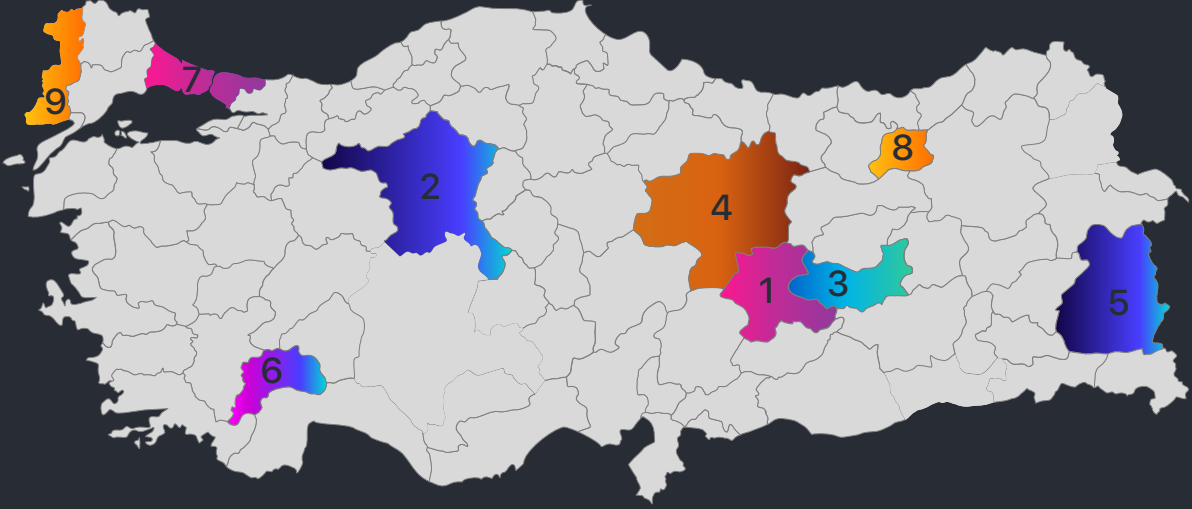


TAVLAMA BENZETİMİ

Miting Güzergahı Problemi



	Malatya	Ankara	Elazığ	Sivas	Van	Burdur	İstanbul	Bayburt
Malatya	0	655	99	246	643	917	1100	447
Ankara	655	0	753	316	1249	424	450	779
Elazığ	246	753	0	316	507	1016	1206	370
Sivas	246	316	316	0	823	767	892	351
Van	643	1249	507	823	0	1496	1606	499
Burdur	917	424	1016	767	1496	0	574	1107
İstanbul	1100	450	1206	892	1606	574	0	1125
Bayburt	447	779	370	351	499	1107	1125	0
Edirne	1337	687	1436	1123	1849	830	239	1362

Problem

Tavlama benzetimi ile çözülen gezici satıcı problemini kullanarak oluşturduğum problem daha önce yayınlanmış "Siyasi Parti Mitinglerinin Gezgin Satıcı Problemi Yaklaşımı ile Analizi" makalesinden esinlenilmiştir. Problemimizde bazı şehirler mevcut ve bu şehirlerin hepsine 1 kere gidilmek şartı ile bütün şehirlere en kısa hangi güzergah ile gidilir konusunu çözeceğiz. Bunun için belirlediğim şehirler ve bunlar arasındaki mesafe yukarıdaki görselde verilmiştir.

