

Yapay Zeka 1. Ödev

Genetic Algorithms

Azad Yaşar 13011048

30 Ekim 2017

1 Giriş

Bu problemde genetik algoritma kullanılarak bir labirentin merkezinden sol en üstteki hücreğine giden yol bulunmaya çalışılmıştır.

2 Bulgular

2.1 Population Size İncelenmesi

Bu denemede **mutation rate** = %20, **labirent size** = 15×15 ve labirentteki engellerin labirentin boyutuna oranı %10'dur. Her population size için 10 deneme yapılmıştır ve μ jenerasyon dizisinin ortalaması, σ jenerasyon dizisinin standart sapmasıdır.

Population Size			
	50	250	500
μ	796.3	8	49.90
σ	877.22	4.82	134.13

2.2 Mutation Rate İncelemesi

Bu denemede **population size** = 250, **labirent size** = 15×15 ve labirentteki engellerin labirentin boyutuna oranı %10'dur.

Mutation Rate			
	%10	%30	%50
μ	2195	72.79	29.6
σ	2110.51	135.74	57.585

2.3 Labirent Engel Oranı İncelemesi

Bu denemede **population size** = 250, **labirent size** = 15×15 ve **mutation rate** = %50'dır.

Engel Oranı			
	%0	%10	%15
μ	26,69	45.2	627.79
σ	21.44	67.325	775.81

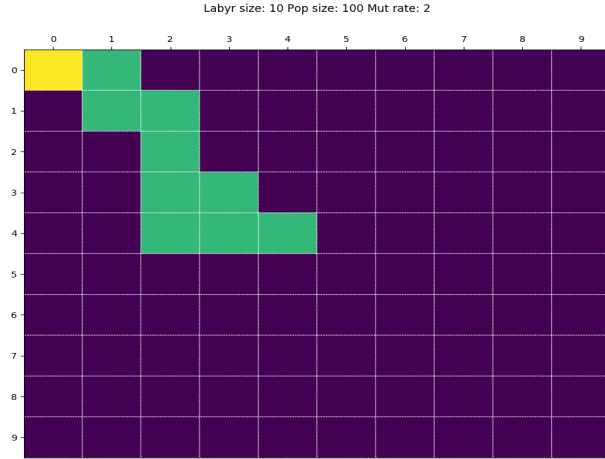
3 Zorluklar ve Çözümler

Her bireyin fitness değerini, o bireyin en son bulunduğu noktanın hedefe olan mesafesine bağlı olarak almak bazı problemler ortaya çıkarabilmektedir. Örneğin birey hedefe iyice yaklaşıp daha sonra uzaklaşmış olabilir. Bu sorunu çözmek için bireylerin izledikleri yolda hedefe en yakın noktaya kadar olan değerleri tutuldu. Doğal olarak bireylerin uzunlukları değişkenlik gösterebilmektedir. Fakat sürekli birey boyutunun belirli bir kısmını atarak bir zaman sonra bireyin uzunluk kapasitesi problemi çözemeyecek seviyeye düşebilir. Bunun için popülasyondaki bireylerin uzunlukları kontrol edilmeli eğer belirli bir eşik değerinin altında ise yeni bir popülasyon oluşturulmalıdır. Ayrıca değişken uzunluklu bireylerde **reproduce** işlemi yaparken yeni bir yöntem kullanılmıştır. Yöntem hem kısaltılan birey uzunluğuna çözüm bulmakta hem de sonuca daha çabuk ulaştırmaktadır. (labirent boyutu 40'tan büyük problemler bu şekilde çözülmüştür). Bu yöntemde, ind_1 ve ind_2 iki birey olmak üzere ve $path$ değişkeni bu bireylerin verilerini göstermekte olsun.

