

T.C.
BALIKESİR ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



201613709028 GİZEM SOLMAZ

**BMM4101 YAPAY ZEKA TEKNİKLERİ FİNAL
ÖDEVİ**

Danışman: Dr. Öğr. Üyesi Kadriye ERGÜN

BALIKESİR, OCAK 2022

Çalışmanın tanımı ve amacı

Genetik algoritma kullanarak, haritada belirtilen tüm noktalardan geçmek şartıyla, kuş uçuşu en kısa yol haritasını çizmek.

Kullanılan program veya yazılımlar hakkında bilgi verilmesi ve gerekçeleri

Kullanılan program MATLAB'dır. MATLAB ileri düzey hesaplamaları diğer dillere göre çok daha iyi yapabilmektedir. Birçok üst düzey matematiksel hesaplamalar gerektiren işlemlerde kullanılmaktadır. Laboratuvar ve bilim insanları için geliştirilen yazılımlarda ve görüntü işleme, yapay zeka vb. teknikleri kullanan projelerde MATLAB sıklıkla karşımıza çıkmaktadır. Bu ödevde de genetik algoritmayı kullanacağımızdan MATLAB dili bunun için biçilmiş kaftandır.

İlgili haritalar, koordinat tablosu, kuş uçuşu mesafenin nasıl hesaplandığı hakkında bilgi

Koordinatların alımı:

Google Maps aracılığı ile Balıkesir Merkez ilçesinin haritasını aldım. Bu haritada rastgele seçmiş olduğum 20 noktayı belirledim. Bu belirtilen noktaların yine Google Maps yardımıyla enlem ve boylam diğer bir deyişle x ve y koordinatlarını aldım. Bu alınan koordinatları program ilk çalıştığında tanımladım. Koordinat tablosu Şekil 1.a'daki gibidir. Matlab ortamında ise kodlaması Şekil 1.b deki gibidir. Aynı zamanda Google Maps'den aldığım konumların ekran çıktıları ödev raporunun son kısmında mevcuttur.

	1	2
1	39.6595	27.8787
2	39.6509	27.9024
3	39.6390	27.9119
4	39.6237	27.8953
5	39.6271	27.8856
6	39.6389	27.8903
7	39.6339	27.8757
8	39.6478	27.8613
9	39.6590	27.8548
10	39.6612	27.8814
11	39.6680	27.8888
12	39.6688	27.9198
13	39.6558	27.9202
14	39.6573	27.9041
15	39.6481	27.8891
16	39.6657	27.9041
17	39.6725	27.8848
18	39.6573	27.8699
19	39.6451	27.8799
20	39.6317	27.8995

Şekil 1.a

```
1 - locations = zeros(20, 2);
2 - locations = [39.6594540000000,27.8787290000000;
3     39.6509470000000,27.9023890000000;
4     39.6389980000000,27.9119390000000;
5     39.6237380000000,27.8953470000000;
6     39.6271390000000,27.8855590000000;
7     39.6389060000000,27.8903330000000;
8     39.6338500000000,27.8756510000000;
9     39.6478220000000,27.8613270000000;
10    39.6590340000000,27.8547620000000;
11    39.6612400000000,27.8813810000000;
12    39.6680390000000,27.8887820000000;
13    39.6687740000000,27.9198170000000;
14    39.6558180000000,27.9201750000000;
15    39.6572880000000,27.9040610000000;
16    39.6480980000000,27.8891400000000;
17    39.6656500000000,27.9040610000000;
18    39.6724500000000,27.8848430000000;
19    39.6572880000000,27.8699220000000;
20    39.6450650000000,27.8799480000000;
21    39.6317190000000,27.8995270000000];
22
```

Şekil 1.b

Kuş uçuşu mesafenin hesaplanması:

Kuş uçuşu mesafenin tanımlaması ise 2 nokta arası mesafeyi hesaplamaya yarayan formülü kullandım. Bu formül Şekil 2'de gösterilmiştir.

$$\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2}$$

Şekil 2

Mesafe tablosu:

Bu formül bize 2 boyutlu bir zemindeki 2 noktanın arasındaki uzaklığı vermektedir. Bu formülü kullanarak elde etmiş olduğum mesafeleri 20x20 boyutundaki matriste belirttim. Bu matris Şekil 3.a'da gösterilmiştir.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	0.0251	0.0390	0.0394	0.0330	0.0236	0.0258	0.0209	0.0240	0.0032	0.0132	0.0421	0.0416	0.0254	0.0154	0.0261	0.0144	0.0091	0.0144	0.0347
2	0.0251	0	0.0153	0.0281	0.0292	0.0170	0.0317	0.0412	0.0483	0.0234	0.0218	0.0249	0.0184	0.0066	0.0136	0.0148	0.0278	0.0331	0.0232	0.0194
3	0.0390	0.0153	0	0.0225	0.0289	0.0216	0.0367	0.0514	0.0606	0.0378	0.0371	0.0308	0.0187	0.0199	0.0245	0.0278	0.0430	0.0458	0.0326	0.0144
4	0.0394	0.0281	0.0225	0	0.0104	0.0160	0.0221	0.0417	0.0538	0.0400	0.0448	0.0513	0.0406	0.0347	0.0251	0.0428	0.0498	0.0421	0.0263	0.0090
5	0.0330	0.0292	0.0289	0.0104	0	0.0127	0.0120	0.0319	0.0443	0.0344	0.0410	0.0539	0.0450	0.0354	0.0213	0.0427	0.0453	0.0340	0.0188	0.0147
6	0.0236	0.0170	0.0216	0.0160	0.0127	0	0.0155	0.0303	0.0409	0.0241	0.0292	0.0420	0.0343	0.0229	0.0093	0.0301	0.0340	0.0275	0.0121	0.0117
7	0.0258	0.0317	0.0367	0.0221	0.0120	0.0155	0	0.0200	0.0327	0.0280	0.0366	0.0563	0.0496	0.0368	0.0196	0.0426	0.0397	0.0241	0.0120	0.0240
8	0.0209	0.0412	0.0514	0.0417	0.0319	0.0303	0.0200	0	0.0130	0.0241	0.0341	0.0621	0.0594	0.0438	0.0278	0.0463	0.0341	0.0128	0.0188	0.0415
9	0.0240	0.0483	0.0606	0.0538	0.0443	0.0409	0.0327	0.0130	0	0.0267	0.0352	0.0638	0.0655	0.0493	0.0361	0.0497	0.0329	0.0153	0.0288	0.0524
10	0.0032	0.0234	0.0378	0.0400	0.0344	0.0241	0.0280	0.0241	0.0267	0	0.0100	0.0392	0.0392	0.0230	0.0153	0.0231	0.0117	0.0121	0.0162	0.0347
11	0.0132	0.0218	0.0371	0.0448	0.0410	0.0292	0.0366	0.0341	0.0352	0.0100	0	0.0310	0.0337	0.0187	0.0199	0.0155	0.0059	0.0217	0.0246	0.0379
12	0.0421	0.0249	0.0308	0.0513	0.0539	0.0420	0.0563	0.0621	0.0658	0.0392	0.0310	0	0.0130	0.0195	0.0370	0.0161	0.0352	0.0512	0.0464	0.0422
13	0.0416	0.0184	0.0187	0.0406	0.0450	0.0343	0.0496	0.0594	0.0655	0.0392	0.0337	0.0130	0	0.0162	0.0320	0.0189	0.0391	0.0503	0.0416	0.0317
14	0.0254	0.0066	0.0199	0.0347	0.0354	0.0229	0.0368	0.0438	0.0493	0.0230	0.0187	0.0195	0.0162	0	0.0175	0.0084	0.0245	0.0341	0.0270	0.0260
15	0.0154	0.0136	0.0245	0.0251	0.0213	0.0093	0.0196	0.0278	0.0361	0.0153	0.0199	0.0370	0.0320	0.0175	0	0.0230	0.0247	0.0213	0.0097	0.0194
16	0.0261	0.0148	0.0278	0.0428	0.0427	0.0301	0.0426	0.0463	0.0497	0.0231	0.0155	0.0161	0.0189	0.0084	0.0230	0	0.0204	0.0351	0.0317	0.0342
17	0.0144	0.0278	0.0430	0.0498	0.0453	0.0340	0.0397	0.0341	0.0329	0.0117	0.0059	0.0352	0.0391	0.0245	0.0247	0.0204	0	0.0213	0.0278	0.0433
18	0.0091	0.0331	0.0458	0.0421	0.0340	0.0275	0.0241	0.0128	0.0153	0.0121	0.0217	0.0512	0.0503	0.0341	0.0213	0.0351	0.0213	0	0.0158	0.0391
19	0.0144	0.0232	0.0326	0.0263	0.0188	0.0121	0.0120	0.0188	0.0288	0.0162	0.0246	0.0464	0.0416	0.0270	0.0097	0.0317	0.0278	0.0158	0	0.0237
20	0.0347	0.0194	0.0144	0.0090	0.0147	0.0117	0.0240	0.0415	0.0524	0.0347	0.0379	0.0422	0.0317	0.0260	0.0194	0.0342	0.0433	0.0391	0.0237	0

Şekil 3.a

2 nokta arasındaki mesafeyi ve bunları matriste gösterimini Matlab ortamındaki kodu Şekil 3.b'de gösterilmiştir.

```

cities = 20;

%bu noktaları ekran gösterme
plot(locations(:,1),locations(:,2),'bo');

%mesafeler için matris oluşturuldu ve 2 nokta arası mesafeler hesaplanıp
%matrise eklendi
distances = zeros(cities);
for count1=1:cities,
    for count2=1:cities,
        x1 = locations(count1,1);
        y1 = locations(count1,2);
        x2 = locations(count2,1);
        y2 = locations(count2,2);
        distances(count1,count2)=sqrt((x1-x2)^2+(y1-y2)^2);
        distances(count2,count1)=distances(count1,count2);
    end;
end;

```

Şekil 3.b

Genetik algoritma çözümünde kullanılan parametreler ve seçilen yöntemler:

Bu ödevde gezgin satıcı problemi baz alınmıştır. Gezgin satıcı problemi nedir? Bir seyyar satıcı var ve bu satıcı, mallarını n şehirde satmak istiyor. ($n > 0$) Herhangi bir şehre gitmek bir maliyeti olacağından. Mümkün olan en kısa şekilde tüm şehirlere gitmeyi amaçlayan bir problemdir. Burada ilk gideceği şehri seçmesi gerekiyor. n tane şehir olduğuna göre ilk şehre geldiğinde $n-1$ tane farklı şehir kalır. Bu şehirlerden bir tanesini seçtiğinde, $n-2$ tane farklı şehir kalır. Tüm şehirler bitene kadar böyle devam eder. Seçim olayı olduğu için sonuç olarak $n!$ kadar farklı seçim yapılabilmektedir. Bu seçimler arasında en kısa sürede tamamlananı bulmamız gerekiyor.

Bu ödevde MATLAB'de bulunan genetik algoritma çözücünü kullandım. Bu genetik algoritma çözücü isteğe göre düzenlenebilmektedir. İsteğimize göre düzenleyebilmek için oluşturma fonksiyonu, çaprazlama fonksiyonu ve mutasyon fonksiyonu oluşturmamız gerekiyor.

