



**KOCAELİ ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ**

**YAZILIM LABORATUVARI - II PROJE - III  
Q-LEARNING İLE YOL PLANLAMASI**

**CEMRE CAN KAYA  
190201137**

**ENGİN YENİCE  
190201133**

**KOCAELİ 2021**

# Q-Learning ile Yol Planlaması

Cemre Can Kaya  
Bilgisayar Mühendisliği  
Kocaeli Üniversitesi  
190201137

Engin Yenice  
Bilgisayar Mühendisliği  
Kocaeli Üniversitesi  
190201133

**Özet—** Pekiştirmeli öğrenme (Reinforcement learning) kullanılarak labirent içerisinde belirli bir hedefe en kısa yoldan ulaşmayı hedefleyen bilgisayar programının geliştirilmesidir.

**Anahtar** *Kelimeler—reinforcement, learning, pekiştirmeli, öğrenme, labirent, ödül, ajan, agent, score, puan, engel, block*

## I. GİRİŞ

Pekiştirmeli öğrenmede ajan (agent) adı verilen öğrenen makinemiz karşılaştığı durumlara bir tepki verir ve bunun karşılığında da sayısal bir ödül sinyali alır. Ajan/öğrenen makine aldığı bu ödül puanını maksimuma çıkartmak için çalışır. Bu şekilde çalışan deneme yanılma yöntemi (brute force), pekiştirmeli öğrenmenin en ayırt edici özelliğidir.

Pekiştirmeli öğrenme bir problem hakkında çalışma ve çözüm yöntemidir.

Pekiştirmeli Öğrenme (Reinforcement learning) algoritması Markov karar süreci denilen bir model kullanmaktadır.

Markov karar süreçlerinin en önemli 3 özelliği;

1. Algılama (sensation)
2. Eylem (action)
3. Hedef (goal)

Pekiştirmeli öğrenme (Reinforcement learning) sürecindeki en önemli zorluklar, keşif (exploration) ve sömürü (exploitation) kavramlarının uygulamaya geçirilmesidir. Ajanın (Agent) daha fazla ödül elde etmesi için geçmişte denediği ve pozitif ödül aldığı eylemleri seçmelidir. Ajan ödül elde etmek için daha önce deneyimlediği eylemleri seçmelidir. Ajan (Agent) ödül elde etmek için daha önce deneyimlediği eylemlerden yararlanır, ancak karşılaştığı bir durumda daha fazla ödül alabileceği eylemler varsa bunları da keşfetmelidir. Böylece Ajan(Agent), çeşitli eylemler denemeli ve en iyi sonuç/ödül alabildiklerini aşamalı olarak desteklemelidir.

Tüm pekiştirmeli öğrenme ajanları açık hedeflere sahip olup çevrenin özelliklerini algılayabilir ve çevrelerini etkileyebilecek eylemleri seçebilirler.

### 1) Pekiştirmeli Öğrenme Öğeleri

Bir Pekiştirmeli Öğrenme sisteminde ajan ve çevre (environment) dışında biri opsiyonel olmak üzere dört unsur bulunur:

- Politika (policy)
- Ödül (reward signal)
- Değer/Durum Değeri (value function)
- Çevre modeli (model)

#### 1.a) Politika (policy)

Ajanın içinde bulunduğu durumda alabileceği aksiyonu belirler. Bir nevi etki-tepki eşleşmesi olarak düşünülebilir. İçinde bulunulan durum bir etki olarak kabul edilirse ajan buna karşılık bir tepki (action) verir. Bu politika basit bir aksiyon olarak tanımlanabileceği gibi bütün durumları karşılayan bir arama tablosu şeklinde de tanımlanabilir. Politika dinamik olarak da nitelenebilir. Bunun temel nedeni, ajanın içinde bulunduğu durumu değerlendirerek alabileceği aksiyonları aramasından (farkına varmasından) kaynaklanmaktadır.

#### 1.b) Ödül (reward signal)

Ajanının gerçekleştirmiş olduğu bir aksiyona karşılık çevreden aldığı puandır. Bir pekiştirmeli öğrenme ajanının amacı, uzun vadede aldığı ödülleri maksimum seviyeye ulaştırmaktır. Ödül alınan aksiyonun ne kadar iyi veya kötü olduğunu belirleyen değerdir (basit bir şekilde mutluluk veya acı ile eşleştirilebilir). Ajan, izlemiş olduğu politikayı bu ödülleri esas alarak zaman içerisinde değiştirir. Örneğin alınan bir aksiyonun sonrasında düşük bir puan elde ediliyorsa, gelecekte ajan aynı duruma geldiğinde farklı bir aksiyon almayı tercih edebilir.

