**YZM 4008**

**Veri Madenciliği Dersi Proje Raporu**

**Algoritma Adı:** Dynamic differential annealed optimization

**Öğrenci Adı ve Soyadı:** Elif Şahinkaya

**Öğrenci No:**334341

**Öğrenci İletişim Bilgileri**

**Cep tel:** -

**\*e-mail adresi:** elifsahinkaya@outlook.com

**Knn Algoritmasının DDAO Meta-Sezgisel Arama Algoritması Kullanılarak**

**Meta Sezgisel-Knn Algoritmasına Çevrilmesi**

**ÖZET**

Bu yaklaşım geniş bir yelpazeyi küresel minimum veya maksimumun gerekli olduğu matematiksel optimizasyon problemleri çözmek için kullanılabilecek yeni bir optimizasyon algoritması önermektedir.Yeni algoritma, rastgele aramaya ve klasik benzetilmiş tavlama algoritmasına dayanmaktadır (Modern yüksek kaliteli çelik üretim süreci) ve dinamik diferansiyel tavlı optimizasyon (DDAO) olarak belirlenmiştir.Dinamik diferansiyel tavlı optimizasyon algoritması, çok sayıda yüksek alıntı yapılan optimizasyon algoritmaları ile karşılaştırılmıştır.Sayısal testlere göre, DDAO, bu algoritmalardan bazılarından daha iyi performans göstermiştir. Birçok durumda yüksek performans göstermiştir.Kısıtlı yol planlaması ve yay tasarımı problemi pratik bir mühendislik optimizasyon problemi olarak seçilmiştir. DDAO, küresel minimum olan sorunlarına ve yay tasarım problemi için birçok algoritmadan daha verimli çözümler üretmiştir.

# GİRİŞ

Mekanik iyileştirme için birçok çalışma yapılmıştır.Çeliğin özellikleri gerekli kütleyi azaltmak için üretim süreci için maliyeti etkiler.Bu çalışmalar, yüksek mukavemetli çeliklerin geliştirilmesi ile ilgilidir,gelişmiş yüksek olarak belirlenen önemli oluşum oranlarıyla dayanıklı çelikler . Çift fazlı çelik bir tür gelişmiş benzersiz bir mikro yapıya sahip yüksek mukavemetli çelikler süper mekanik özelliklerdir. Genel olarak, yüksek bir sıcaklıkta demir erir sıvı hale ve sürekli soğutma ile katı hale oda sıcaklığında duruma gelir. Metalin özellikleri bağlı olduğu katı haldeki mevcut fazlar hakkında: örneğin, ferritik faz esneklik verir, martensit fazı sertlikten sorumludur ve bainit ve östenit gibi başka aşamalar da vardır. Genellikle, birçok demir fazı aynı katıda bir arada bulunur ve nihai mekanik özelliğe bir istenen kombinasyonu verirler. Çift fazlı çelik (DP) üretmek için bir parça demir mikro yapının yaklaşık olarak homojen olduğu yeniden kristalleşme noktasının üzerinde yüksek bir sıcaklığa ısıtılır.Bir süre sonra demir parçası daha ince bir kalınlıkta haddelenmelidir ve haddeleme sırasında metal soğuyacaktır. Sonra mekanik deformasyon uygulanarak metal müteakip diferansiyel soğutmaya tabi tutulur; hızlandırılmış soğutma, hava soğutma, başka bir hızlandırılmış soğutma ve son olarak oda sıcaklığına yavaş soğutma. Bu süreç, bir mikro yapıya sahip çeliğe yol açar. Adalar şeklinde sert martensitik faz içeren ferritik matris. Yumuşak ferrit ve sertin birleşik etkilerimartenzit, çift fazlı çeliğe kabul edilebilir bir oluşum hızı ile yüksek mukavemet verir ,bu özellikle otomotiv endüstrisinde çok önemlidir. Daha önce açıklanan sistematik prosedür, bir yüksek kaliteli çelik tipi (çift fazlı) net bir optimization. Metal, düşük kaliteden yüksek kaliteye doğru geliştirilir. Bu davranış, dinamik farklılığa ilham verdi. Bu çalışmada geliştirilen tavlı optimizasyon algoritmasıdır.