Oluşturma fonksiyonu, Şekil 4'de belirtilmiştir.

```
1 function pop = create_permutations(NVARS, FitnessFcn, options)
2 % NVARS: Degisken sayisi
3 % FitnessFcn: Uygunluk fonksiyonu
4 % options: Genetik Algoritma için kullanılan seçenekler yapisi
5
6 totalPopulationSize = sum(options.PopulationSize);
7 n = NVARS;
8 pop = cell(totalPopulationSize, 1);
9 for i = 1:totalPopulationSize
10     pop{i} = randperm(n);
11 end
```

Şekil 4

Çaprazlama fonksiyonu, Şekil 5'de gösterilmiştir. Burada hücre dizisinin, popülasyonunu alır ve çaprazlama sonucu elde edilen çocukları, bir hücre dizisini dönüştürür. Burada tek noktalı çaprazlama uygulanmıştır.

```
1 function xoverKids = crossover_permutation(parents, options, NVARS, ...
2     FitnessFcn, thisScore, thisPopulation)
3
4 % Parents: secilen ebeveynler
5 % options: OPTIMOPTIONS tarafından olusturulan options
6 % NVARS: verilerinin sayisi
7 % FitnessFcn: uygunluk fonksiyonu
8 % thisScore: Mevcut nufusun puanlar?n?n vektoru
9 % thisPopulation: Mevcut populyasyondaki bireylerin matrisi
10
11 nKids = length(parents)/2;
12 xoverKids = cell(nKids, 1);
13 index = 1;
14
15 for i=1:nKids
16     parent = thisPopulation{parents(index)};
17     index = index + 2;
18
19     p1 = ceil((length(parent) - 1) * rand);
20     p2 = p1 + ceil((length(parent) - p1 - 1) * rand);
21     child = parent;
22     child(p1:p2) = fliplr(child(p1:p2));
23     xoverKids{i} = child;
24
25 end
```

Şekil 5

Mutasyon fonksiyonu, Şekil 6’da gösterilmiştir. Bu fonksiyonda belirtilen noktalardan birini alır ve mutasyona uğramış sıralı bir kümeye dönüştürür.

```

1 function mutationChildren = mutate_permutation(parents ,options,NVARS, ...
2     FitnessFcn, state, thisScore,thisPopulation,mutationRate)
3
4 % parents: Secilen ebeveynler
5 % options: OPTIMOPTIONS tarafından olusturulan options
6 % NVARS: verilerinin sayisi
7 % FitnessFcn: uygunluk fonksiyonu
8 % state: Genetik algoritma tarafından kullanılan durum yapisi
9 % thisScore: Mevcut nufusun puanlarinin vektoru
10 % thisPopulation: Mevcut populasiyondaki bireylerin matrisi
11 % mutaionRate: Mutasyon orani
12
13 mutationChildren = cell(length(parents),1);
14 for i=1:length(parents)
15     parent = thisPopulation{parents(i)};
16     p = ceil(length(parent) * rand(1,2));
17     child = parent;
18     child(p(1)) = parent(p(2));
19     child(p(2)) = parent(p(1));
20     mutationChildren{i} = child;
21 end

```

Şekil 6

Seyyar satıcı probleminde bize uygunluk fonksiyonu da gereklidir. Toplam ne kadar mesafe gittiğimize göre uygunluğunu düzenliyorum. Bu fonksiyon Şekil 7’de gösterilmiştir.

```

1 function scores = traveling_salesman_fitness(x,distances)
2
3 scores = zeros(size(x,1),1);
4 for j = 1:size(x,1)
5     p = x{j};
6     f = distances(p(end),p(1));
7     for i = 2:length(p)
8         f = f + distances(p(i-1),p(i));
9     end
10    scores(j) = f;
11 end

```

Şekil 7

Daha sonra bunu ana kodumda çağırıyorum. Bu çağırma işlemi Şekil 8’de gösterilmiştir.

```

43 - FitnessFcn = @(x) traveling_salesman_fitness(x,distances);

```

Şekil 8

Bu yapılan işlemleri ve yol haritasını çıktı olarak görebilmek için `traveling_salesman_plot.m` fonksiyonunu kullandım. Fonksiyonun kodları Şekil 9'dadır.

```

1 function state = traveling_salesman_plot(~, state, flag, locations)
2
3 -     [~,i] = min(state.Score);
4 -     genotype = state.Population{i};
5
6 -     plot(locations(:,1),locations(:,2),'go');
7 -     plot(locations(genotype,1),locations(genotype,2));
8 -     hold off

```

Şekil 9

Ardından Şekil 9'daki fonksiyonu ana kodda çağırıyorum. Bu çağırma işlemi Şekil 10'da gösterilmiştir.

```

46 -     my_plot = @(options,state,flag) traveling_salesman_plot(options, ...
47 -         state,flag,locations);
48

```

Şekil 10

Kodlarım hazır. Şimdi Genetik algoritmanın ayarlarını kurma aşamasına geçiyorum. Burada tanımlamış olduğum fonksiyonları genetik algoritma ayarlarında kullanacağım. Öncelikle, veri türümü ve popülasyon aralığımı belirtmek için seçenek kapsayıcısı oluşturdum. Bu oluşturma Şekil 11'de gösterilmiştir.

```

49 -     options = optimoptions(@ga, 'PopulationType', 'custom','InitialPopulationRange', ...
50 -         [1;cities]);

```

Şekil 11

Ardından oluşturduğum seçenek kapsayıcısını, çaprazlamayı, mutasyonu ve çizim fonksiyonlarını ve durdurma koşullarını ayarladığım kod satırını ana koduma yerleştiriyorum. Aynı zamanda başlangıç popülasyonunu 60 belirttim. Bu kod satırı Şekil 12'de gösterilmiştir.

```

52     %başlangıç popülasyonu 60 belirtildi
53 -     options = optimoptions(options,'CreationFcn',@create_permutations, ...
54 -         'CrossoverFcn',@crossover_permutation, ...
55 -         'MutationFcn',@mutate_permutation, ...
56 -         'PlotFcn', my_plot, ...
57 -         'MaxGenerations',500,'PopulationSize',60, ...
58 -         'MaxStallGenerations',200,'UseVectorized',true);

```

Şekil 12

Artık genetik algoritmayı çağırabilirim. Bu çağırma işlemini Şekil 13'de gösterilmiştir.

```

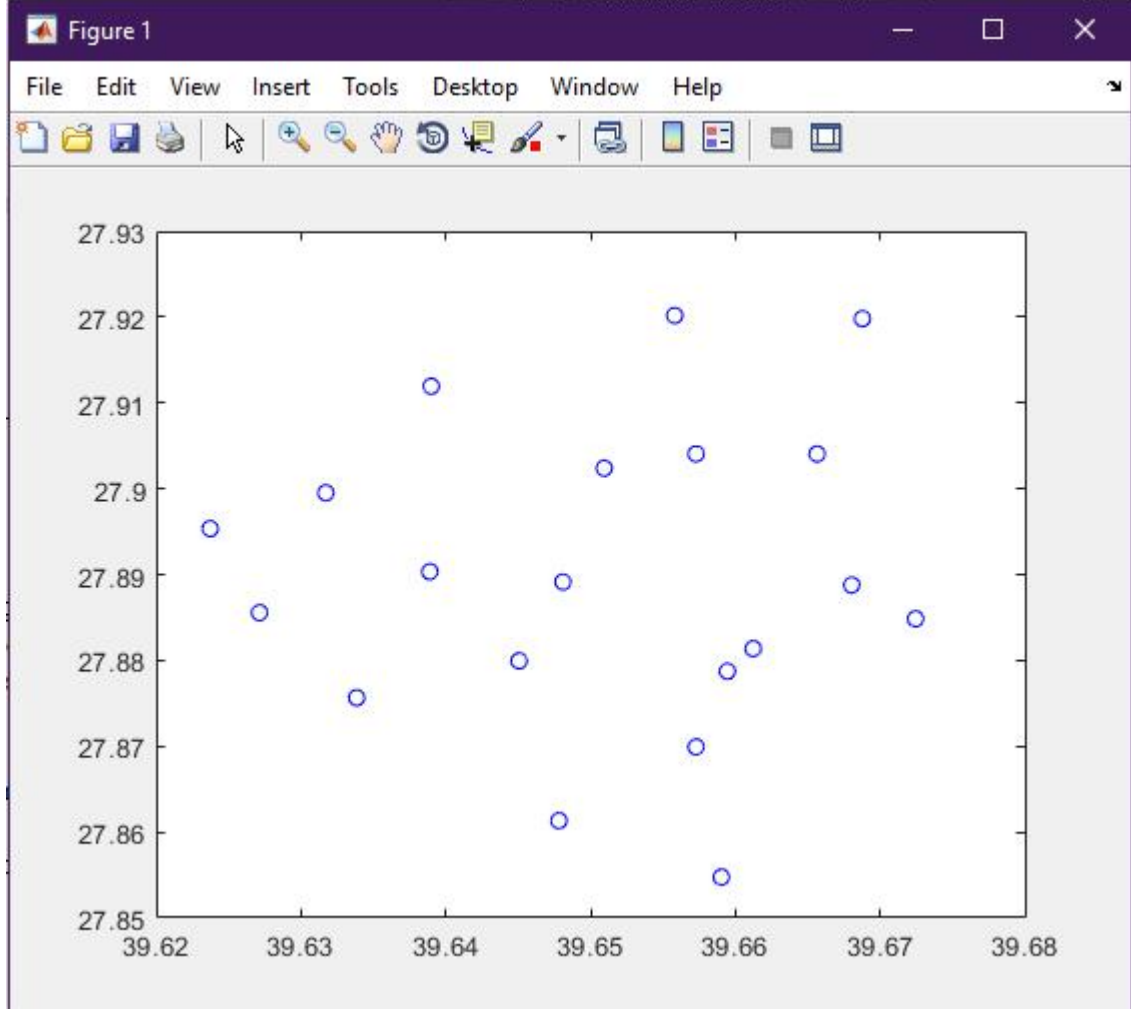
61 -     [x,fval,reason,output] = ...
62 -         ga(FitnessFcn,numberofVariables,[],[],[],[],[],[],[],options)