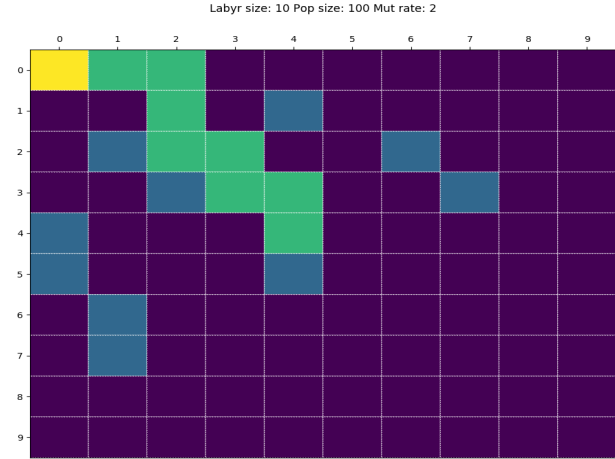
Algorithm 1 Sistem Akışı

```
1: procedure REPRODUCE(ind1, ind2)
2:   init new_child
3:   loc  $\leftarrow$  random(0, length(ind1.path))
4:   for i = 0 to loc do
5:     new_child.path.append(ind1.path[i])
6:   for i = loc to length(ind2.path) do
7:     new_child.path.append(ind2.path[i])
8:   for i = loc to length(ind1.path) do
9:     new_child.path.append(ind1.path[i])
10:  return new_child
```

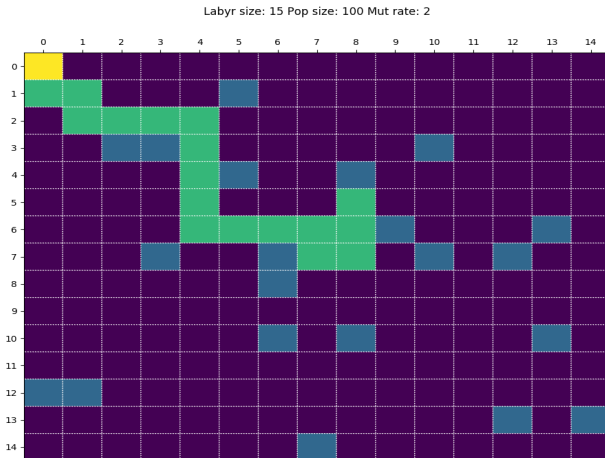
4 Sonuçlar



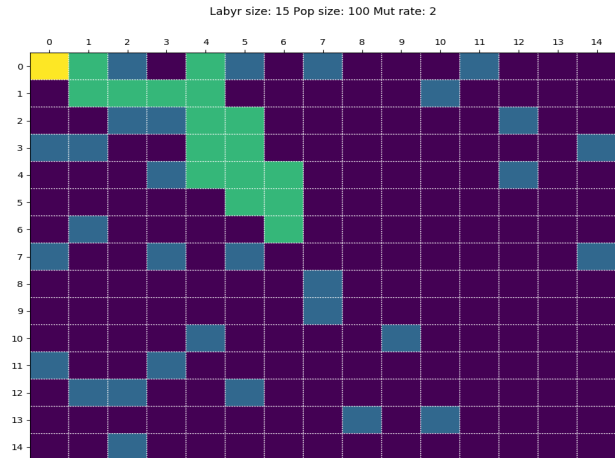
(a)



(b)

