



T.C

SAKARYA ÜNİVERSİTESİ
BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

BSM 310 – YAPAY ZEKA

Grup üyeleri:
G161210065-Mehmet Tayfur ÜNAL
G161210053-Emre MUTLU
G161210025-Berat YONCACILAR

Sakarya
2020

TABU ARAMA ALGORİTMASI

Tabu Algoritması Giriş

Tabu Arama Algoritması, optimizasyon problemlerinin çözümü için Fred W. Glover tarafından 1986'da oluşturulan ve 1989'da geliştirilip resmileştirilen meta-sezgisel iteratif bir araştırma algoritmasıdır.

Temel yaklaşım, son çözüme götüren adımın dairesel hareketler yapmasını önlemek için bir sonraki döngüde tekrarın yasaklanması veya cezalandırılmasıdır. Böylece yeni çözümlerin incelenmesiyle Tabu Arama algoritması, bölgesel en iyi çözümün daha ilerisinde bulunan çözümlerin araştırılabilmesi için bölgesel-sezgisel araştırmaya kılavuzluk etmektedir.

Tabu Arama algoritmasının bölgesel optimalliği aşmak amacıyla kullandığı temel prensip, değerlendirme fonksiyonu tarafından her iterasyon da en yüksek değerlendirme değerine sahip hareketin bir sonraki çözümü oluşturmak amacıyla seçilmesine dayanmaktadır. Bunu sağlamak amacıyla bir tabu listesi oluşturulur, tabu listesinin orijinal amacı önceden yapılmış bir hareketin tekrarından çok tersine dönmesini önlemektir. Tabu listesi kronolojik bir yapıya sahiptir ve esnek bir hafıza yapısı kullanır. Tabu arama algoritması her ne kadar istenmeyen noktaların işaretlenmesi olarak açıklanmış olsa da daha cazip noktaların işaretlenmesi olarak da kullanılır.

Meta-Sezgisel Yöntemler Nedir?

- Optimizasyon problemlerini çözümünde sezgisel- doğrusal programlamanın yeterli olmadığı durumlarda kullanılan ve temel sezgisel yöntemleri birleştirmeye çabalayan yaklaşımlardır.
- Kısa zamanda çözüme ulaşmak için kullanılır.
- Amacı en iyiye yakın çözümleri bulmak için arama uzayını hızlı bir şekilde araştırmaktır.
- En iyi çözümü garanti etmedikleri için yaklaşık algoritmalar olup genellikle deterministik değildirler.

Tabu Algoritması Metodolojisi Genel Kavramlar

Başlangıç çözümünün oluşturulması:

En genel şekilde başlangıç çözümü rastlantısal olarak elde edilir. Ancak ilgilenilen, problem için geliştirilmiş olan bir sezgisel algorithmadan yararlanarak da başlangıç çözümünden elde edilmesi mümkündür.

Hareket Mekanizması:

Mevcut bir çözümde yapılan bir değişiklikle yeni bir çözümün elde edilmesi hareket mekanizmasıyla gerçekleştirilir. Hareket mekanizmasındaki olası hareketler, mevcut çözümün komşularını oluşturur.

Komşuluk:

Tabu Aramada en önemli bileşenlerden birisi de komşuluk yapısıdır. Çözümü iyileştirmek için amaç fonksiyonun değeri açısından en iyi hareketlerin seçilmesidir. Komşulukların oluşturulmasında seçilen komşuluk üretme yapısına göre problem boyutu n olduğu durumda $(n-1)$ tane komşuluk üretilir.

Tabu Listesi:

Tabu listesi, araştırmanın herhangi bir iterasyonu içinde hangi seçimlerin tabu grubu olmak zorunda olduğu ve kaç tanesi üzerinde karar verme ve tabu listesini güncelleme ile ilgilenir. Tabu listesinin boyutu, sonucu önemli derecede etkileyebilir. Deneysel sonuçlar problemin boyutu büyüdükçe problem boyutu ile orantılı olarak tabu listesi uzunluğunun da büyümesi gerektiğini göstermiştir.

Tabu Yıkma Kriterleri:

Tabu yıkma kriterleri, tabunun ortadan kalkabileceği durumları ifade etmektedir. En genel tabu yıkma kriteri, mevcut durumdan daha iyi bir sonuç verecek tabu hareketinin yapılmasına izin verilmesidir. Bu kriterin kullanılması TA algoritmasının etkinliğini artırmaktadır. Ayrıca, eğer tüm mümkün hareketler tabu ise bu hareketlerden tabu süresinin bitmesine en yakın olan bir tabu harekete izin verilir.