```

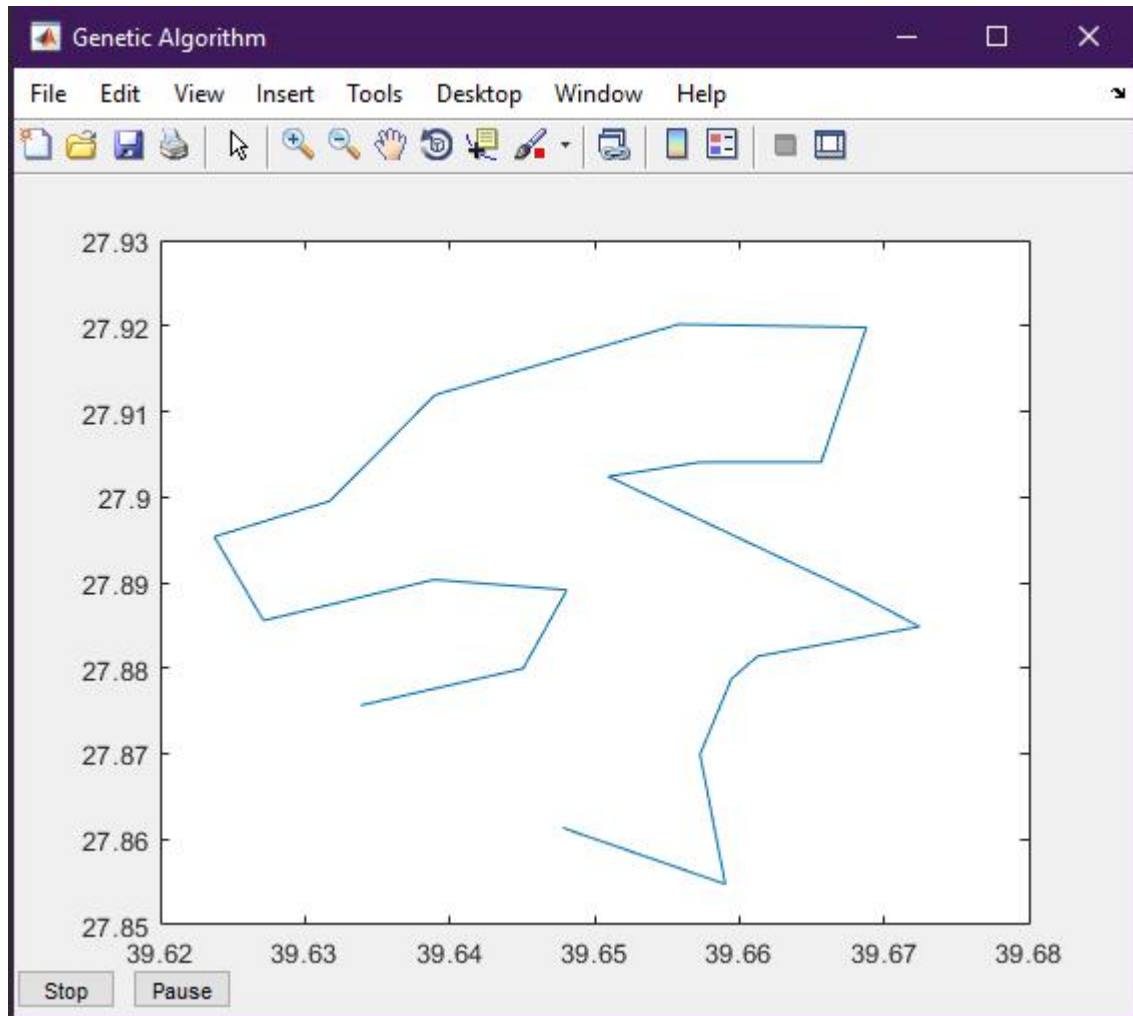
Şekil 13

Sonuç

Programı çalıştırdığımda öncelikle koordinatlarını belirttiğim noktaları gösteriyor. Şekil 14’de gösterilmiştir. Ardından en iyi yol haritasını oluştururken denemeleri program çalışırken görebiliyoruz. En sonunda en iyi yol haritasını bulduğunda duruyor. Bu aşamadaki ekran çıktısı Şekil 15’de gösterilmiştir. Aynı zamanda Command Window çıktısı Şekil 16’da gösterilmiştir.



Şekil 14



Şekil 15

```
>> main
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.

x =

1x1 cell array

    {1x20 double}

fval =

    0.2401

reason =

    1

output =

struct with fields:

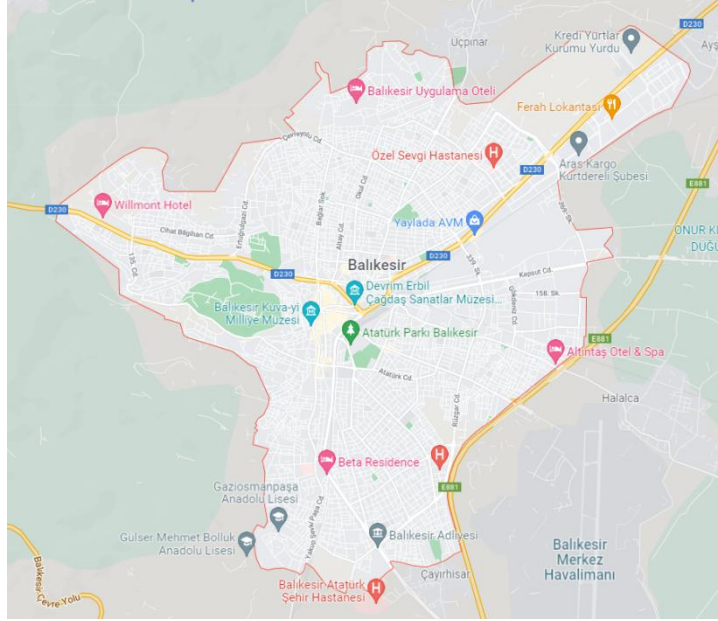
    problemtype: 'unconstrained'
    rngstate: [1x1 struct]
    generations: 294
    funccount: 17700
    message: 'Optimization terminated: average change in the fitness value less than options.FunctionTolerance.'
    maxconstraint: []

fx >>
```

Şekil 16

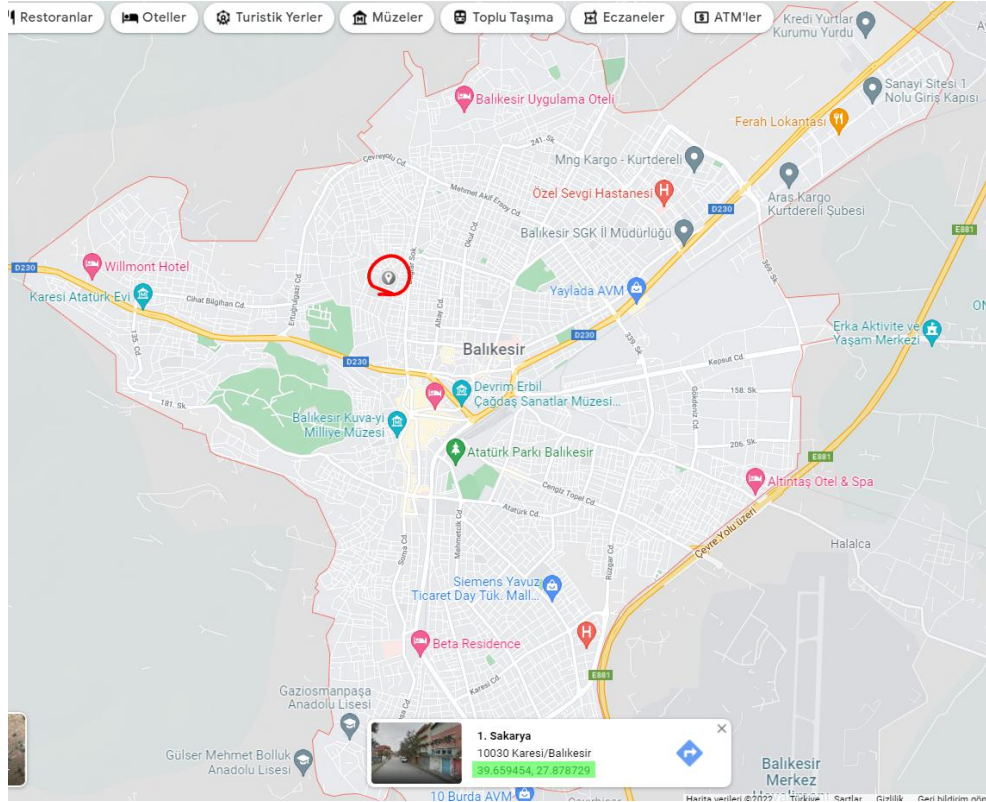
Google Maps çıktıları

Balıkesir Merkez ilçesinin haritası: (Şekil 17)