Çözüm

Bir noktadan çözüme başlayabilmek için önce rassal olarak bir güzergah belirleriz. Ardından bu güzergahın ana fonksiyon değerini hesaplarız. Yani güzergahın toplam uzunluğunu buluruz. Ardından şehirleri 2'şer olarak seçerek değiştiririz. Bunun için yine 2 adet rassal sayı belirleriz. Bu işlemin ardından tekrar ana fonksiyon değeri hesaplarız. Son olarak ilk değerle ikinci değeri kıyaslarız. Yeni değer ilk değerden daha iyiye onu ana değer olarak seçeriz. Eğer daha kötü bir değer ise ne kadar kötüleştiğini bulup vereceğimiz sıcaklık parametresi ile kabullenme fonksiyonunu (Acceptance Function) hesaplarız. Bu sayede kendi belirleyeceğimiz sıcaklık aralığına ve katsayısına göre her adımda hesaplanan değerler sürekli kıyaslanarak sürekli daha iyileşir. Bu sayede optimum nokta olan en kısa mesafeyi bulmuş oluruz.

```
function [cozum cozumeniyi objeniye sayac objit] = saticisiman(gezici,T,Tend,sk)
```

```
sehirsayisi = size(gezici,1); % Rassal bir güzergah oluştur.
cozum=randperm(sehirsayisi); % Başlangıç çözümü. (Ana Fonksiyon Değeri)
```

```
obj=0;
```

```
% Aşağıda random oluşan güzergahın kat edilen mesafeyi hazırlar
```

```
for i=1:sehirsayisi-1
```

```
    sehir1=cozum(i);
```

```
    sehir2=cozum(i+1);
```

```
    obj=obj+gezici(sehir1,sehir2);
```

```
end
```

```
sehir1=cozum(end)
```

```
sehir2=cozum(1);
```

```
obj=obj+gezici(sehir1,sehir2);
```

```
sayac=1;
```

```
cozumeniyi=cozum;
```

```
objeniye=obj;
```

```
objit=obj;
```

```
gezici = xlsread('mesafeler.xlsx');
```

```
save C:\Users\al\Documents\MATLAB\sehirler gezici
```

```
[cozum cozumeniyi objeniye sayac objit]=saticisiman(gezici,1000,1,0.9)
```

```
plot(objit);
```

```
% İl uzaklıklarımızı mesafeler.xlsx dosyasından alıyoruz.
```

```
% saticisiman dosyasına sıcaklık,sıcaklık bitişi ve sıcaklık katsayılarımızı gön-  
deriyoruz.
```

```
while (T>Tend) % Belirlenen sıcaklık değeri kadar döngü kurar. Tend belirlenirken daha küçük belirlenmelidir.
```

```
    deg=randperm(sehirsayisi);
```

```
    deg=deg(1:2);
```

```
    komsu=cozum; % Değişim sağlamak için rassal 2 şehir seçer.
```

```
    temp=komsu(deg(1));
```

```
    komsu(deg(1))=komsu(deg(2));
```

```
    komsu(deg(2))=temp;
```

```
    obj_komsu=0;
```

```
    for i=1:sehirsayisi-1
```

```
        sehir1=komsu(i);
```

```
        % Yeni oluşan değer Ana fonksiyon değerini hesaplar.
```

```
        sehir2=komsu(i+1);
```

```
        obj_komsu=obj_komsu+gezici(sehir1,sehir2);
```

```
    end
```

```
    sehir1=komsu(end)
```

```
    sehir2=komsu(1);
```

```
    obj_komsu=obj_komsu+gezici(sehir1,sehir2);
```

```
    if (obj_komsu<=obj)
```

```
        cozum=komsu;
```

```
        obj=obj_komsu;
```

```
    else %Eğer komşu iyiye orayı kabul et değilse kabul olasılığını hesapla.
```

```
        de=obj_komsu-obj;
```

```
        kabul=exp(-de/T); % Kabullenme Fonksiyonu
```

```
        rs=unifrnd(0,1);
```

```
        if (rs<kabul)
```

```
            cozum=komsu;
```

```
            obj=obj_komsu;
```

```
        end
```

```
    end
```

```
T=T*sk;
```

```
sayac=sayac+1;
```

```
    if(obj<min(objit)) %Eğer hesaplanan değer daha kötü ise onu alma.
```

```
    objit(sayac)=obj;
```

```
    else
```

```
        objit(sayac)=objit(sayac-1)
```

```
    end
```

```
    if(objit(sayac)<objeniye)
```

```
        cozumeniyi=cozum; %En iyi çözümü hespalar.
```

```
        objeniye=obj;
```