Hafıza:

TA algoritmasının temel elemanlarından biride hafızadır. Arama boyunca ortaya çıkan durumlar, H hafızasına kayıt edilir. Yapılmasına izin verilmeyen hareketler "tabu" olarak adlandırılır ve esnek hafıza içinde "tabu listesi" adı altında kaydedilirler. Bu hareketler belli bir süre sonra tabu listesinden çıkarılır ve yapılmasına izin verilir.

Hafıza Türleri:

Yakınlık Geçmişe dayalı Bellek Yapısı (Kısa Dönem Hafıza) (YBY)

En temel Tabu Arama bellek yapısıdır. Esas olarak görevi yakın geçmişte yapılan hareketlerin bir süre için tabu olarak işaretlemektir. Burada kastedilen süre tabu süresi yani çerim sayısıdır. Belirlenen t değeri süre kadar hareket tabu olarak belirlenir ve daha sonra belirlenen hareket tabu olmaktan çıkarılır. Örneğin $t = 5$ olacak şekilde statik olarak değer verilebildiği gibi t değeri min t ve max t sınırları arasında sistematik veya rastgele seçilebilir.

Sıklığa Dayalı Bellek Yapısı (Orta ve Uzun Dönem Hafıza) (SBY)

Sıklığa dayalı bellek yapısı tamamlayıcı bir özelliğe sahiptir genellikle YBY ile birlikte ikincil bir bellek olarak kullanılır. SBY adından da anlaşılacağı gibi yapılan hareketlerin sıklığını bellekte tutar. Ancak bir hareketin kaç kez yapıldığı bilgisi yerine, çözümün kalitesine ve hareketin etkisine bağlı bilgilerin bellekte tutulması daha faydalı yaklaşımdır. Sıklık ölçütünün dört çeşidi vardır:

- Her bir hareketin toplam tekrar sayısı
- Toplam hareket sayısı
- En yüksek hareket tekrar sayısı
- Ortalama hareket tekrar sayısı

Durdurma Koşulu:

Tabu Arama algoritması, bir veya birden fazla durdurma koşulunu sağlayıncaya kadar aramasını sürdürmektedir. Bu koşullardan bazıları aşağıda verilmiştir.

- Seçilen bir komşu çözümün komşusunun olmaması
- Belirli bir iterasyon sayısına ulaşılması
- Belirli bir çözüm değerine ulaşılması
- Algoritmanın bir yerde tıkanması ve daha iyi sonuç üretememesi

Tabu Arama Algoritması Adımları

ADIM 1: Tabu Arama algoritması, bir başlangıç çözümü ile aramaya başlar.

ADIM 2: Algoritmanın her iterasyonunda tabu olmayan bir hareket ile mevcut çözümün komşuları içerisinde bir tanesi seçilerek değerlendirilir.

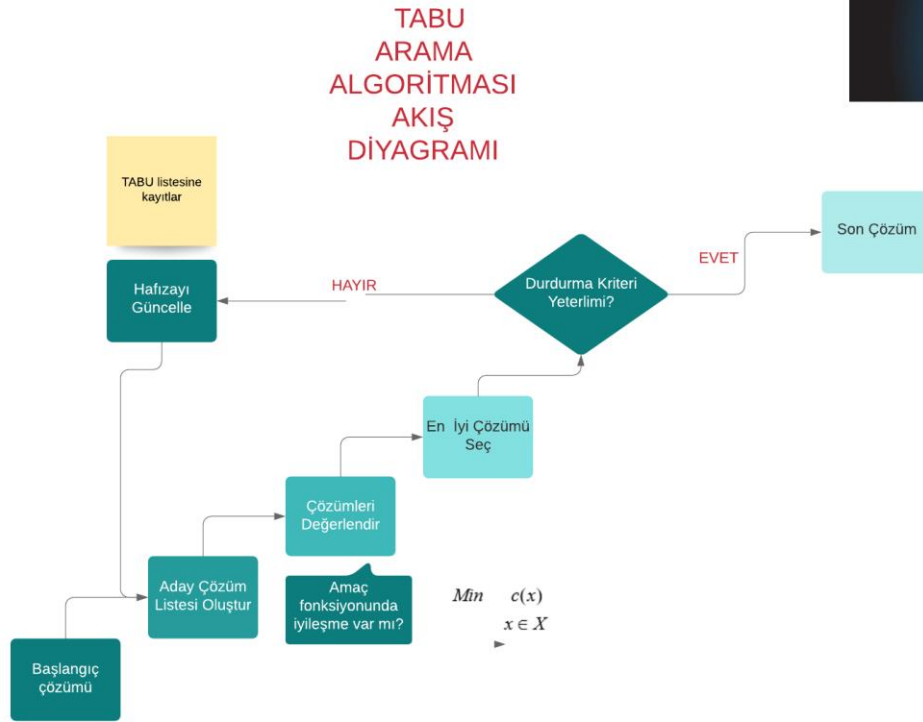
ADIM 3: Eğer amaç fonksiyonunun değerinde bir iyileştirme sağlanmışsa komşu çözüm, mevcut çözüm olarak dikkate alınır.

