**YZM 4032**

**Meta-Sezgisel Optimizasyon Dersi Ara Raporu**

**Algoritma Adı:** Dynamic differential annealed optimization

**Öğrenci Adı ve Soyadı:** Elif Şahinkaya

**Öğrenci No:**334341

**Öğrenci İletişim Bilgileri**

**Cep tel: 0**5537714333

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

**DDAO(Dynamic differential annealed optimization )Algorithm:**

* Bu yaklaşım geniş bir yelpazeyi küresel minimum veya maksimumun gerekli olduğu matematiksel optimizasyon problemleri çözmek için kullanılabilecek yeni bir optimizasyon algoritması önermektedir.
* Yeni algoritması, 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 yapılanlan 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.

**Diferansiyel evrim**

Benzerliğe rağmen, genetik algoritma (GA) ve diferansiyel evrim (DE) arasındaki temel fark,mutasyon süreci ve seçim . Genetik algoritmada daha iyi uygunlukta çözümlerin seçilmesi oldukça olasıdır.Ebeveyn olarak DE'deyken tüm çözümler ebeveyn olarak seçilmek için aynı şansa sahiptir. DE şu çözümleri seçer: GA, yakınsama için önemli bir avantajı olmasa bile çözümleri seçerken daha iyi yakınsama sağlar. Yakınsama performansında DE'nin GA'dan daha hızlı olmasının ana nedeni budur. DE'nin ana adımları şunlardır:

Initialization başlatma

Evaluation -Değerlendirme

Repeat -Tekrar et

Mutation -Mutasyon

Recombination -Rekombinasyon

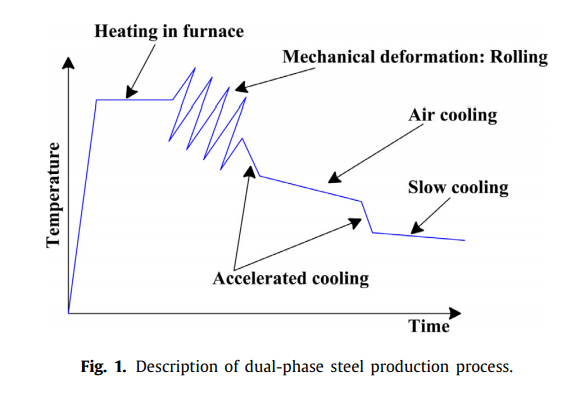
Evaluation- Değerlendirme

Selection- seçim

Until (termination criteria are met) -Kadar (sonlandırma kriterleri karşılanır)

Bu bölüm DE ile DDAO arasındaki temel farklılıkları belirtmek için eklenmiştir.

**Dinamik diferansiyel tavlı optimizasyon**

****

Mekanik iyileştirme için birçok çalışma yapılmıştır.çeliğin özellikleri gerekli kütleyi azaltmak için üretim süreci için ; maliyet . 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 özellikler.

Çift fazlı çelik üretmek, defor- ile birleştirilmiş doğru şekilde kontrol edilen bir tavlama sistemi çelik bileşenin eşleşmesi . Şekil 1 genel bir mikro yapıyı iyileştirme işlemini açıklayan şema sıradan çeliğin ultra mekanik ile çift fazlı çelik haline gelmesi ical özellikleri.

Genel olarak, yüksek bir sıcaklıkta demir erir sıvı hale ve sürekli soğutma ile katı hale gelir. oda sıcaklığında durumu. Metalin özellikleri bağlıdır katı haldeki mevcut fazlar hakkında: örneğin, ferritik faz süneklik verir, martensit fazı sorumludur sertlik ve bainit ve östenit gibi başka aşamalar da vardır. Genellikle, birçok demir fazı aynı katıda bir arada bulunur ve nihai mekanik özelliğe bir efekt kombinasyonu verirler. çelik bağları.

Bir demir parçası olan çift fazlı çelik (DP) üretmek için yeniden kristalleşme noktasının üzerinde yüksek bir sıcaklığa ısıtılır, mikroyapı yaklaşık olarak homojendir. Sonra dönem, demir parçası daha ince bir kalınlığa kadar yuvarlanmalıdır ve yuvarlanma sırasında metal soğuyacaktır. Mechani'yi uyguladıktan sonra- cal deformasyon, metal müteakip diferansiyele tabi tutulur soğutma; hızlandırılmış soğutma, hava soğutma, başka bir hızlandırılmış soğutma ve son olarak oda sıcaklığına yavaş yavaş soğutma. Bu süreç mikro yapıya sahip ferritik matris içeren çeliğe yol açar adalar şeklinde sert bir martensitik faz.