# DDAO Algoritması

**1.** Çeliğin kütlesi başlangıçta molekül gruplarından oluşur iyileştirilmesi gereken (aday çözümlerin nüfusu) bir ferrit ve martensit karışımı (küresel çözüm) olmalıdır.

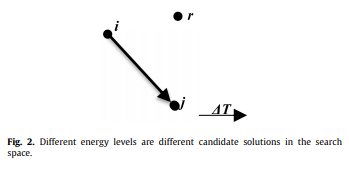
**2**. Çift fazlı çelik üretiminde sıcaklık düşer ve sıcaklıktaki her düşüş için farklı bir çelik fazı oluşturma şansı vardır. Bu, küresel bir çözüm ararken matematiksel optimizasyondaki yineleme sürecine eşdeğerdir.

**3.** Metaldeki her fazın kendi iç enerjisi vardır ve bu matematiksel optimizasyonun amaç fonksiyonunun değerine eşdeğer olur.

**4.** Şekil 1 farklı soğutma hızlarını, havayla soğutmayı, hızlandırılmış soğutmayı ve yavaş soğutmayı gösterir; soğutma işlemini temsil etmek için aşağıdaki denklemi önerilir:

 (1)

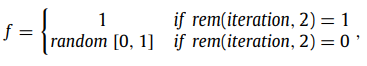
burada S k yineleme numarası ( k ) için önerilen yeni bir çözümdür , k = 1 ... n burada n yineleme sayısıdır ve Sc i ve Sc j , rastgele seçilen popülasyondan rastgele seçilen çözümlerdir. ( i ) ve ( j ) endeksleri içinde , Sr rastgele oluşturulmuş bir çözümdür. Daha açıklamak gerekirse, şek. 2, sıcaklık değiştiğinde I noktasından j noktasına √ T ile varır.

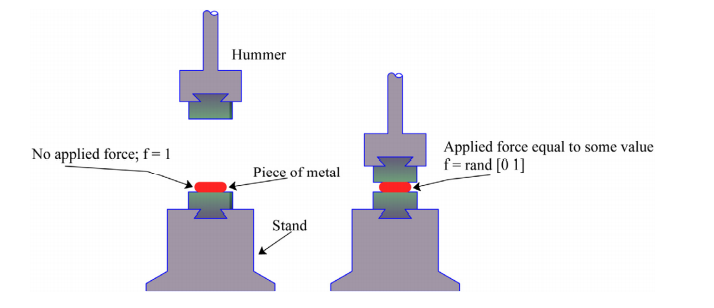


Her birinin enerji seviyesi, nesnel değere sahip bir çözüme eşdeğerdir başka bir deyişle, iki rastgele enerji seviyesi arasındaki fark bir çözüm de vardır. Ayrıca, bu farkı rastgele ekleyerek uzayda enerji, şek. 2, yeni bir değer dönecektir farklı enerjiye sahip bir çözüm olan enerji (nesnel değer).

Bu ekolayzır için hayali bir temeldir. (1), omurgası olan önerilen algoritmanın arama motoru için ve ana yakınsama için optimizasyon problemleri sorumludur. Buna değer, Ekolayzırdan bahsetmişken. (1) ve tamamlayıcı denklemi (3) diferansiyel evrimde belirtilen mutasyon sürecine benzer , ancak bizim durumumuzda evrim yoktur, random vardır seçim ve hatta SR popülasyondan rastgele seçilir başka bir alt popülasyondan, daha sonra açıklayacağımız gibi.