#### 1.c) Değer/Durum Değeri (value function)

Ajanın içinde bulunduğu durumdan ve o durumu takip eden diğer durumlardan bekleyebileceği ödüllerin toplamıdır. Ödüller anlık olarak neyin iyi neyin kötü olduğunu ifade ederken, durum değeri uzun vadede neyin iyi neyin kötü olduğunu ifade eder. Örneğin; bir durum, düşük bir ödül fakat yüksek bir değere sahip olabilir. Bunun nedeni düşük ödül veren durumu takip eden yüksek ödüllü diğer durumlardır. Tam tersi de mümkündür. Yüksek ödül veren durumlar da olabilir. Buradaki durum “ileri görüşlülük” gibi düşünülebilir.

#### 1.d) Çevre modeli (model)

İsteğe bağlı olarak sisteme dahil edilen bir unsurdur. Çevrenin bir simülasyonu olup ajanın bir aksiyonu gerçekleştirmeden önce bu aksiyon sonucunda alabileceği ödülü ve doğuracağı durumu tahmin etmesini sağlamaktadır. Bu sayede bir planlama yapılarak ajanın davranışında değişiklik meydana gelebilecektir.

### 2) Kapsam ve Sınırlamalar

Pekiştirmeli Öğrenme, yoğun bir şekilde durum(state) kavramına dayanmaktadır. Politika ve değer fonksiyonunda girdi olarak kullanılırken; modelde ise hem girdi hem de çıktı olarak kullanılmaktadır.

## II. YÖNTEM

### 1) Pekiştirmeli Öğrenme (Reinforcement learning) Yaklaşımı

Pekiştirmeli öğrenmede ajan (agent) adı verilen öğrenen makinemiz karşılaştığı durumlara bir tepki verir ve bunun karşılığında da sayısal bir ödül sinyali alır. Ajan/öğrenen makine aldığı bu ödül puanını maksimuma çıkartmak için çalışır. Bu şekilde çalışan deneme yanılma yöntemi (brute force), pekiştirmeli öğrenmenin en ayırt edici özelliğidir.

Pekiştirmeli öğrenme bir problem hakkında çalışma ve çözüm yöntemidir.

$$QTable[State, Action] = RTable[State, Action] + \gamma * \max(QTable[ActionState])$$

### 2) Görsel Üzerinden Formül İncelemesi

- **State:** Ajanın o anki bulunduğu konum
- **Action:** Ajanın seçtiği yöndeki hareket
- **ActionState:** Ajanın seçtiği yöndeki hareketi sonucu bulunduğu yeni konum

#### 2.a) Turuncu Alan

Ajanın bulunduğu durumdan seçilen yöndeki hareket sonucu formüle göre elde edilen kazanç değerinin güncelleneceği alan.

#### 2.b) Kırmızı Alan

R tablosundaki durum ve seçilen yöndeki ajanın geçiş yapmaya çalıştığı durumun incelendiği, hedefe mi, duvara mı ya da hareket edilebilir bir yolda mı olduğunun bilgisi edinilen ve bu hareket sonucu elde edilen kazanç değeri.

#### 2.c) Mavi Alan

Ajanın yolu öğrenme oranı. Bu oran sayesinde ajanın bir sonraki hamlesinden elde etmesi gereken kazanç azaltılıp, artırılabilir. Oran azaltıldığında ajan kısa vadede en karlı yolu takip eder. Oran artırıldığında ajan uzun vadeli ödülleri hedefler.

#### 2.d) Siyah Alan

Ajanın seçtiği yöndeki durumdan bir sonraki hamlede gidebileceği tüm durumlardan q tablosundaki değeri en yüksek olan durumun değerini ifade eder.

## III. KULLANILAN PAKETLER

Proje geliştirilirken ihtiyaç duyulan paketler ve kullanım amaçları

1. pygame
2. pygame-menu
3. texttable
4. matplotlib

### 1) Pygame

Multimedya uygulamaları yapmak için kullanılan açık kaynaklı ve ücretsiz bir Python kütüphanesidir. Tüm işletim sistemlerinde çalışmaktadır. (Windows, Linux ve MacOS).