ADIM 4: Seçilen bir hareket tabu olmasına rağmen tabu yıkma kriterlerini sağlıyorsa, mevcut çözümü oluşturmak için uygulanabilir.

ADIM 5: Geriye dönüşleri önlemek için, birtakım hareketler tabu listesine kaydedilerek tekrar yapılması belirli bir süre için yasaklanır.

ADIM 6: Belirlenen bir durdurma koşuluna göre algoritmanın çalışması sonlanmaktadır. Eğer durdurma koşuluna uyulmuyorsa adım-2 ye dönlür.

Tabu Algoritması Akış Diyagramı



Tabu Arama Sözde Kod

```
1 sBest ← s0
2 bestCandidate ← s0
3 tabuList ← []
4 tabuList.push(s0)
5 while (not stoppingCondition())
6     sNeighborhood ← getNeighbors(bestCandidate)
7     bestCandidate ← sNeighborhood[0]
8     for (sCandidate in sNeighborhood)
9         if ( (not tabuList.contains(sCandidate)) and (fitness(sCandidate) > fitness(bestCandidate)) )
10             bestCandidate ← sCandidate
11         end
12     end
13     if (fitness(bestCandidate) > fitness(sBest))
14         sBest ← bestCandidate
15     end
16     tabuList.push(bestCandidate)
17     if (tabuList.size > maxTabuSize)
18         tabuList.removeFirst()
19     end
20 end
21 return sBest
```

Örnek Soru-1

Bir işletme, değişik tipte dört yeni makine satın almıştır. İşletme içinde makinaların yerleştirilebileceği dört seçenek yer bulunmaktadır. Birim zaman başına malzeme aktarma giderleri her makina için öngörölmüş ve aşağıda çizelge biçiminde sunulmuştur.

	Aday Konumlar				
Yeni Makinalar		1	2	3	4
	A	180	10	96	146
	B	138	28	166	172
	C	114	186	4	158
	D	14	154	150	46

Amaç Tabu Arama Algoritmasını kullanarak en düşük maliyetli veya en düşük maliyetli çözüme yakın makine yerleştirmeyi elde etmek.

Başlangıç konumu ve değeri (başlangıç olarak en iyi yerleştirme)

D-C-B-A: $14+186+166+146 = 512$ olsun

En iyi çözüm: 512

İterasyon sayısı: 5

Tabu Listesi uzunluğu :3

Tabu Listesi: (),(),()

Komşuluklar: 1. iterasyon

- a) C-D-B-A: $114+154+166+146 = 580$
- b) B-C-D-A: $138+186+150+146 = 620$
- c) A-C-B-D: $180+186+166+46 = 578$

En iyi çözümden daha iyi bir çözüm yok bu nedenle bölgesel çeşitlendirme yapmak için en iyi yerleştirme konumundaki makinelerden 2 tanesinin yeri rasgele değiştirilir. Bu değişim 2. ve 4. konumdaki makineler olsun.

D-A-B-C: $14+10+166+158=348$

En iyi yerleştirme: DABC

En iyi Çözüm: 348

Tabu Listesi: (a, c), (), ()

Komşuluklar: 2. İterasyon

- d) A-D-B-C: $180+154+166+158 = 658$
- e) B-A-D-C: $138+10+150+158 = 456$
- f) C-A-B-D: $114+10+166+46 = 336$

En iyi yerleştirme: CABD

En iyi Çözüm: 336

Tabu Listesi: (a, c), (e, f) , ()

Komşuluklar: 3. İtersayon

- g) A-C-B-D: $180+186+166+46 = 578$
- h) B-A-C-D: $138+10+4+46 = 198$
- i) D-A-B-C: $14+10+166+158 = 348$

En iyi yerleştirme: B-A-C-D

En iyi Çözüm: 198

Tabu Listesi: (a, c), (e, f), (h, i)

Komşuluklar: 4. İterasyon

$$A-B-C-D: 180+28+4+46 = 258$$

$$C-A-B-D: 114+10+166+46 = 336$$

$$D-A-C-B: 14+10+4+172 = 200$$

En iyi çözümden daha iyi bir çözüm yok bu nedenle bölgesel çeşitlendirme yapmak için en iyi yerleştirme konumundaki makinelerden 2 tanesinin yeri rasgele değiştirilir. Bu değişim 2. ve 4. konumdaki makineler olsun.