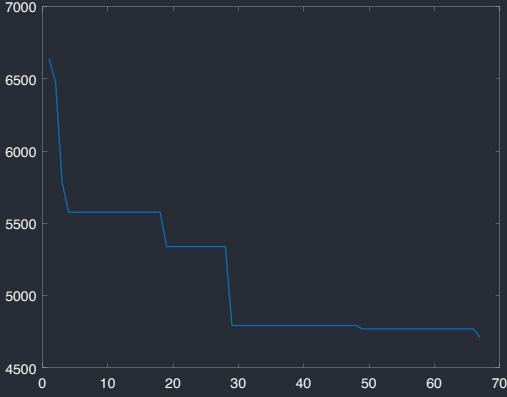
```
    end
```

```
end
```

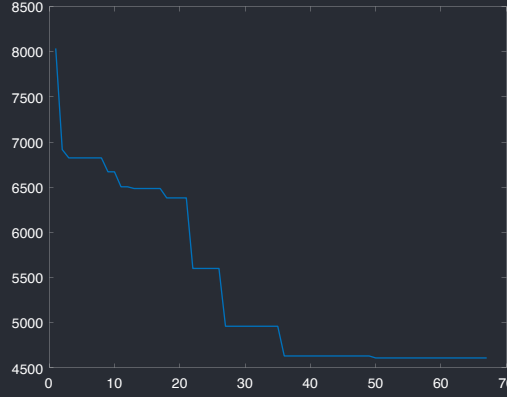
```
end
```

Çözüm Çıktıları

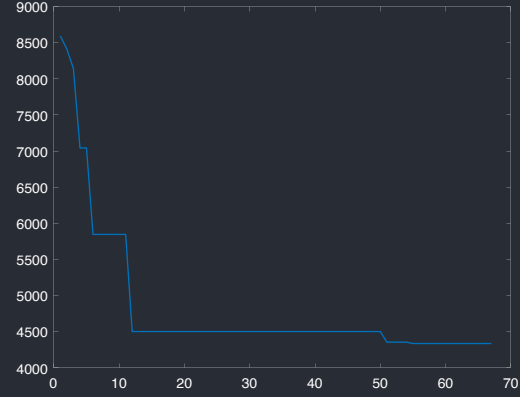
Örnek 1



Örnek 2



Örnek 3



Name	Value
cozum	[1,2,7,9,6,4,3,5,8]
cozumeniye	[1,2,7,9,6,4,3,5,8]
gezici	9x9 double
objeniye	4710
objit	1x67 double
sayac	67

Name	Value
cozum	[4,8,5,3,1,6,7,9,2]
cozumeniye	[4,8,5,3,1,6,7,9,2]
gezici	9x9 double
objeniye	4336
objit	1x67 double
sayac	67

Name	Value
cozum	[7,9,6,3,8,5,1,4,2]
cozumeniye	[7,9,6,3,8,5,1,4,2]
gezici	9x9 double
objeniye	4609
objit	1x67 double
sayac	67

Sonuç

Daha önceki yaptığım denemelerde dahil olarak en iyi sonuç olarak **4336 KM** ve en iyi güzergah olarak $4 > 8 > 5 > 3 > 1 > 6 > 7 > 9 > 2$ olarak hesapladım. Bu sırasıyla Sivas > Bayburt > Van > Elazığ > Malatya > Burdur > Edirne > İstanbul > Ankara olarak hesaplanır. Unutulmamalı ki verilen parametreleri değiştirerek en optimum nokta bulunabilir ve sistemin çalışması dahada hızlandırılabilir. Bunun için parametre değerlerini iyi ayarlamak önemlidir. Programın görselleştirilmiş hali aşağıdadır.