Pygame SDL, Simple DirectMedia Layer kütüphanesinden güç almaktadır. SDL (Simple DirectMedia Layer) C ve Assembly dili kullanılarak hazırlanan, çeşitli optimizasyon aşamalarından geçmiş ve hız konusunda başarılı bir platformdur.

#### 1.a) Nasıl Kurulur

**pip install pygame** komutu kullanılarak kurulum gerçekleştirilir.

### 2) Pygame-Menu

Menüler oluşturulmak için geliştirilen python-pygame kütüphanesidir. Buton, giriş nesnesi, seçiciler gibi bir çok özelliği içerisinde bulundurmaktadır.

#### 2.a) Nasıl Kurulur

**pip install pygame-menu** komutu kullanılarak kurulum gerçekleştirilir.

### 3) Texttable

Metinleri tablo şeklinde görüntülemek için kullanılan python kütüphanesidir.

#### 3.a) Nasıl Kurulur

**pip install texttable** komutu kullanılarak kurulum gerçekleştirilir.

### 4) Matplotlib

Matplotlib, Python programlama dili ve onun sayısal matematik uzantısı NumPy için bir çizim kitaplığıdır. Tkinter, wxPython, Qt veya GTK gibi genel amaçlı GUI araç takımlarını kullanarak çizimleri uygulamalara yerleştirmek için nesne tabanlı bir API sağlar.

#### 4.a) Nasıl Kurulur

**pip install matplotlib** komutu kullanılarak kurulum gerçekleştirilir.

## IV. NASIL KULLANILIR

### 1) Menü



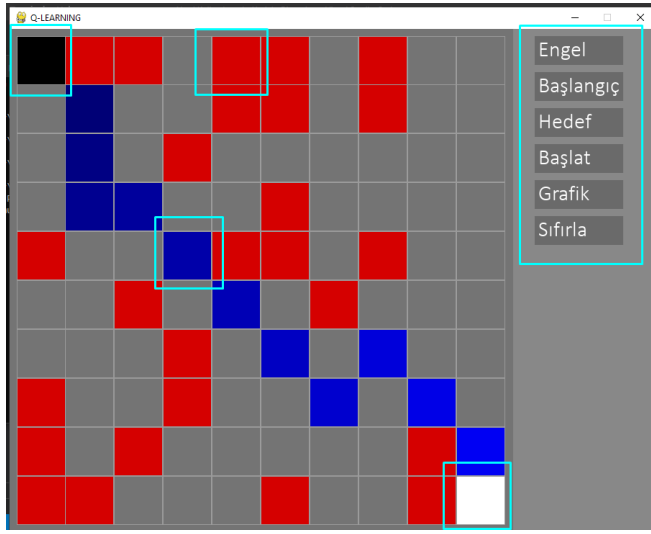
1.a) *Matrix X*  
Matrisinin genişlik değerini yönetebilirsiniz.

1.b) *Matrix Y*  
Matrisin yükseklik değerini yönetebilirsiniz.

1.c) *Başlat*  
Labirent üzerinde yol tespiti yapılacak uygulamayı başlatır.

1.d) *Çıkış*  
Projeyi sonlandırır.

## 2) Oyun Ekranı



### 2.a) Oyun Alanı (Harita)

1. *Gri Kare*  
Yol olarak nitelendirilir.
2. *Siyah Kare*  
Ajan olarak nitelendirilir.
3. *Beyaz Kare*  
Hedef olarak nitelendirilir.
4. *Kırmızı Kare*  
Engel olarak nitelendirilir.
5. *Mavi Tonlama*  
Anlık olarak yolu göstermektedir.

### 2.b) Sağ Menü

1. *Engel*  
Engel seçeneği seçilerek ardından yol üzerine tıklanması durumunda yeni engel oluşturulur. Engel üzerine tıklandığında engel harita üzerinden kaldırılır.
2. *Başlangıç*  
Başlangıç(Ajan)'ın harita üzerinde hangi konumda olacağını belirleyebilirsiniz. Harita üzerinde başlangıç(ajan) olmaması durumunda

haritaya başlangıç(ajan) ekleyebilirsiniz. Varolan bir başlangıç(ajan)'ın üzerine tıklayarak başlangıç(ajan)'ı harita üzerinden kaldırabilirsiniz. **Harita üzerinde sadece bir adet başlangıç(ajan) eklenebilir.**

### 3. Hedef

Hedef'in harita üzerinde hangi konumda olacağını belirleyebilirsiniz. Harita üzerinde Hedef olmaması durumunda haritaya hedef ekleyebilirsiniz. Varolan bir hedefin üzerine tıklayarak hedefi harita üzerinden kaldırabilirsiniz. **Harita üzerinde sadece bir adet hedef eklenebilir.**

### 4. Başlat

Pekiştirmeli öğrenme (Reinforcement learning) algoritmasını başlatır. Başlat butonuna basılması durumunda buton ismi Durdur olarak güncellenir.