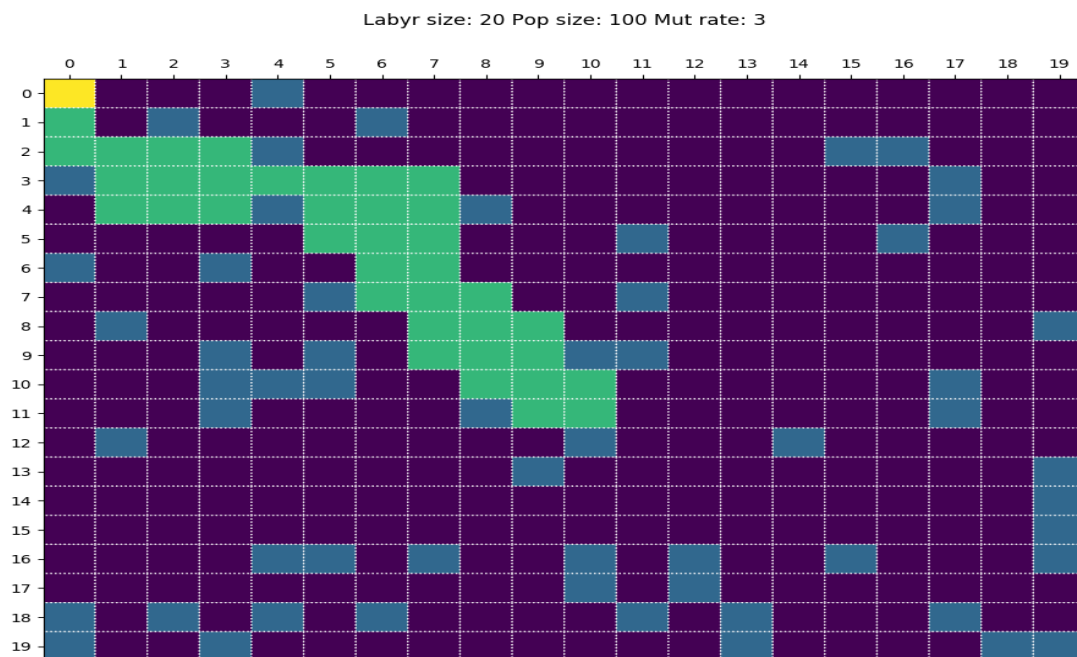


(c)

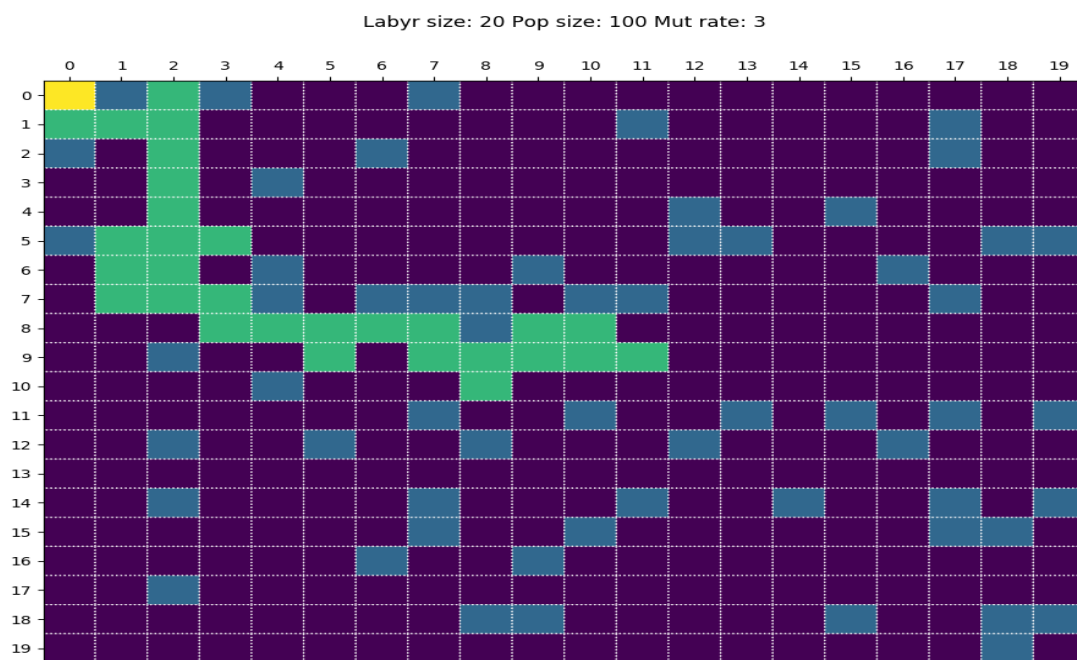


(d)