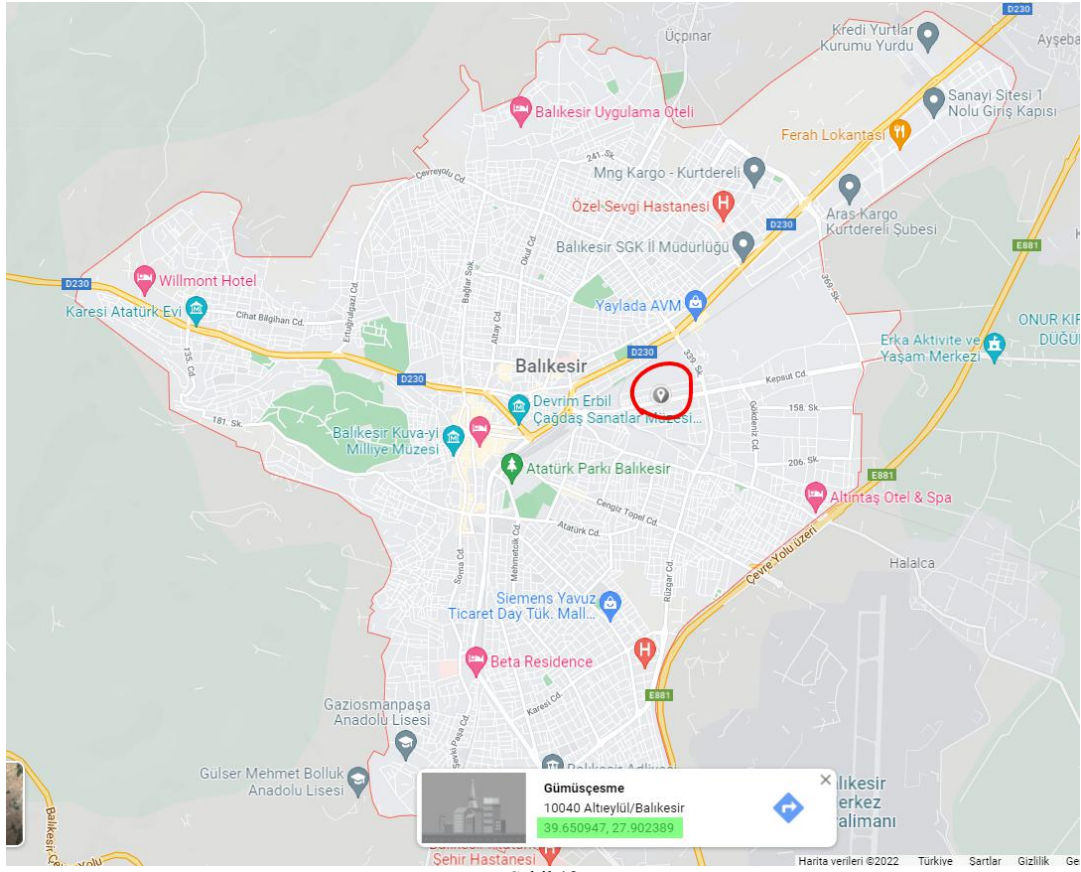


Şekil 17

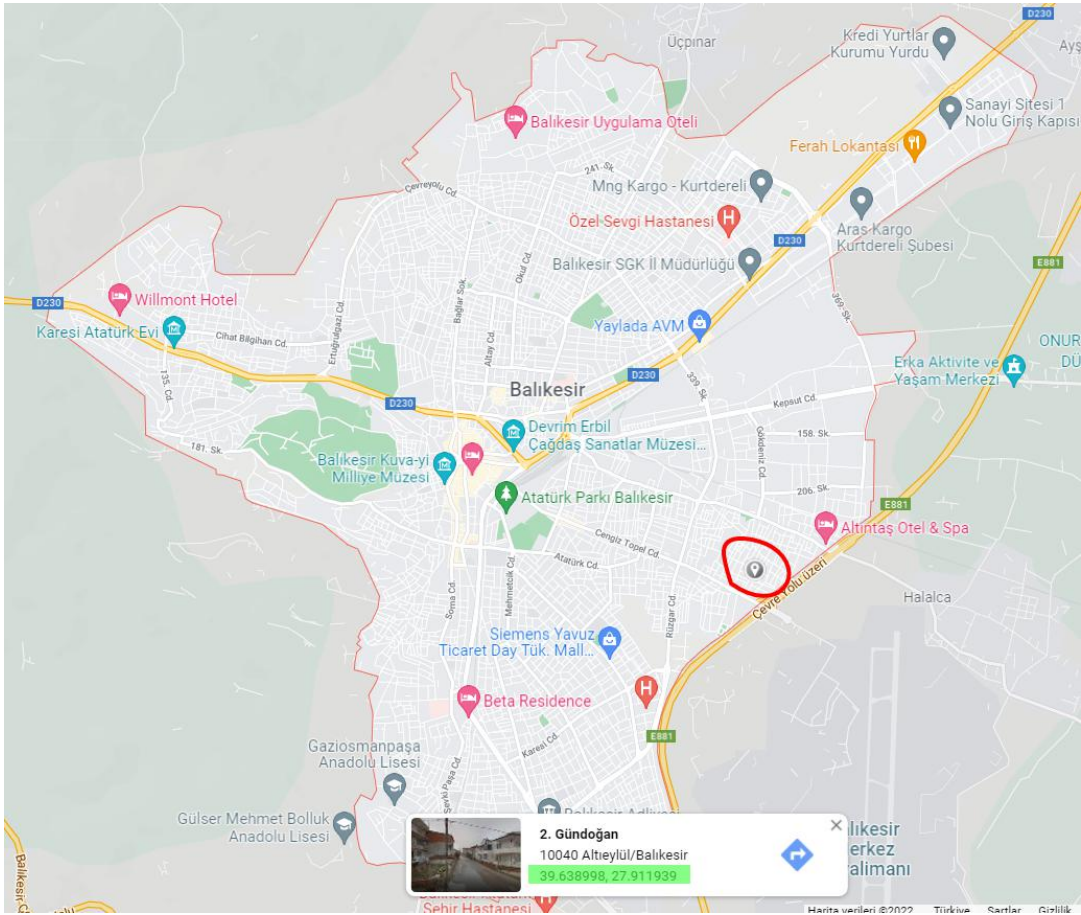
Seçilen 1'den 20'ye kadar olan noktaların ekran çıktıları: (Şekil 18- 36) Kırmızı daireler, varılması istenilen noktayı gösteren imlecin anlaşılabilmesi için çizilmiştir. Altta yeşil kutucuk içine alınan sayılar ise x ve y koordinatlarını belirtiyor.