Kombinen yumuşak ferrit ve sert martensitin etkileri çift faz verir kabul edilebilir bir oluşum hızına sahip yüksek mukavemetli çelik, özellikle otomotiv endüstrisinde çok önemli . Daha önce açıklanan sistematik prosedür, bir yüksek kaliteli çelik tipi (çift fazlı) net bir optimization. Metal, düşük kaliteden yüksek kaliteye doğru geliştirilir. kalite durumu. Bu davranış, dinamik farklılığa ilham verdi. Bu çalışmada geliştirilen tavlı optimizasyon algoritması.

**DDAO matematiksel olarak aşağıdaki gibi formüle edilebilir:**

**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.

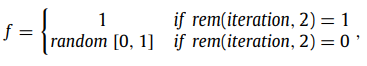
**4.** Şekil 1 farklı soğutma hızlarını, havayla soğutmayı, hızlandırılmış soğutmayı ve yavaş soğutmayı gösterir; soğutma işlemini temsil etmek için aşağıdaki denklemi önerilir:

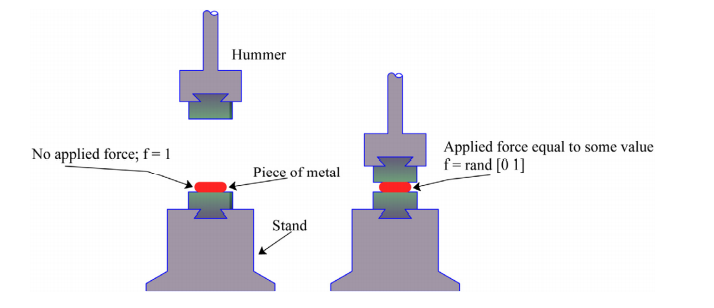
 (1)

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 Sr , içinde rastgele oluşturulmuş bir çözümdür. Daha açıklamak gerekirse, şek. 2, sıcaklık değiştiğinde I noktasından j noktasına √ T ile.

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 olarak seçim vardır ve hatta SR popülasyondan rastgele seçilir.

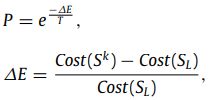
**5.** Isının diferansiyel indirgenmesi sırasında metal yuvarlanır ve bu mekanik işlem matematiksel olarak modellenmelidir. Programlama nedenleriyle, metalin haddeleme yerine dövme işlemine tabi tutulduğunu varsaymalıyız. Dövme sırasında çekicin dinamik davranışı, 1 ile rastgele bir sayı arasında dalgalanan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

**bir gelişme varsa**

**X ben = Sk**

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

**endif**

**sonu**

**X b'yi güncelle**

**T=T\*soğutma hızı**

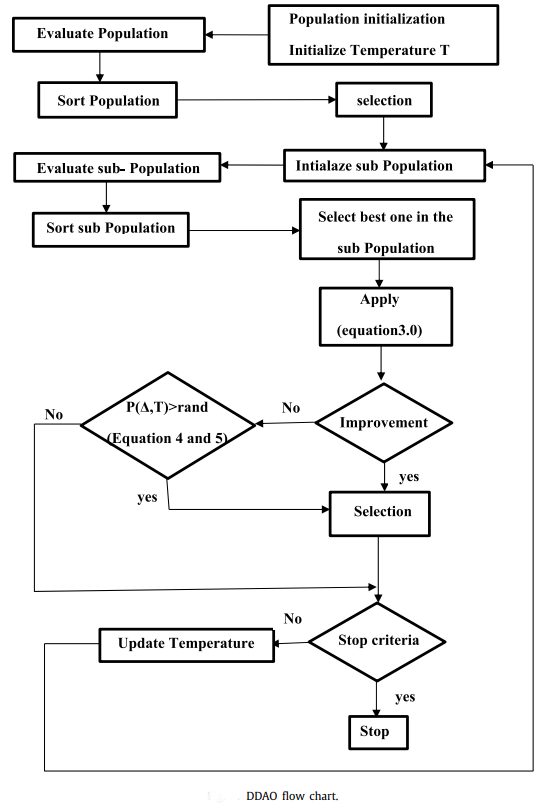
**t=t+1**

**son zaman**

**dönüş X b**

Yeni parametre; dövme parametresi f , optimizasyon sırasında DDAO'nun genel performansı üzerinde matematiksel problemler gözle görülür bir etkiye sahiptir.