Figure 1: Yol Bulma Sonuçları

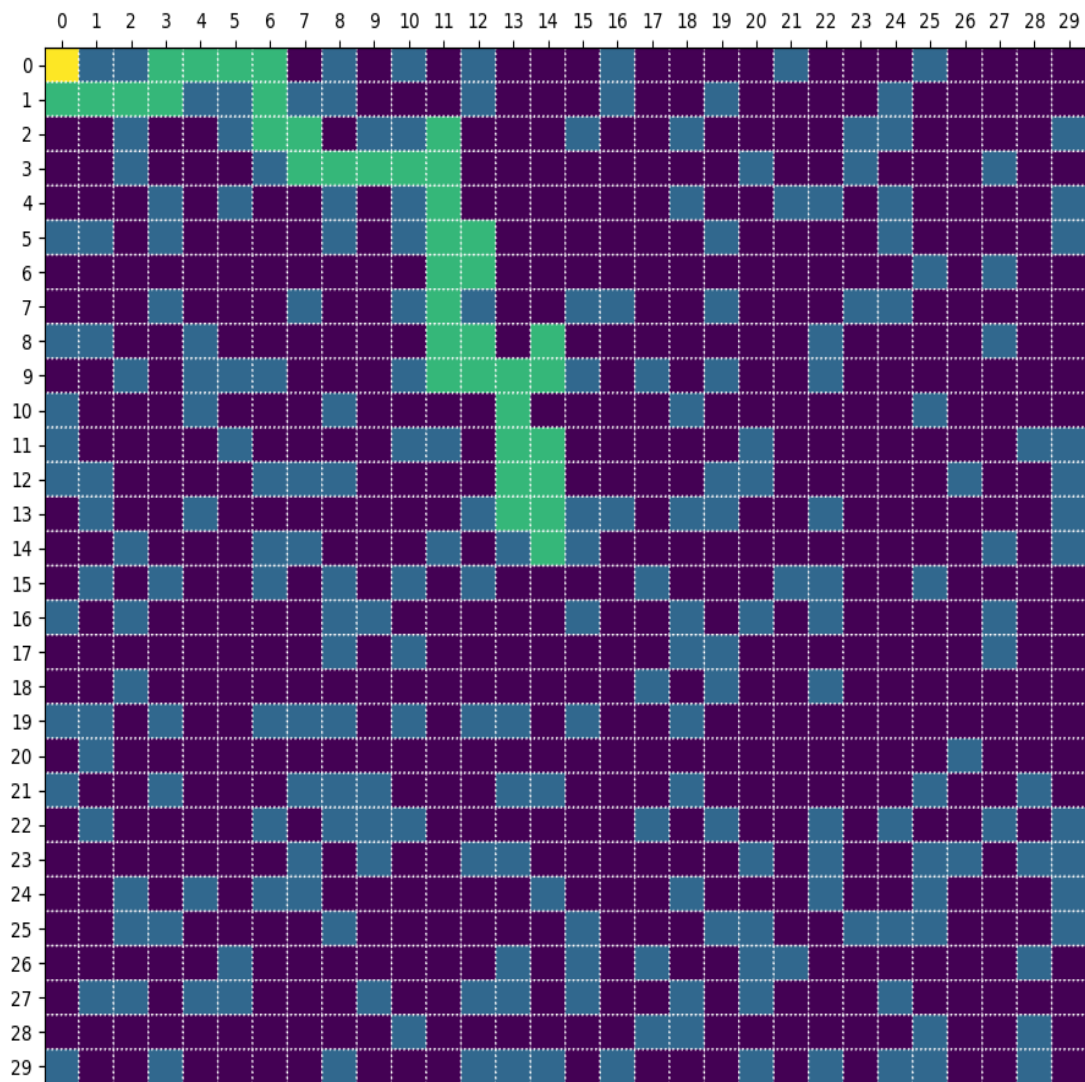


(a)

