9 | 1 |
| CASE4 (%20) | 20 | 7 | 3 |
| CASE5 (%25) | 20 | 5 | 5 |
| CASE6 (%30) | 20 | 4 | 6 |
| 50 | CASE1 (%50) | 20 | 2 | 8 |
| CASE2 (%70) | 19 | 3 | 8 |
| CASE3 (%10) | 22 | 7 | 1 |
| CASE4 (%20) | 21 | 3 | 6 |
| CASE5 (%25) | 20 | 4 | 6 |
| CASE6 (%30) | 20 | 3 | 7 |
| 100 | CASE1 (%50) | 19 | 2 | 9 |
| CASE2 (%70) | 19 | 0 | 11 |
| CASE3 (%10) | 21 | 7 | 2 |
| CASE4 (%20) | 21 | 3 | 6 |
| CASE5 (%25) | 19 | 5 | 6 |
| CASE6 (%30) | 19 | 4 | 7 |

***Tablo 7. (Klasik Benchmark AP=0.5)***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem Havuzu** | **Boyut** | **Senaryo** | İYİ | EŞİT | KÖTÜ |
| CEC2014 | 30 | CASE1 (%50) | 16 | 5 | 9 |
| CASE2 (%70) | 14 | 8 | 8 |
| CASE3 (%10) | 11 | 18 | 1 |
| CASE4 (%20) | 14 | 9 | 7 |
| CASE5 (%25) | 10 | 12 | 8 |
| CASE6 (%30) | 14 | 9 | 7 |
| 50 | CASE1 (%50) | 18 | 4 | 8 |
| CASE2 (%70) | 17 | 5 | 8 |
| CASE3 (%10) | 11 | 19 | 0 |
| CASE4 (%20) | 14 | 10 | 6 |
| CASE5 (%25) | 17 | 5 | 8 |
| CASE6 (%30) | 18 | 4 | 8 |
| 100 | CASE1 (%50) | 19 | 2 | 9 |
| CASE2 (%70) | 20 | 2 | 8 |
| CASE3 (%10) | 18 | 10 | 2 |
| CASE4 (%20) | 20 | 5 | 5 |
| CASE5 (%25) | 19 | 3 | 8 |
| CASE6 (%30) | 20 | 1 | 9 |

***Tablo 8. (CEC 2014 Benchmark AP=0.5)***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Problem Havuzu** | **Boyut** | **Senaryo** | İYİ | EŞİT | KÖTÜ |
| CEC2017 | 30 | CASE1 (%50) | 12 | 4 | 14 |
| CASE2 (%70) | 11 | 4 | 15 |
| CASE3 (%10) | 10 | 20 | 0 |
| CASE4 (%20) | 11 | 8 | 11 |
| CASE5 (%25) | 11 | 9 | 10 |
| CASE6 (%30) | 11 | 5 | 14 |
| 50 | CASE1 (%50) | 12 | 2 | 16 |
| CASE2 (%70) | 12 | 2 | 16 |
| CASE3 (%10) | 10 | 18 | 2 |
| CASE4 (%20) | 11 | 6 | 13 |
| CASE5 (%25) | 11 | 5 | 14 |
| CASE6 (%30) | 11 | 5 | 14 |
| 100 | CASE1 (%50) | 13 | 2 | 15 |
| CASE2 (%70) | 14 | 0 | 16 |
| CASE3 (%10) | 14 | 14 | 2 |
| CASE4 (%20) | 13 | 7 | 10 |
| CASE5 (%25) | 13 | 4 | 13 |
| CASE6 (%30) | 13 | 3 | 14 |

***Tablo 9. (CEC 2017 Benchmark AP=0.5)***

Sonuçlarda görüldüğü üzere AP değeri 0.5 yapıldığında çeşitlilik oranımızın artması ile sonuçlarda büyük oranda bir gelişme yaşandı. %10 oranında FDB kullandığımız CASE3, algoritmayı en ileri taşıyan senaryomuz oldu.