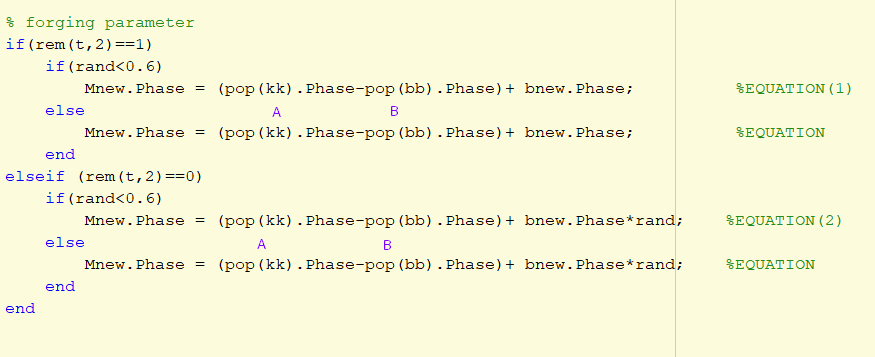
Bazı problemler için DDAO, denklem (3) 'te f eşittir 1 ile daha iyidir, f eşittir f ile daha kötüdür .Rastgele bir değer ve bunun tersi de başka bir problem kümesi için denklem (2) bu problem için bir çözüm verir.Yinelemeler f = 1 olarak kabul edilir ve yinelemenin geri kalanı f= rastgele [0 1] olarak kabul edilir. Bununla birlikte, dövme parametresi gelecekteki çalışmalar için daha fazla dikkat göstermelidir. DDAO'nun bir üç parametreli basit yapı; maksimum yineleme sayısı, alt yineleme sayısı ve soğutma ayarlanabilen oran. DDAO, popülasyon büyüklüğünden bağımsız olsa da, gelecek için sabit bir popülasyon kullanılabilir.



**Baz algoritma adı: DDAO**

**Caseler:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| equation | uygulanan yer |  |  |  |  |
| 1 | A | CASE1 | CASE2\_1 | CASE3\_1 |  |
| 1 | B | CASE2 | CASE2\_2 | CASE3\_2 |  |
| 2 | A | CASE3 | CASE2\_3 | CASE3\_3 |  |
| 2 | B | CASE4 | CASE2\_4 | CASE3\_4 |  |
| 1 | A,B | CASE5 | CASE2\_5 | CASE3\_5 |  |
| 1,2 | A,A | CASE6 | CASE2\_6 | CASE3\_6 |  |
| 1,2 | A,B | CASE7 | CASE2\_7 | CASE3\_7 |  |
| 1,2 | B,A | CASE8 | CASE2\_8 | CASE3\_8 |  |
| 1,2 | B,B | CASE9 | CASE2\_9 | CASE3\_9 |  |
| 2 | A,B | CASE10 | CASE2\_10 | CASE3\_10 |  |
| 1,1,2 | A,B,A | CASE11 | CASE2\_11 | CASE3\_11 |  |
| 1,1,2 | A,B,B | CASE12 | CASE2\_12 | CASE3\_12 |  |
| 1,2,2 | A,A,B | CASE13 | CASE2\_13 | CASE3\_13 |  |
| 1,2,2 | B,A,B | CASE14 | CASE2\_14 | CASE3\_14 |  |
| 1,2,1,2 | A,B,A,B | CASE15 | CASE2\_15 | CASE3\_15 |  |
|  |  | 0.6 | 0.3 | 0.9 | Oranlar |



**Deney**

Algoritmanın arama performansında iyileşme elde etmek için FDB uygulaması yapılmıştır.