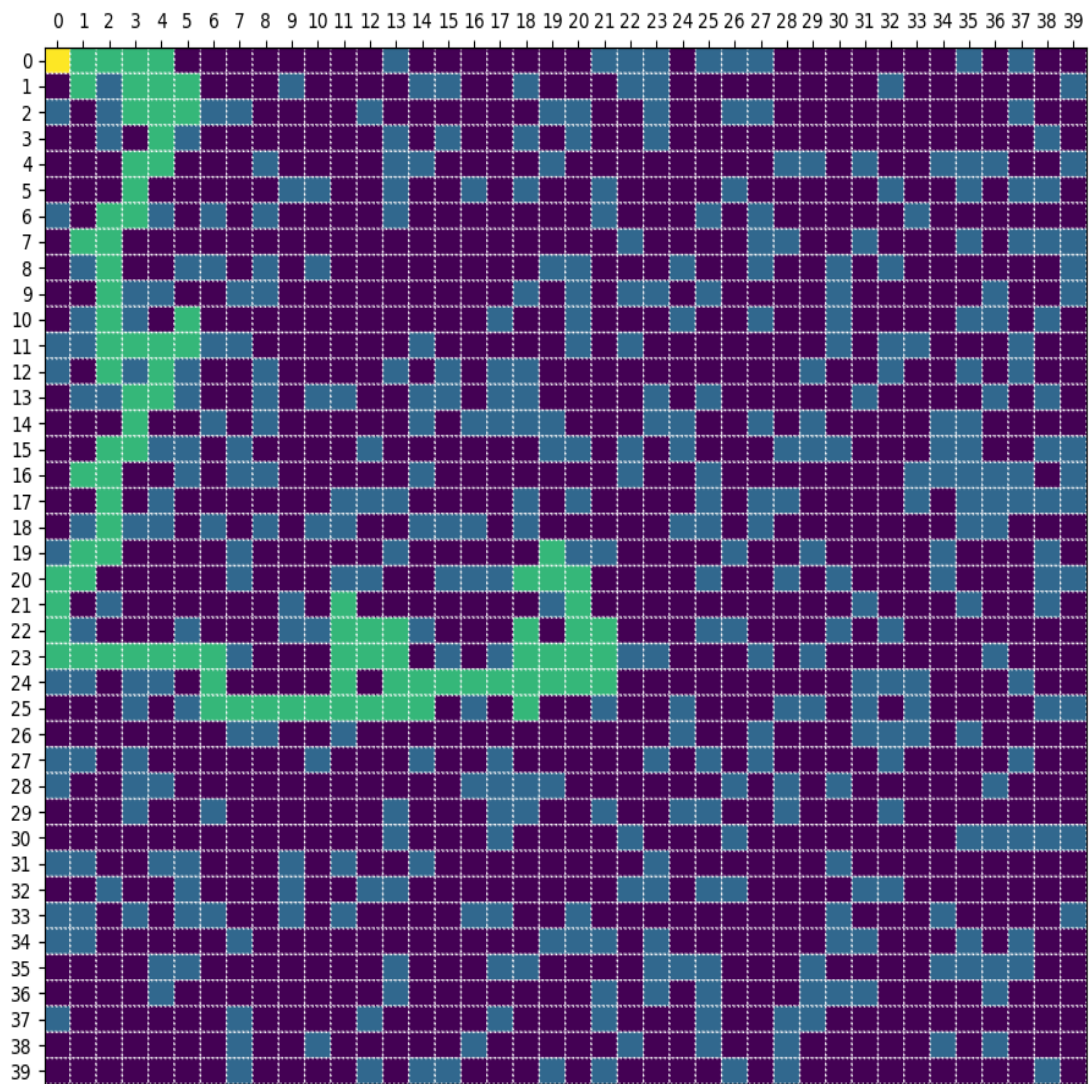


(b)