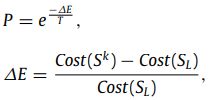
**5.** Isının diferansiyel indirgenmesi sırasında metal yuvarlanır ve bu mekanik işlem matematiksel olarak modellenmelidir. Programlama nedenleriyle, metalin haddeleme yerine dövme işlemine tabi tutulduğunu varsaymalıyız. Dövme sırasında çekicin dinamik davranışı, 1 ile rastgele bir sayı arasında dalgalanan bir parametre olarak temsil edilebilir.

(2)

burada f, dövme parametresidir ve rem, 2'ye bölme işleminden sonra kalandır. Yinelemeler sırasında, denklem (2), geçerli yineleme sayısı tek olduğunda, yani 1, 3, 5..vb. f bir olacak, mevcut iterasyon sayısı çift ise (2,4,6…vb.) f sıfır ile bir arasında rastgele bir sayı olacaktır. Bu, hummer tarafından uygulanan kuvvetin değişiklik olmaması ile rastgele bir değer arasında dalgalandığı Şekil 2’de gösterilen gerçek hayatta dövme ile tutarlıdır. 

Dövme, diferansiyel soğutma ile çevrim içi olarak yapıldığından, denklem (1) şu şekilde değiştirilir:(3)

**6.** Gerçek tavlama işleminde, düşük sıcaklıklarda olduğundan daha yüksek sıcaklıklarda yeni fazların oluşumunu kabul etme olasılığı daha yüksektir. Optimizasyon sürecinde, açıklanan olasılık formülüne bağlı olarak aynı prosedürü kullanıyoruz. SA algoritmasıyla:

(4) (5) 

P, yeni bir çözümü kabul etme olasılığı olduğunda, ΔE denklem (3)'ten önerilen çözümün amaç değeri ile popülasyondaki L indeksinin bir çözümü olan SL çözümünün amaç değeri arasındaki farktır, L =1,…, popülasyon büyüklüğü. T, yüksek değerle başlaması ve yinelemeler sırasında sürekli olarak daha düşük bir değere güncellenmesi gereken sıcaklık değişkenidir.

Önerilen çözüm, P > rasgele ∈ [0,1] ise kabul edilebilir. Aramanın başında T, yüksek değerle başlar; sonuç olarak, denklem (4)'e göre P bire yakın olacaktır. Bu, geniş bir rasgele sayı aralığının birden az olabileceği ve çözümün seçileceği anlamına gelir. T'nin düşük değerinde, P olasılığı sıfıra yakın olacaktır; denklem (4)'e göre,bu, çok dar bir rastgele sayı aralığının P'den daha az olabileceği ve çözümün seçilme olasılığının daha düşük olduğu anlamına gelir. Örneğin, ,  iken, bu, sıcaklık düştükçe yeni bir çözüm seçme olasılığını azaltmak için basit bir mekanizmadır.

**7.** İşlem 4. adımdan itibaren tekrarlanır ve her iterasyon için en iyi çözüm saklanır.

Ana arama motoru rastgele arama iken, dinamik tavlama işlemi arama üzerinde bir düzeltmedir; bu yüzden dinamik diferansiyel tavlanmış optimizasyon olarak adlandırılmıştır. Dinamik diferansiyel uygulamak için bir MATLAB platformu kullanılmıştır.

DDAO algoritmasının sözde kodu:

***X i popülasyonunu başlat (i=1,2,…,n)***

***T parametresini başlat, soğutma hızı***

***Her çözümün maliyetini hesaplayın***

***X b = En iyi çözüm***

***while (t<Max yineleme)***

***Alt popülasyon S'yi başlat***

***Alt popülasyonun maliyetini hesaplayın***

***Alt popülasyonu sırala***

***S r = Alt popülasyondaki en iyi çözüm***

***Popülasyondan iki rastgele çözüm X m ve X n seçin***

***(3) denkleminden S k hesaplayın***

***Popülasyon X'i sırala***

***X popülasyonunda foreach çözümü***

***bir gelişme varsa***

***X ben = Sk***

***aksi takdirde, denklem (4) ve (5)'i kullanarak X popülasyonundaki en kötü çözümü değiştirin***

***endif***

***son***

***X b'yi güncelle***

***T=T\*soğutma hızı***

***t=t+1***

***son zaman***

***dönüş X b***

**f**

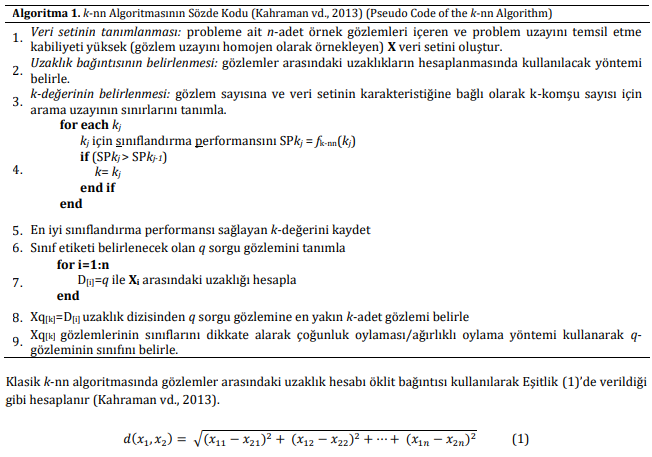
****

** **

** **

# K-en Yakın Komşu Algoritması (*k-NearestNeigbourAlgorithm*)

k-nn, örnek tabanlı bir sınıflandırma algoritmasıdır. Algoritmanın temeli, sınıfı belirlenecek olan bir gözlem için problem uzayını temsil eden örnek gözlemler arasından benzer özelliklere sahip olanların referans alınmasına dayanır. k-nn sınıflandırıcı bir öğrenme yöntemi içermediği gibi probleme ait bir model de geliştirmez. k-nn sınıflandırma sürecinin ilk adımlarından biri k-değerinin belirlenmesidir. k değeri için en uygun değerin belirlenmesi zor bir görevdir. Çünkü k-değerinin belirlenmesinde kullanılan bir matematiksel bir eşitlik bulunmamaktadır. Üstelik k-değeri probleme bağlı olarak değiştiği kadar örnek gözlemleri içeren veri setindeki gözlem sayısına ve veri setinin oluşturulmasında izlenen yola bağlı olarakta değişir. Algoritmanın performansı üzerinde etkili ola bir başka öğe de gözlemler arasındaki uzaklıkların hesaplanmasında kullanılan uzaklık metrikleridir. Hızlı ve etkili oldukları için bu süreçte Öklit, Manhattan ve Minkowski gibi uzaklık bağıntıları sıklıkla kullanılmaktadır. Bunun yanında bulanık uzaklık metriği klasik metriklere kıyasla çok daha etkili sonuçlar vermektedir. Bulanık uzaklık metriğinin ortalama olarak k-nn algoritmasının sınıflandırma performansını klasik metriklere göre %10’dan fazla iyileştirdiği bilinmektedir (Kahraman, 2016; Kahraman vd., 2013; Yilmaz vd., 2018).