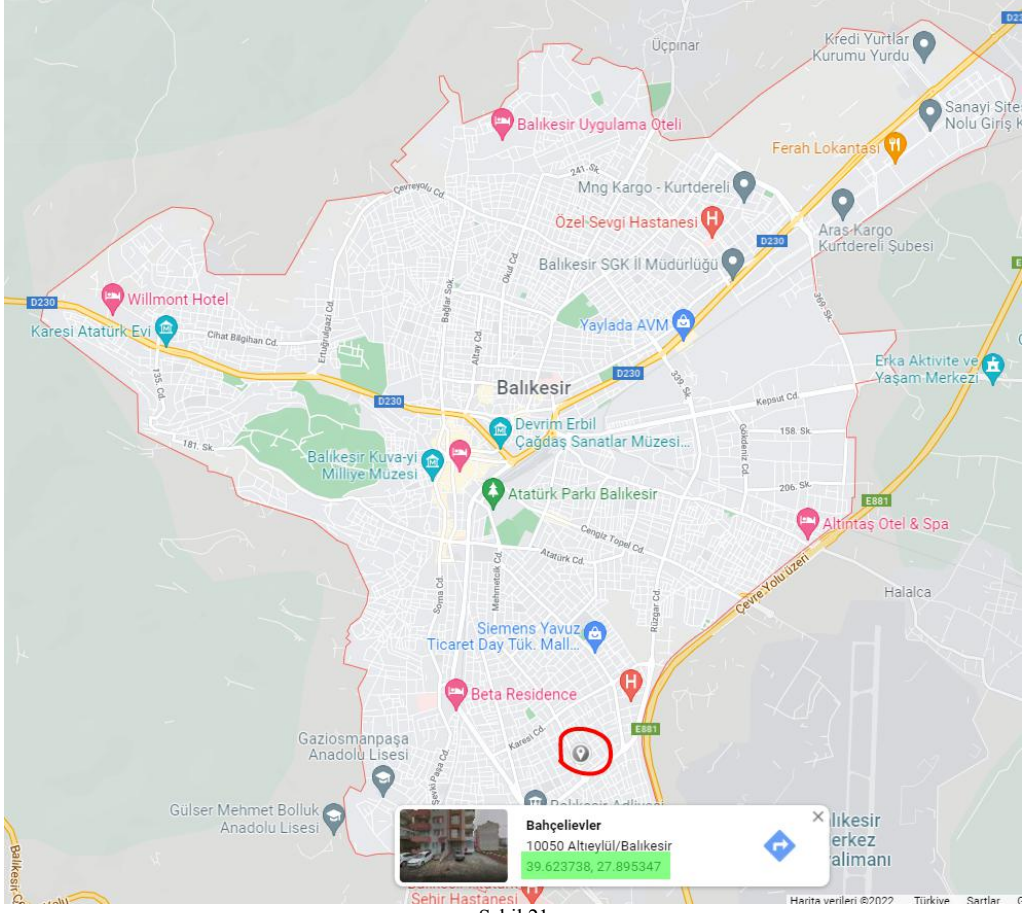
Şekil 18



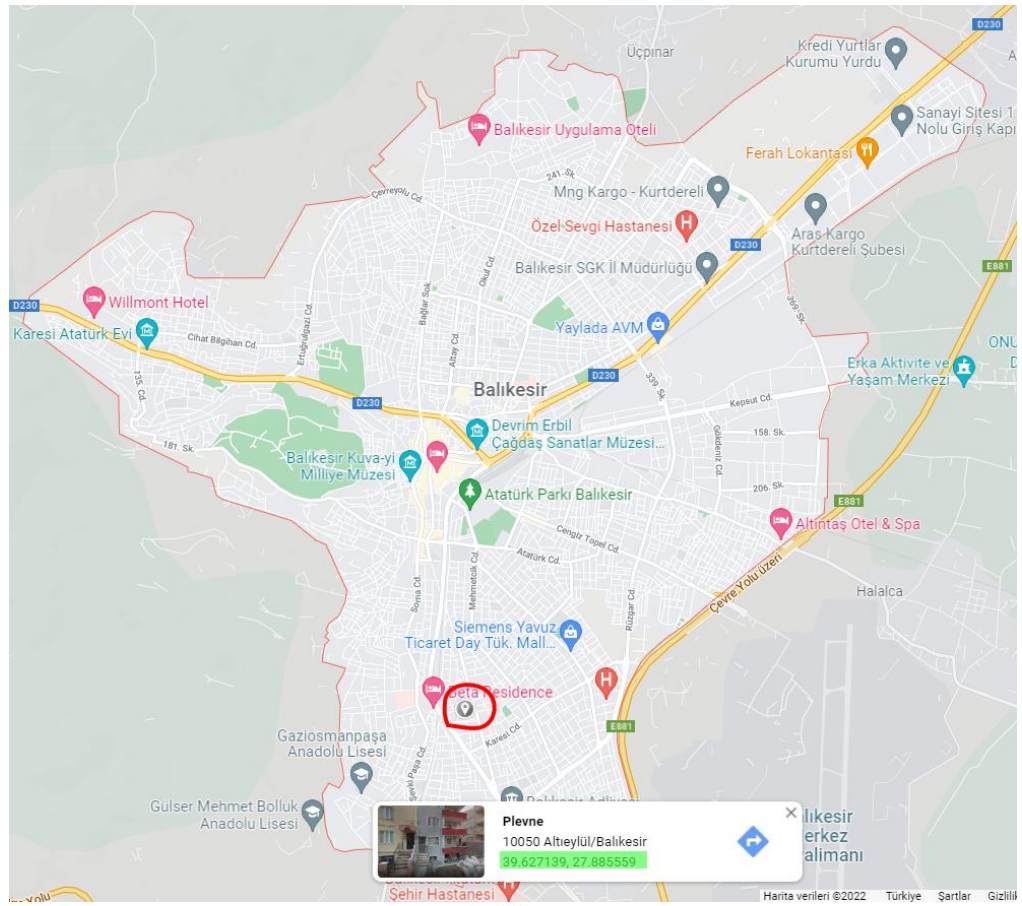
Şekil 19



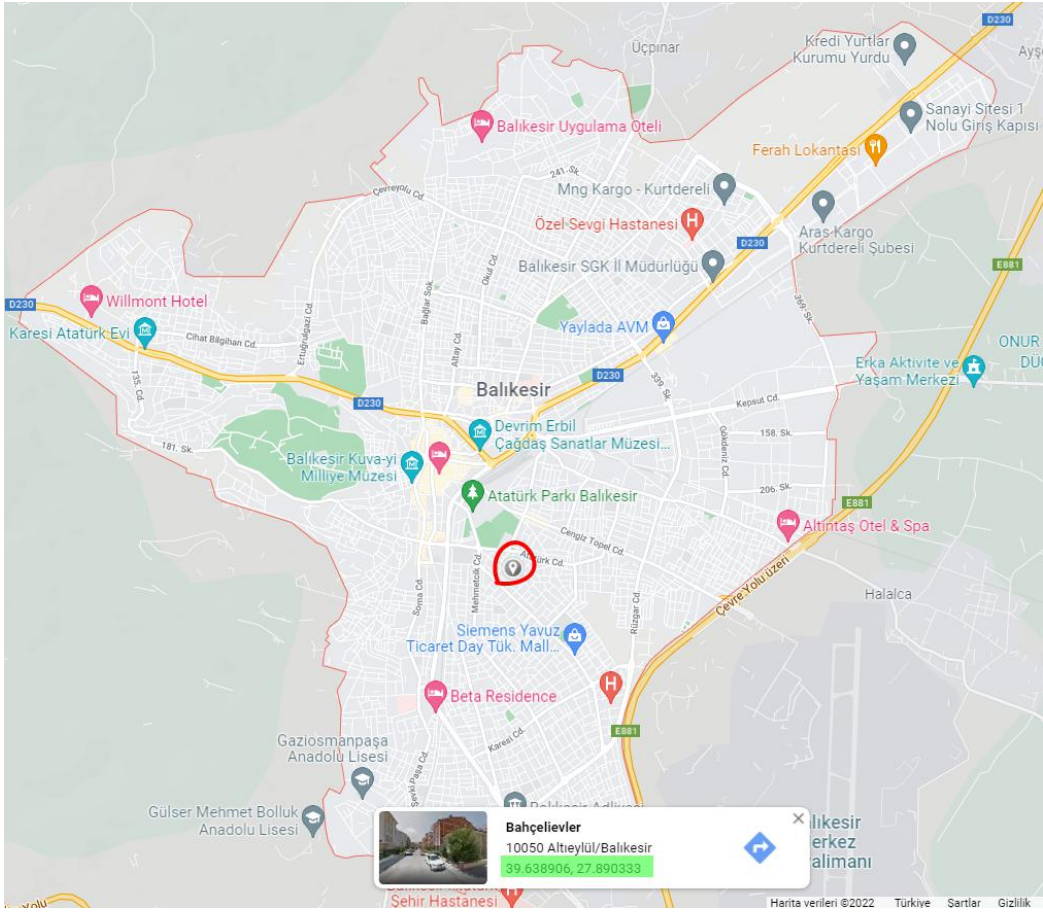
Şekil 20



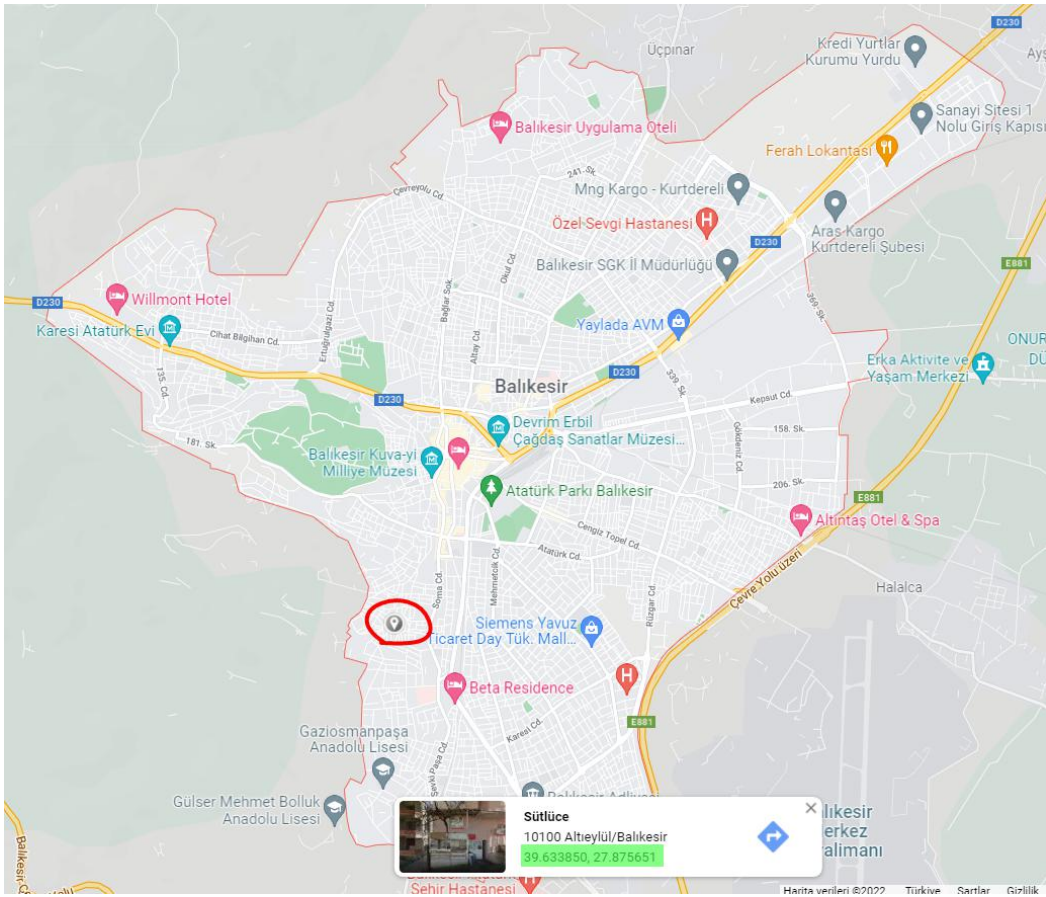
Şekil 21



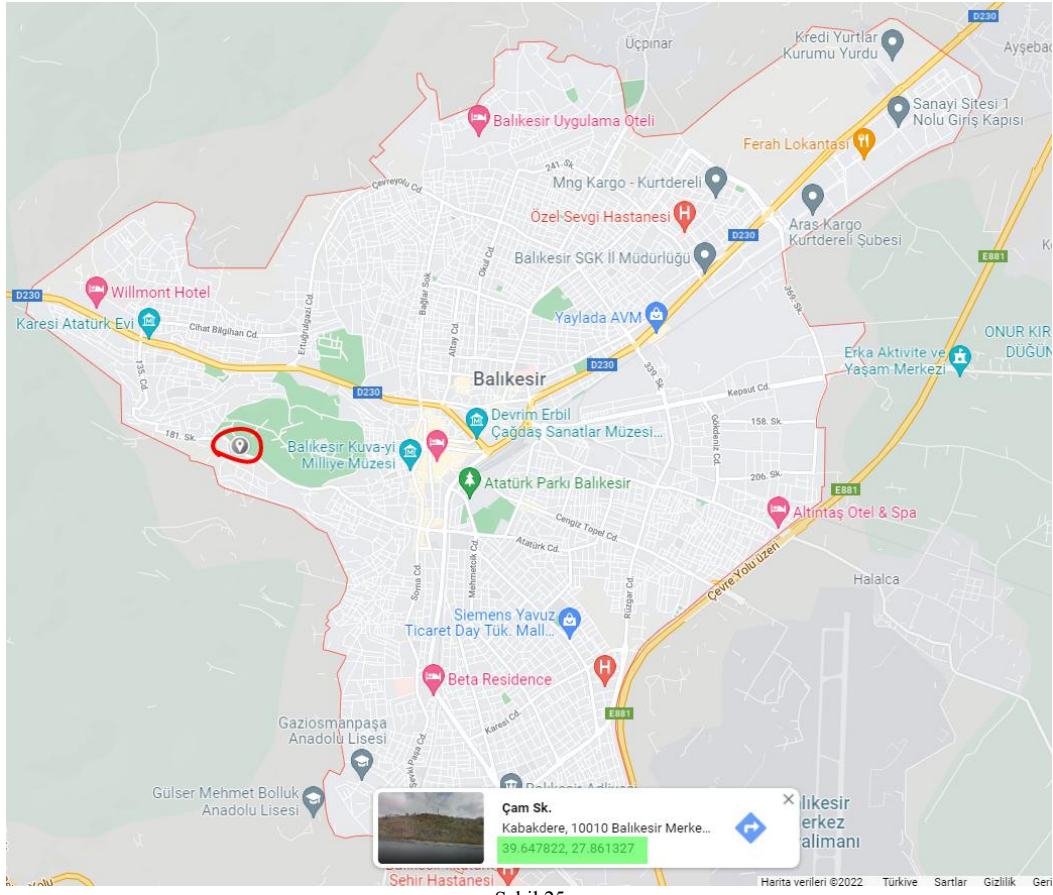