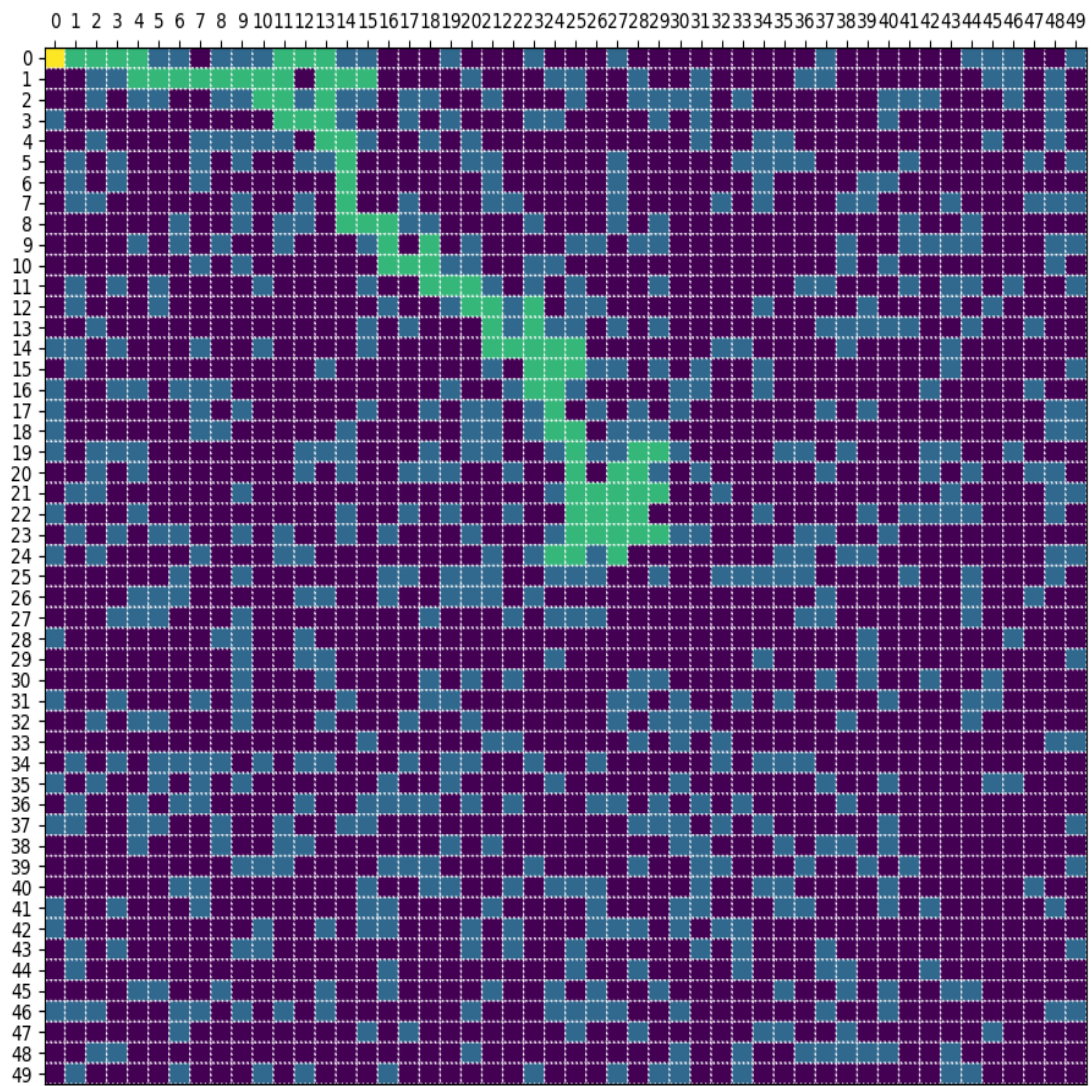
Labyr size: 30 Pop size: 100 Mut rate: 3



Labyr size: 40 Pop size: 250 Mut rate: 3



Labyr size: 50 Pop size: 250 Mut rate: 3



Labyr size: 60 Pop size: 250 Mut rate: 3