* Çalışmalar CEC 2020 10 problem üzerinde yapılmıştır.
* Maksimum iterasyon Sayısı 10.000 \* D olarak belirlenmiştir
* **equation sutunu için resimlere ve yukarıdaki tabloya bakınız.**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Case Adı | Boyut / D | Uygulanan  equation | Uygulanan Yer | Uygulanan  Oran | İyi | Aynı | Kötü |
| Case1 | 30 | 1 | A | 0.6 | 0 | 10 | 0 |
| Case2\_1 | 30 | 1 | A | 0.3 | 0 | 10 | 0 |
| Case3\_1 | 30 | 1 | A | 0.9 | 0 | 10 | 0 |
|  | | | | | | | |
| Case2 | 30 | 1 | B | 0.6 | 0 | 10 | 0 |
| Case2\_2 | 30 | 1 | B | 0.3 | 2 | 8 | 0 |
| Case3\_2 | 30 | 1 | B | 0.9 | 1 | 9 | 0 |
|  | | | | | | | |
| Case3 | 30 | 2 | A | 0.6 | 0 | 9 | 1 |
| Case2\_3 | 30 | 2 | A | 0.3 | 0 | 10 | 0 |
| Case3\_3 | 30 | 2 | A | 0.9 | 0 | 10 | 0 |
|  | | | | | | | |
| Case4 | 30 | 2 | B | 0.6 | 1 | 9 | 0 |
| Case2\_4 | 30 | 2 | B | 0.3 | 0 | 10 | 0 |
| Case3\_4 | 30 | 2 | B | 0.9 | 0 | 9 | 1 |
|  | | | | | | | |
| Case5 | 30 | 1 | A,B | 0.6 | 1 | 8 | 1 |
| Case2\_5 | 30 | 1 | A,B | 0.3 | 1 | 9 | 0 |
| Case3\_5 | 30 | 1 | A,B | 0.9 | 0 | 8 | 2 |
|  | | | | | | | |
| Case6 | 30 | 1,2 | A,A | 0.6 | 1 | 8 | 1 |
| Case2\_6 | 30 | 1,2 | A,A | 0.3 | 0 | 10 | 0 |
| Case3\_6 | 30 | 1,2 | A,A | 0.9 | 1 | 8 | 1 |
|  | | | | | | | |
| Case7 | 30 | 1,2 | A,B | 0.6 | 0 | 10 | 0 |
| Case2\_7 | 30 | 1,2 | A,B | 0.3 | 0 | 9 | 1 |
| Case3\_7 | 30 | 1,2 | A,B | 0.9 | 1 | 9 | 0 |
|  | | | | | | | |
| Case8 | 30 | 1,2 | B,A | 0.6 | 1 | 8 | 1 |
| Case2\_8 | 30 | 1,2 | B,A | 0.3 | 0 | 9 | 1 |
| Case3\_8 | 30 | 1,2 | B,A | 0.9 | 0 | 10 | 0 |
|  | | | | | | | |
| Case9 | 30 | 1,2 | B,B | 0.6 | 0 | 10 | 0 |
| Case2\_9 | 30 | 1,2 | B,B | 0.3 | 0 | 10 | 0 |
| Case3\_9 | 30 | 1,2 | B,B | 0.9 | 0 | 9 | 1 |
|  | | | | | | | |
| Case10 | 30 | 2 | A,B | 0.6 | 0 | 9 | 1 |
| Case2\_10 | 30 | 2 | A,B | 0.3 | 1 | 9 | 0 |
| Case3\_10 | 30 | **2** | **A,B** | **0.9** | **2** | **8** | **0** |
|  | | | | | | | |
| Case11 | 30 | 1,1,2 | A,B,A | 0.6 | 0 | 9 | 1 |
| Case2\_11 | 30 | 1,1,2 | A,B,A | 0.3 | 0 | 9 | 1 |
| Case3\_11 | 30 | 1,1,2 | A,B,A | 0.9 | 0 | 8 | 2 |
|  | | | | | | | |
| Case12 | 30 | 1,1,2 | A,B,B | 0.6 | 0 | 9 | 1 |
| Case2\_12 | 30 | 1,1,2 | A,B,B | 0.3 | 0 | 10 | 0 |
| Case3\_12 | 30 | 1,1,2 | A,B,B | 0.9 | 0 | 8 | 2 |
|  | | | | | | | |
| Case13 | 30 | 1,2,2 | A,A,B | 0.6 | 0 | 9 | 1 |
| Case2\_13 | 30 | 1,2,2 | A,A,B | 0.3 | 2 | 8 | 0 |
| Case3\_13 | 30 | 1,2,2 | A,A,B | 0.9 | 2 | 7 | 1 |
|  | | | | | | | |
| Case14 | 30 | 1,2,2 | B,A,B | 0.6 | 1 | 9 | 0 |
| Case2\_14 | 30 | 1,2,2 | B,A,B | 0.3 | 1 | 9 | 0 |
| Case3\_14 | 30 | **1,2,2** | **B,A,B** | **0.9** | **3** | **7** | **0** |
|  | | | | | | | |
| Case15 | 30 | 1,2,1,2 | A,B,A,B | 0.6 | 1 | 8 | 1 |
| Case2\_15 | 30 | 1,2,1,2 | A,B,A,B | 0.3 | 9 | 17 | 4 |
| Case3\_15 | 30 | 1,2,1,2 | A,B,A,B | 0.9 | 1 | 8 | 1 |
|  | | | | | | | |

Baz algoritma = DDAO: 24,102

Best caseler = Case3\_10: 21,05

Case3\_14: 21,33