Şekil 22



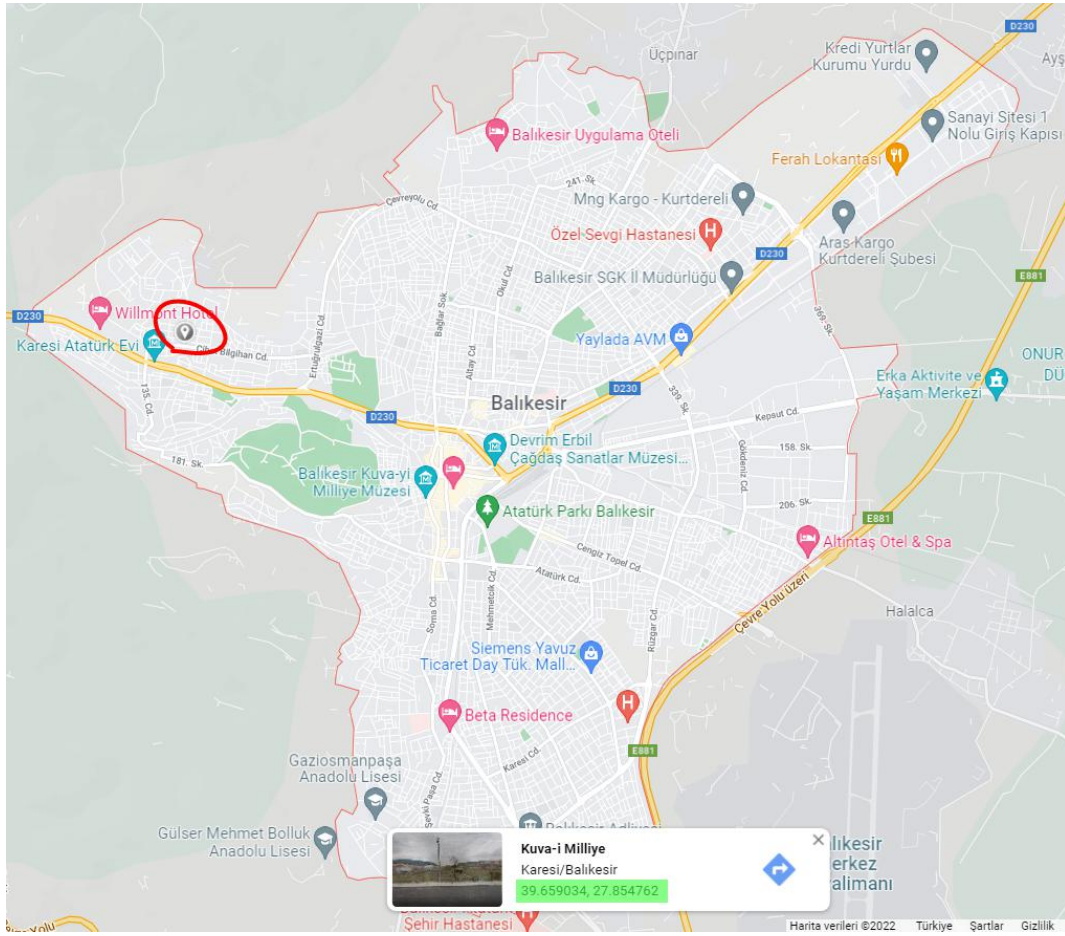
Şekil 23



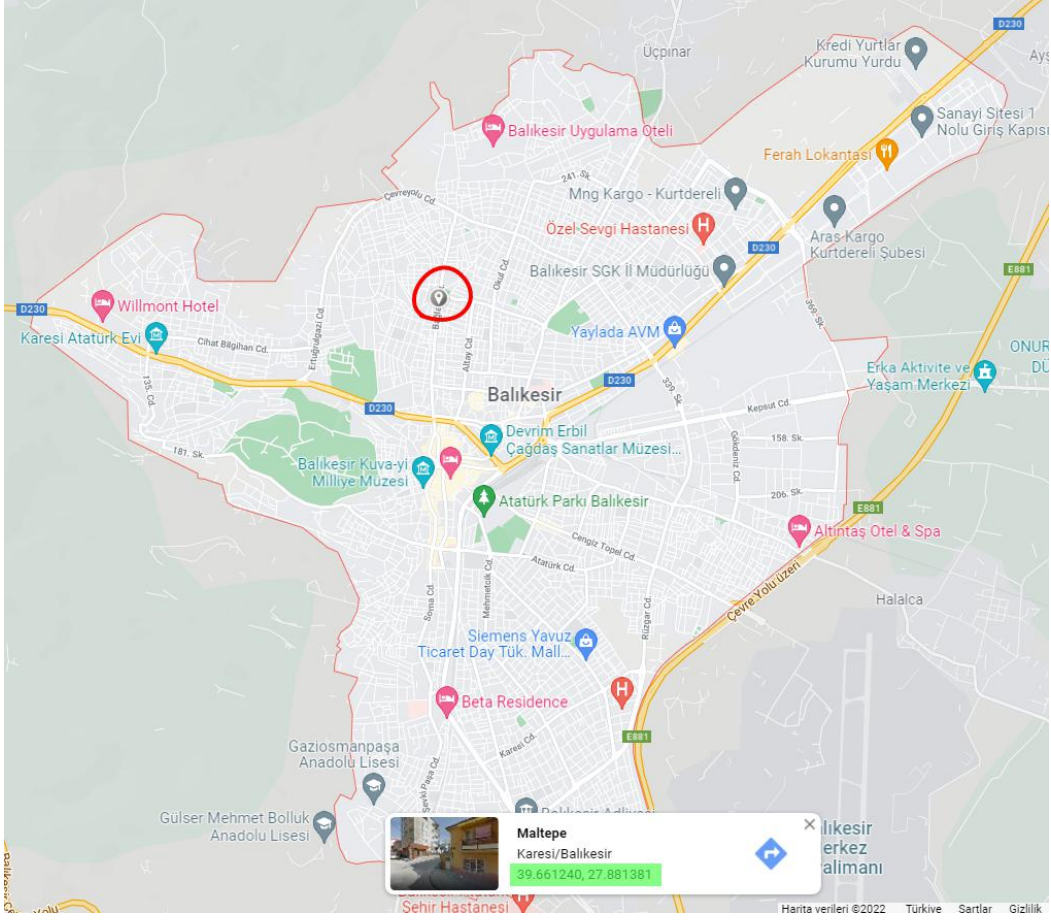
Şekil 24



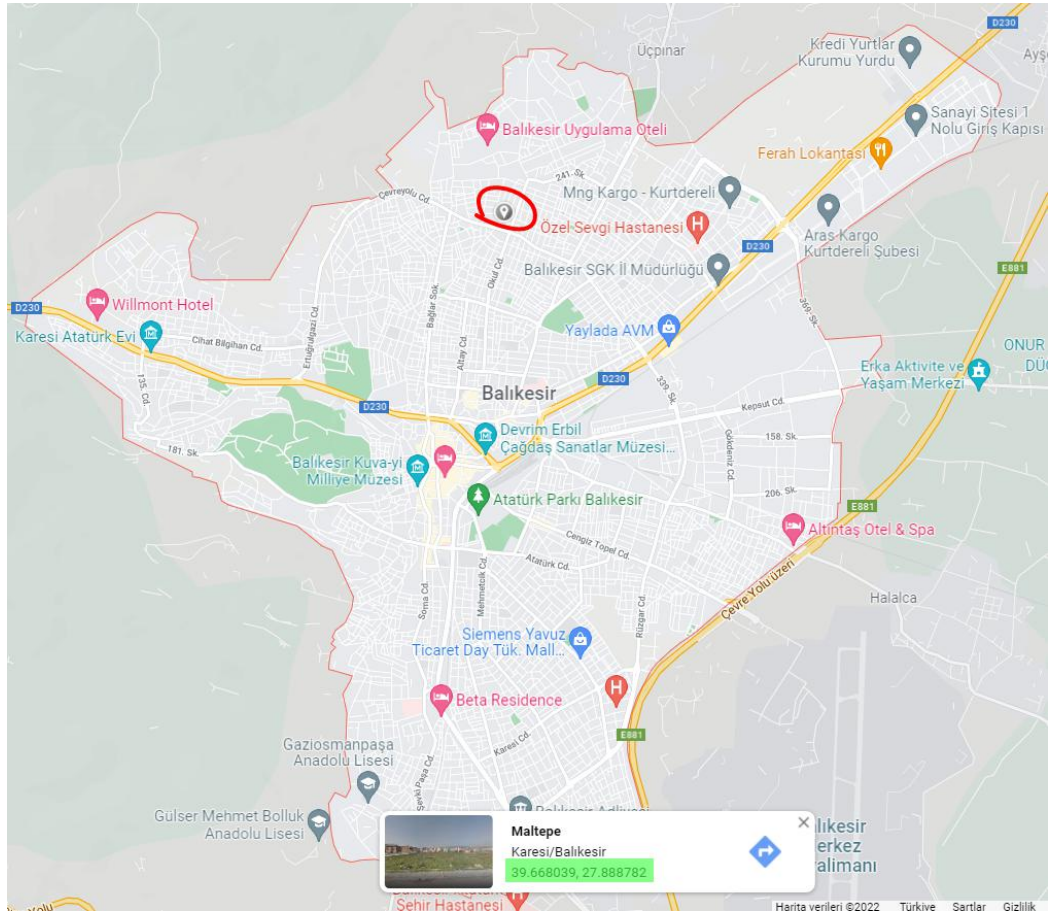
Şekil 25



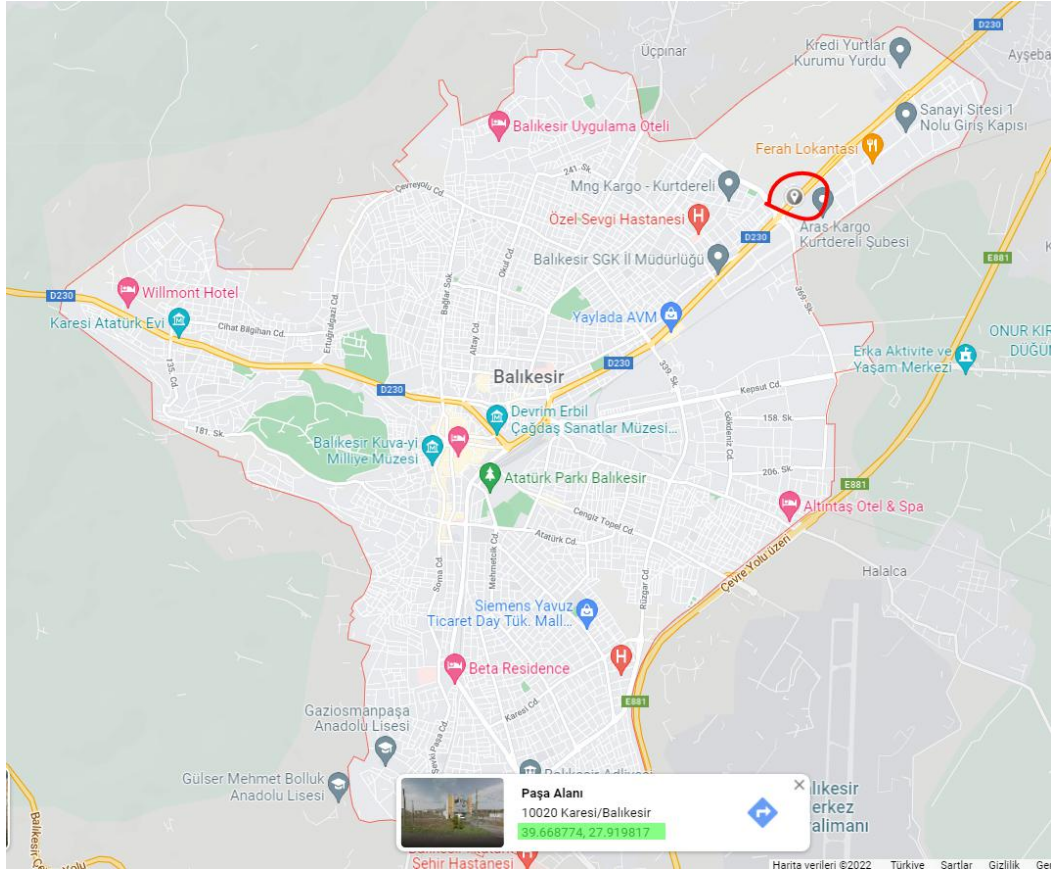