Eşitlik (1)’de sınıflandırma probleminin nitelik sayısı n-ile, X1 ve X2 gözlemleri ise ve ile temsil edilmektedir. Buna göre uzaklık hesabında problemin niteliklerinin eşit derecede etkiye sahip olduğu görülmektedir. Bunun yanında her bir niteliğin alt ve üst sınır değerlerine bağlı olarak sayısal büyüklüklerinin de uzaklık hesabı üzerinde farklı bir etki yaratabileceği dikkate alınmalıdır. Bu durum niteliklerin birbirlerine baskınlık kurması olarak nitelendirilir. Baskınlık durumunu bertaraf etmek içi her bir niteliğin normalizasyonu [0, 1] arasında ölçeklendirilmesi mümkündür. Ancak bu durumda ise probleme ait niteliklerin, gözlemlerin sınıflandırılmasında eşit derecede etkiye sahip olmaları gibi bir durum ortaya çıkar. Örneğin bir müşteriye verilecek banka kredisi hesaplanırken kişinin mesleği, aylık geliri, kredi skoru, borcu, cinsiyeti, yaşı, yaşadığı şehir, varlıkları gibi bilgiler dikkate alınabilir. Ancak bu niteliklerin her biri kişiye verilecek kredinin miktarının belirlenmesinde farklı derecede öneme sahiptir. Dolayısıyla klasik k-nn algoritmasının işleyişi incelendiğinde, sınıflandırma sürecinde niteliklerin etkilerinin eşit olduğunun kabul edilmesi doğru ve etkili bir yaklaşım değildir. Bu nedenle niteliklerin önem/etki derecelerinin belirlenmesi için sezgisel arama algoritmalarının kullanıldığı yöntemler geliştirilmiştir. En iyi sınıflandırma performansının elde edildiği ağırlık değerlerinin arandığı bu yöntemler literatürde oldukça yaygın bir şekilde kullanılmaktadırlar.

Klasik k-nn de oklit bağıntısı ile uzaklık hesabı aşağıda verildiği gibidir:

*EU Metric:* 

Sezgisel k-nn’de iseoklit bağıntısı ile uzaklık hesabı aşağıda verildiği gibidir:

*EU Metric:* 

# Kullanılan Yöntem :DDAO-knn ile Sezgisel Nitelik Ağırlıklandırması

Bu yöntemle veri setindeki niteliklere ağırlık katsayısı vererek hangi niteliklerin diğerlerine göre daha etkin nitelikler olduğunu bulmayı amaçlıyoruz.Popülasyonun oluşturulmasıyla başlayan yöntem sınır değerler arasında rastgele çözümler üretir. Bu problemdeki ağırlık sınırlarımız [-1,1] olacaktır. DDAO işlemleri boyunca tüm çözüm adayları amaç fonksiyona gönderilir.



**4.1.**Deneysel Sonuçlar (Experimental Results)

Bu bölümde uygulamada kullanılan veri setleri, önerilen yöntem kullanılarak ağırlık katsayılarının bulunması, bulunan bu ağırlık katsayılarının veri setinde tatbik edilmesine bakılmaktadır.

**4.2.**Ayarlar (Settings)

Algoritma ayarları şu şekildedir:

|  |  |
| --- | --- |
| **Algoritma** | **Parametre Değerleri** |
| DDAO | Popülasyon Büyüklüğü=3,MaxIt=1000 Dim=30-50-100 |
| K-nn | Uzaklık Hesaplama Bağıntısı=Öklid |

**4.3.**Veri Setleri

Önerilen yöntemin sonuçlarını görebilmek için 1 veri seti ile karşılaştırma yapılmıştır. Veri setinin özellikleri şu şekildedir:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Boyut | Eğitim Örnek Sayısı | Test Örnek Sayısı |
| Örnek Veri Seti | 5 | 258 | 145 |

**4.4.**Deneysel Sonuçların Analizi

k-nn sonuç grafiği ve DDAO-knn sonuçları aşağıdaki grafik ve tablolarda gösterilmiştir.

|  |  |  |  |
| --- | --- | --- | --- |
| Algoritma | DDAO K-NN | | |
| 30 iterasyon | 50  iterasyon | 100  iterasyon |
| En İyi Değer (%) | 3,448276 | 5,517241 | 8,965517 |
| En Kötü Değer (%) | 51,72414 | 52,41379 | 49,65517 |
| Ortalama Değer (%) | 28,44828 | 28,27586 | 26,81034 |
| Standart Sapma (%) | 12,88663 | 12,42045 | 11,98029 |
| k-nnSonuç(%) | 13,79 | | |