$$B-D-C-A: 138+154+4+146 = 442$$

Komşuluklar: 5. İterasyon

$$D-B-C-A: 14+28+4+146 = 192$$

$$C-D-B-A: 114+28+166+146 = 580$$

$$A-D-C-B: 180+154+186+172 = 692$$

En iyi yerleştirme: D-B-C-A

En iyi Çözüm: 192

Tabu Listesi: (c, d), (b, c), (d, b)

Örnek Soru-2

Bu kısımda TA algoritmasının işleyişi literatürde sıklıkla yer alan iki basit örnek üzerinden adım adım gidilerek gösterilecektir. Bu örneklerden ilki Denklem 2.1 de verilen fonksiyonun global en küçük değerli noktasının bulunmasıdır.

$$f(x) = f(x) \begin{cases} x^2, & \text{eger } x \leq 1 \\ (x-3)^2 - 3 & \text{Degilse} \end{cases}$$

Denklem 2.1

Söz konusu fonksiyonun en küçük değeri $f(3) = -3$ tür. Rastlantı bir noktadan aramaya başlanacaktır. Aramaya $f(-7.39) = 54.61$ noktasından başlanmaktadır. Dört farklı komşu seçme fonksiyonu vardır. ($\eta \in [0,1]$ - rastlantı bir sayıdır) ne bunlar denklem 2.2 – 2.5 de verilmektedir. Örnek Tabu Arama algoritmasının temel işleyişini canlandırmak için verildiğinden yalnız yakınlığa bağlı bellek yapısı kullanılmıştır. Ters hareketlerin yapılması tabu belirlenmiştir. Basit bir örnek olduğundan tabu süresi 1 olarak alınmıştır. Her bir hareketin tabu listesi $h(x)$ ile temsil edilmiştir. Denklem 2.1 de ilk 18 çevrim gösterilmektedir

Tabu Arama ve Kapasite Kısıtsız Tedarikçi Seçme Problemi

$$\psi^1(i) = 2\eta + i \quad \text{Denklem 2.2}$$

$$\psi^2(i) = 4\eta + i \quad \text{Denklem 2.3}$$

$$\psi^3(i) = 1 - 2\eta \quad \text{Denklem 2.4}$$

$$\psi^4(i) = i - 2\eta \quad \text{Denklem 2.5}$$

Böylece $x=1$ 'in tersi $x=3$, $x=3$ 'ün tersi $x=1$, $X=2$ 'nin tersi $x=4$ ve $x=4$ 'ün tersi $x=2$ olur.

Denklem 2.1 $f(x)$ 'in küçük olduğu noktayı arayan bir TA algoritması

t	i	$\hat{i} = \psi_x(i)$	$f(i)$	$f(\hat{i})$	η	x	h1	h2	h3	h4
0	-7.39	-7.75	54.61	60.20	0.18	3	0	0	0	0
1	-7.39	-5.71	54.61	32.55	0.42	2	2	0	0	0
2	-5.71	-6.13	32.55	37.60	0.21	3	2	0	0	3
3	-5.71	-2.36	32.55	5.56	0.84	2	4	0	0	3
4	-2.36	-3.09	5.56	9.56	0.37	3	4	0	0	5
5	-2.36	-3.97	5.56	15.73	0.80	3	6	0	0	5
6	-2.36	-3.98	5.56	15.84	0.81	3	7	0	0	5
7	-2.36	-4.44	5.56	19.74	0.52	4	8	0	0	5
8	-2.36	-6.19	5.56	38.37	0.96	4	8	9	0	5
9	-2.36	-3.82	5.56	14.63	0.73	3	8	10	0	5
10	-2.36	-0.12	5.56	0.02	0.56	2	11	10	0	5
11	-0.12	-0.37	0.02	0.14	0.12	3	11	10	0	12
12	-0.12	2.41	0.02	-2.65	0.63	2	13	10	0	12
13	2.41	1.39	-2.65	-0.41	0.51	3	13	10	0	14
14	2.41	2.23	-2.65	-2.40	0.09	3	15	10	0	14
15	2.41	2.48	-2.65	-2.73	0.02	2	16	10	0	14

KAYNAKÇA

- Genel Amaçlı Arama Algoritmaları ile Benzetim En İyilemesi: En İyi Kanban Sayısının Bulunması, Fulya Altıparmak, Berna Dengiz
- Structural Optimization Tabu Search, Mustafa Mohammed Rashid
- Tabu Araştırma Algoritmasına Dayalı Sistem Kimliklendirme İşleminde Kontrol Parametrelerinin Çözümüne Etkisi, Aytekin BAĞIŞ
- Glover, F., Tabu Search-Part I, ORSA Journal on Computing