Şekil 26



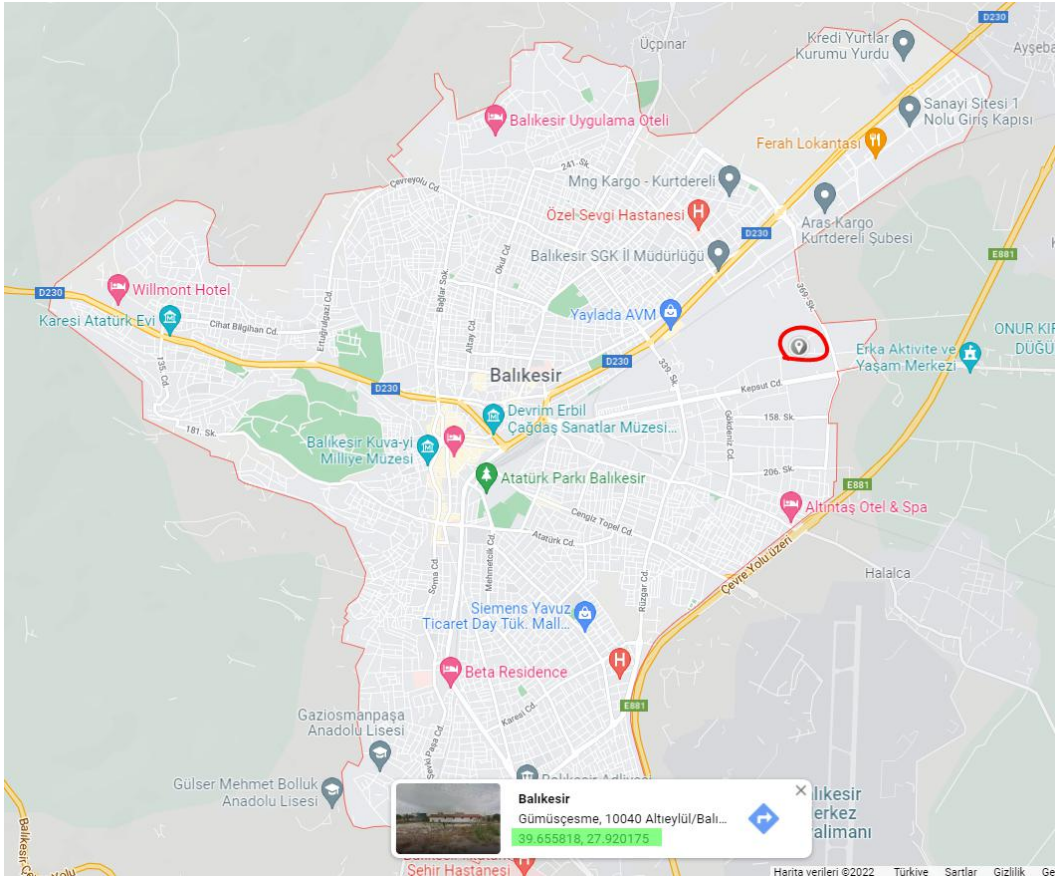
Şekil 27



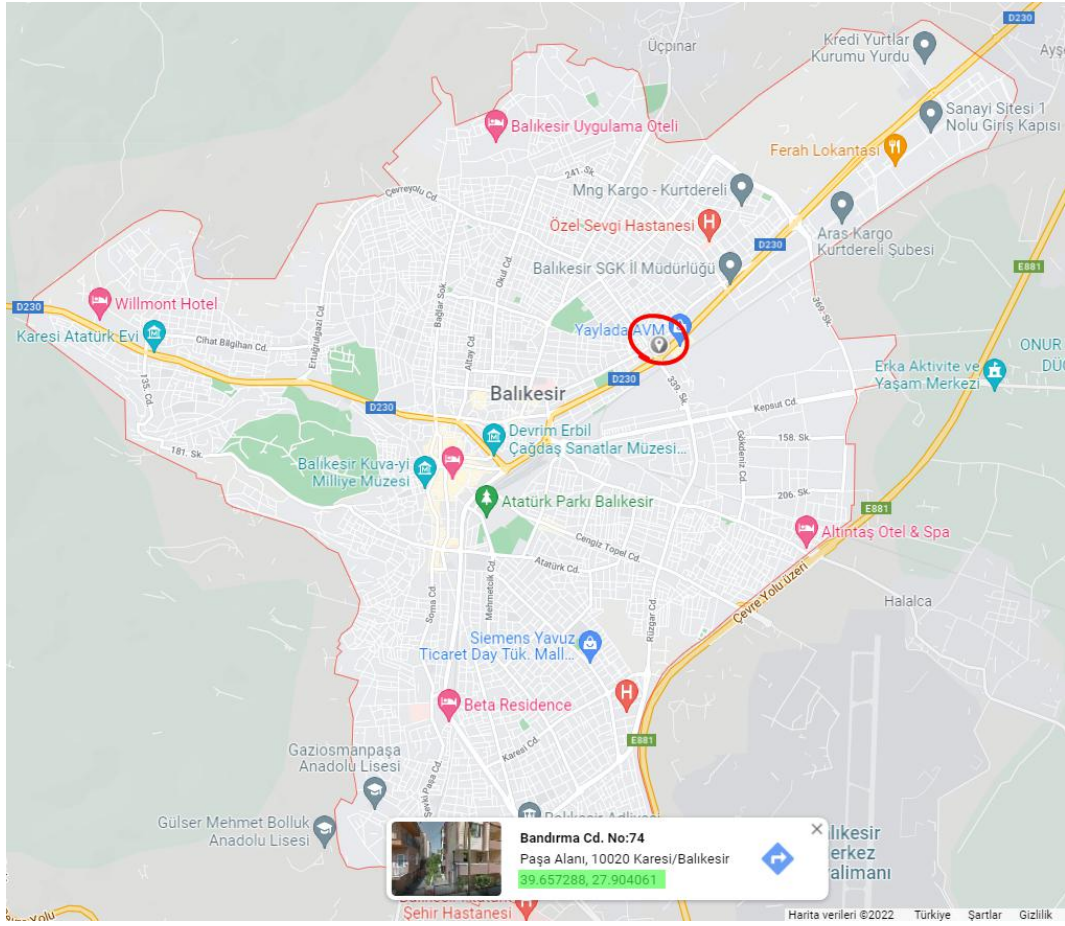
Şekil 28



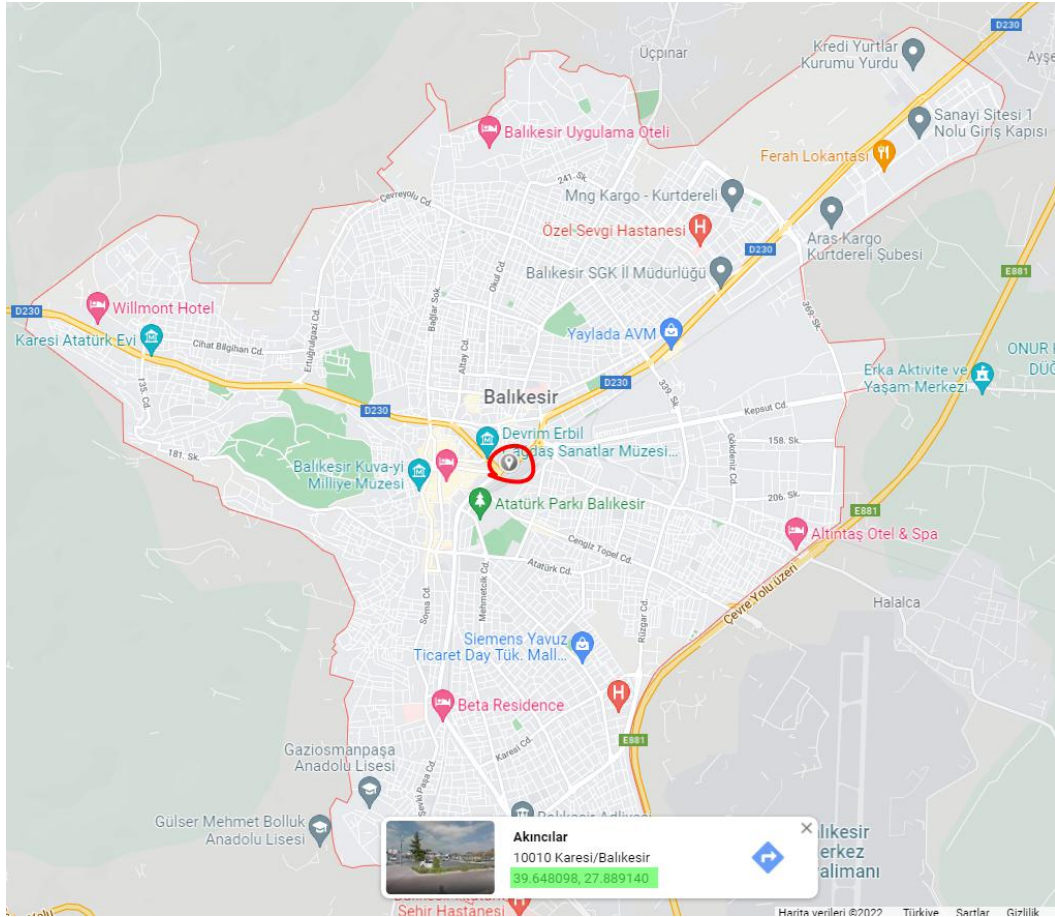
Şekil 29



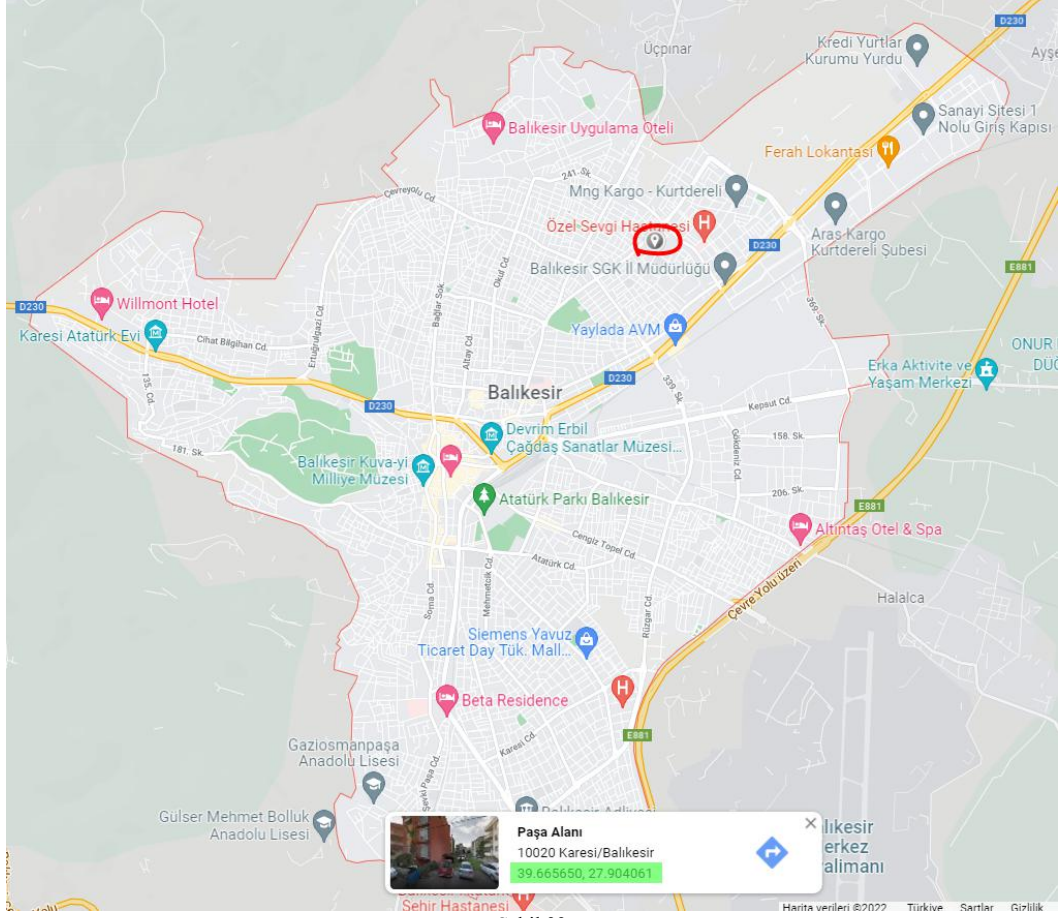
Şekil 30