### 5. Durdur

Pekiştirmeli öğrenme (Reinforcement learning) algoritmasını durdurur. Durdur butonuna basılması durumunda buton ismi Başlat olarak güncellenir.

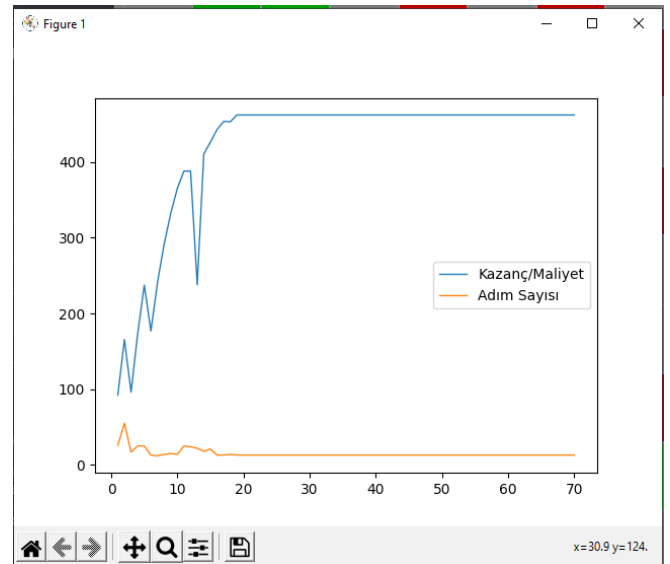
### 6. Grafik

Pekiştirmeli öğrenme (Reinforcement learning) algoritması sonucunda oluşan değerlerin grafiksel olarak gösterildiği pencereyi açar.

### 7. Sıfırla

Haritayı sıfırlar.

## 3) Grafik



### 3.a) Kazanç/Maliyet

Hedefe ulaşan yollar içerisinde kazanç/maliyet değerlerini grafiksel olarak gösterir.

### 3.b) Adım Sayısı

Hedefe ulaşan yollar içerisinde hedefe ulaşmak için sarf edilen adım sayısını gösterir.

## V. SONUÇ

Uygulama çalıştırıldığında da işlemlerin sonucunda bulunan yol ve harita bilgileri result.txt dosyası içerisinde kayıt edilmektedir.

### 1) Bilgilendirme Tablosu

BİLGİLENDİRME TABLOSU	
BILGI	ICERIK
Duvar	[K]
Yol	[G]
Yukseklık	5
Genislik	5
Baslangıc(Ajan)	X:0 Y:0
Hedef(Odul)	X:4 Y:4

Dosya içerisinde bulunan kavramların sembolünü gösteren tablodur.

### 2) En Kısa Yol

EN KISA YOL HAREKET TABLOSU			
ADIM SAYISI	X	Y	
1. adım	0	0	
2. adım	1	1	
3. adım	2	2	
4. adım	3	3	
5. adım	4	4	

Bulunan en kısa yolun baştan sona tüm adımlarının listelendiği tablodur.

### 3) Harita

```
HARITA
[ (0,0,G) (0,1,G) (0,2,G) (0,3,G) (0,4,G) ],
[ (1,0,K) (1,1,G) (1,2,G) (1,3,K) (1,4,G) ],
[ (2,0,G) (2,1,K) (2,2,G) (2,3,K) (2,4,G) ],
[ (3,0,K) (3,1,G) (3,2,G) (3,3,G) (3,4,K) ],
[ (4,0,G) (4,1,K) (4,2,K) (4,3,G) (4,4,G) ],
```

Labirentin haritasının yazılı olarak yansıtıldığı ekrandır.

## VI. KAYNAKÇA

- 1) [python.org](https://python.org)
- 2) [pygame.org](https://pygame.org)
- 3) [pygame-menu.readthedocs.io](https://pygame-menu.readthedocs.io)
- 4) [texttables.readthedocs.io](https://texttables.readthedocs.io)
- 5) [matplotlib.org](https://matplotlib.org)
- 6) [pypi.org/project/pip](https://pypi.org/project/pip)
- 7) [enginyenice.com](https://enginyenice.com)