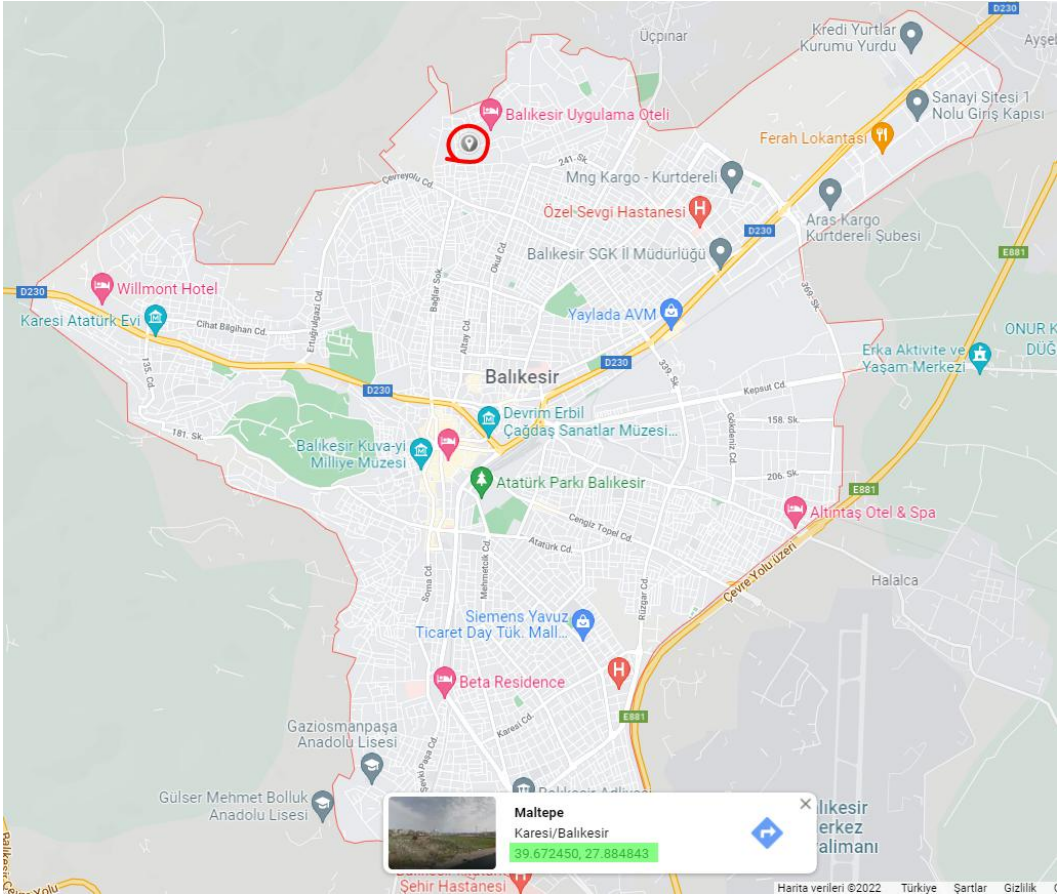
Şekil 31



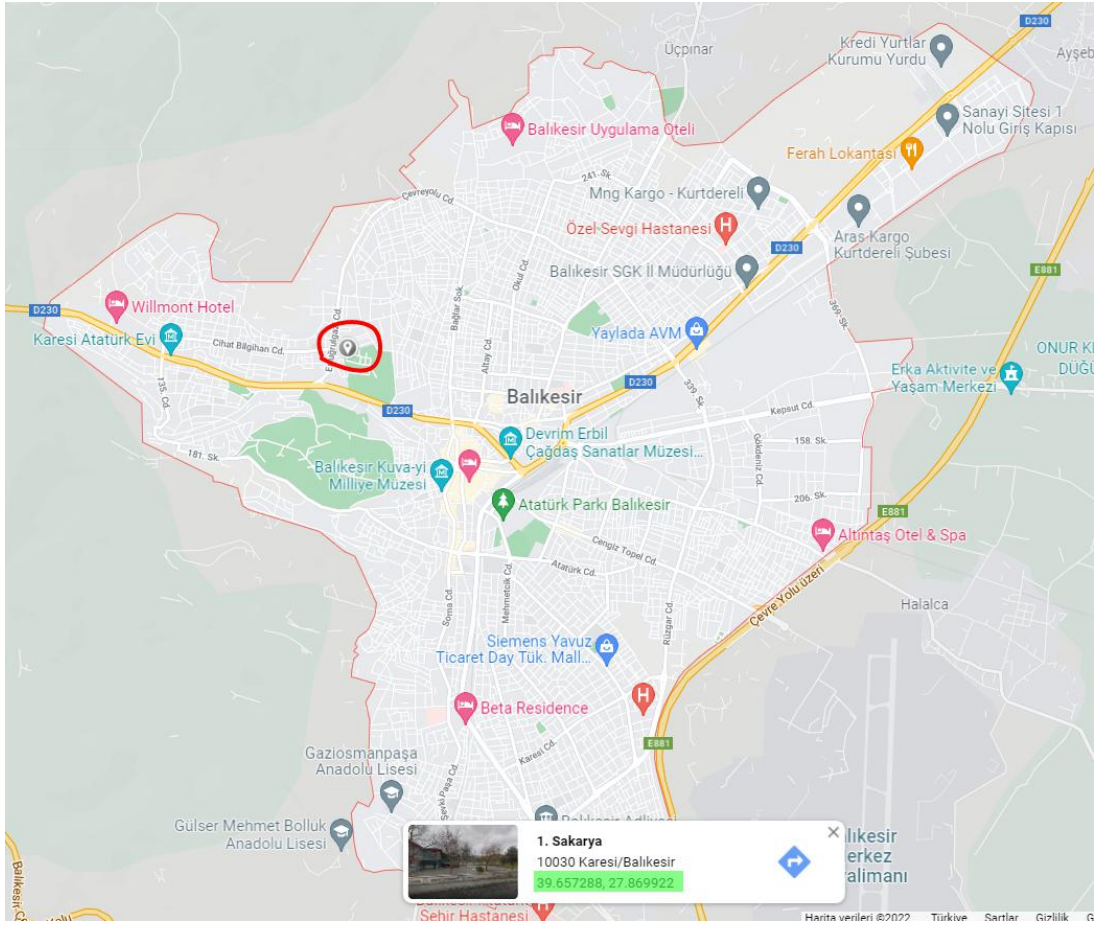
Şekil 32



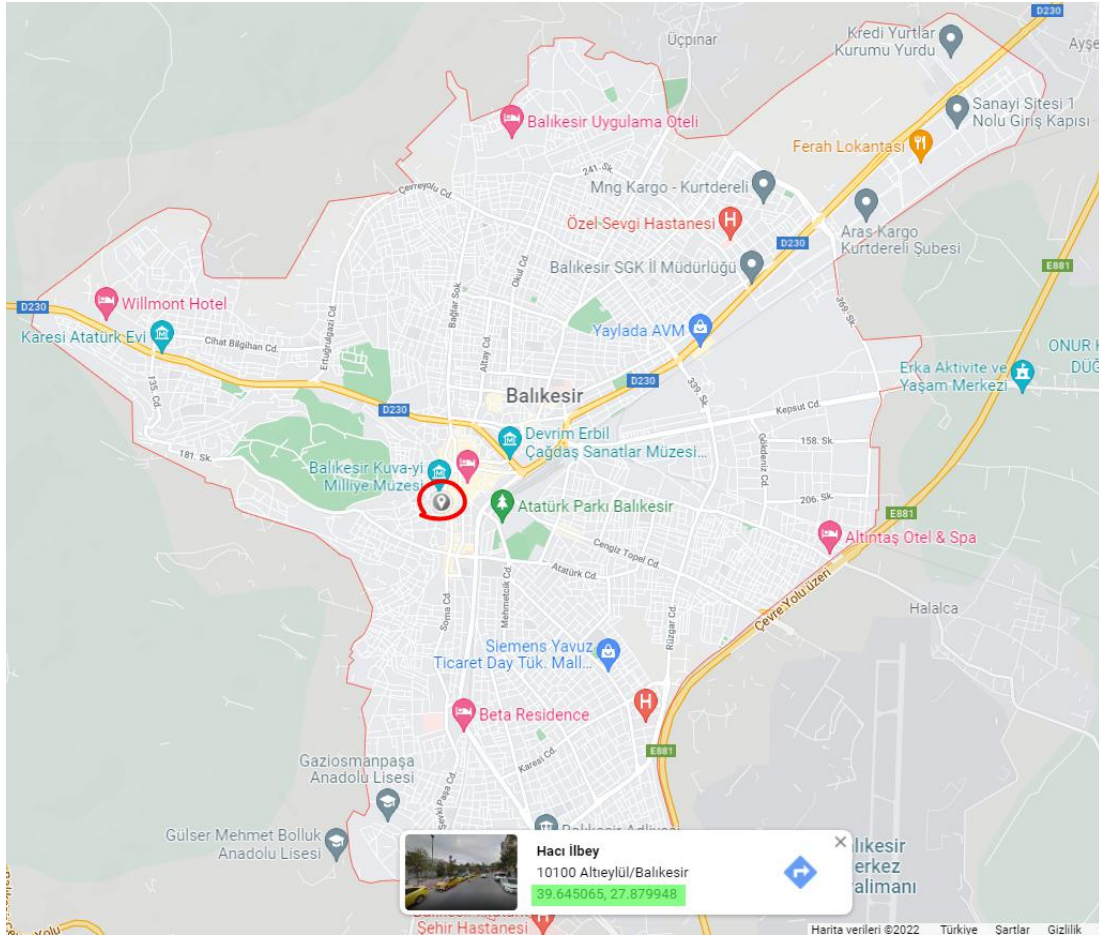
Şekil 33



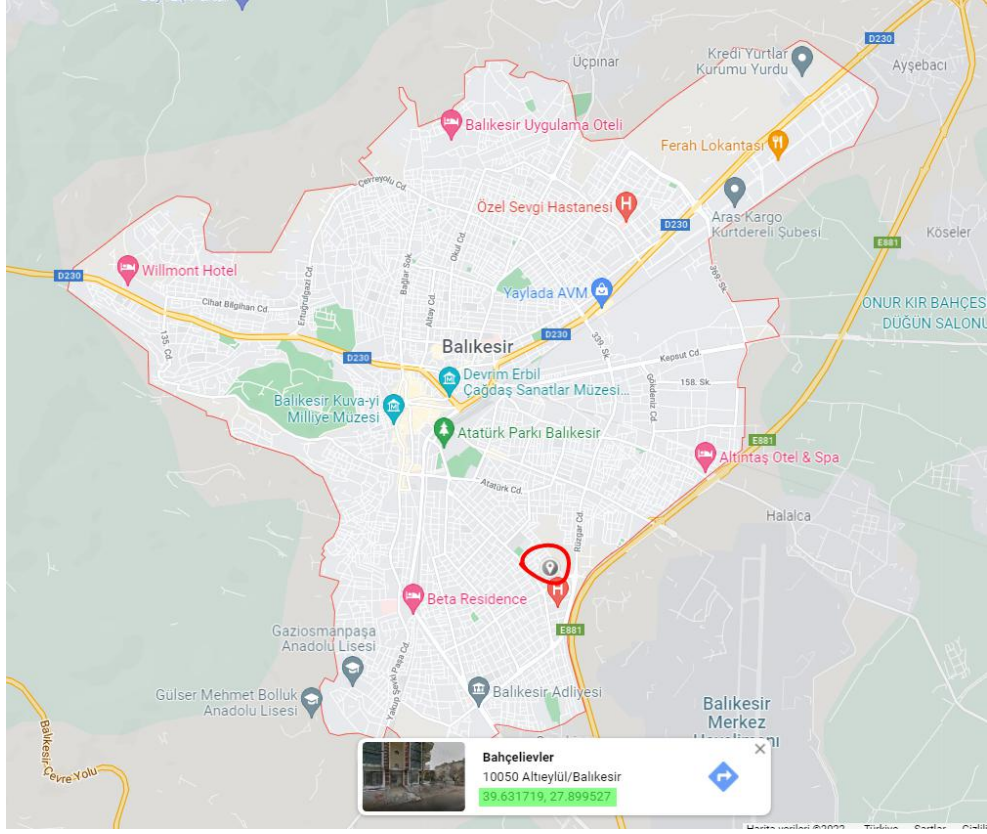
Şekil 34



Şekil 35



Şekil 36



Şekil 37