

YILDIZ TEKNİK ÜNİVERSİTESİ BİLGİSAYAR MÜHENDİSLİĞİ



Öğrenci No:

Ad-Soyad : x Y

Öğrenci E-Postası: @std.yildiz.edu.tr

BLM-1012 —YAPISAL PROGRAMLAMAYA GİRİŞ FİNAL PROJESİ

MİNİMAX ALGORİTMASI

Ders Yürütücüsü

Öğr. Gör. Dr. Ahmet Elbir

Haziran,2021

İÇERİK

- Minimax Algoritması Nedir? Ne işe yarar? Nasıl Çalışır?
- Kullanım Yerleri
- Avantaj-Dezavantajları
- Karmaşıklığı
- Sınırları ve rakipleri
- Çalışmasını açıklarken kullanılan ekran çıktıları
- C dilindeki kodu
- Kaynaklar

VIDEO ADRESİ

<https://youtu.be/>

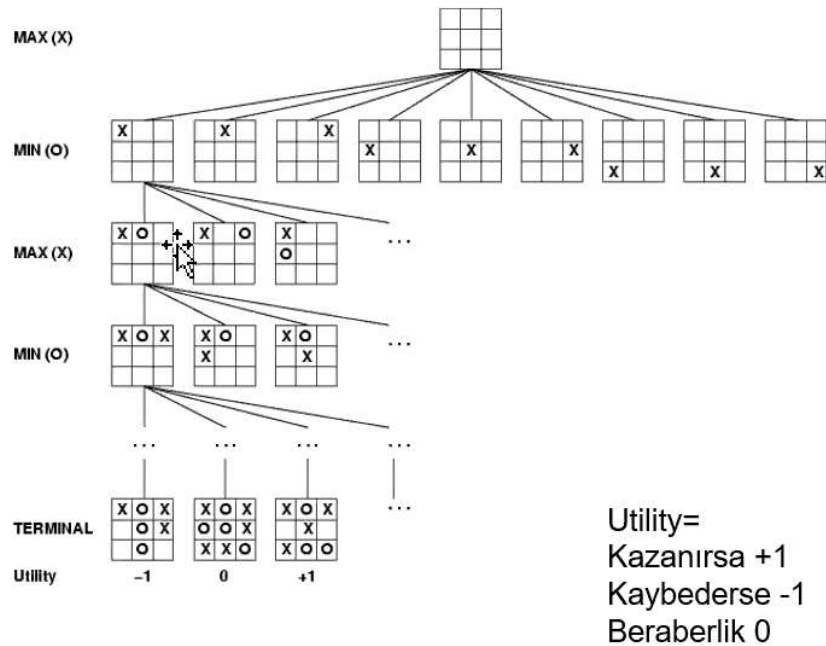
MiniMax Algoritması:

- İhtimaller havuzu içinden en iyi ve en kötü senaryoları değerlendiren bir karar verme algoritmasıdır.
- 2 oyunculu bir oyunda diğer oyuncunun da en iyi şekilde oynadığını varsayarak bir oyuncu için en uygun hareketi seçmek için kullanılan özyinelemeli bir algoritmadır.

Tic Tac Toe Oyunu MiniMax Algoritması ile Nasıl Çalışır?

-Minimax algoritması ile her bir hamlede son duruma bakarak, bundan sonraki hamlelerin n cxsce

Oyun ağacı – Game tree (2-oyuncu)

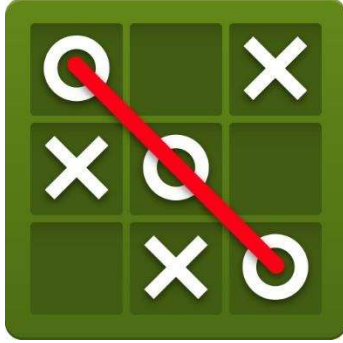


- İhtimaller hesaplandıktan sonra, son durumlardan başlayarak yukarıya doğru hamleler puanlandırılır. (Eğer son hamlede galip gelmişsek +1 puan, mağlup olduysak -1 puan, beraberlik oluştuysa da 0 puan verilebilir.)

Şimdi bu puanların minimize ve maksimize edilmesi gerekir. Burada her bir hamledeki puanların minumumunu veya maksimumunu taşımak. Minimax algoritması rakibin her

Kullanım Yerleri:

- Tic Tac Toe, satranç, dama, go, Isola gibi birçok iki oyunculu oyunlar gibi oyunlarda kullanılır.
- Bu oyunlara mükemmel bilgi oyunları denir çünkü belirli bir oyunun tüm olası hareketlerini görmek mümkündür.



Avantajları:

- problem çözme algoritmasıdır.
- Yeni ve akıllı makineler, sistemler ve bilgisayarların geliştirilmesine yol açan Yapay Zeka'da karar vermeyi mümkün kılar.

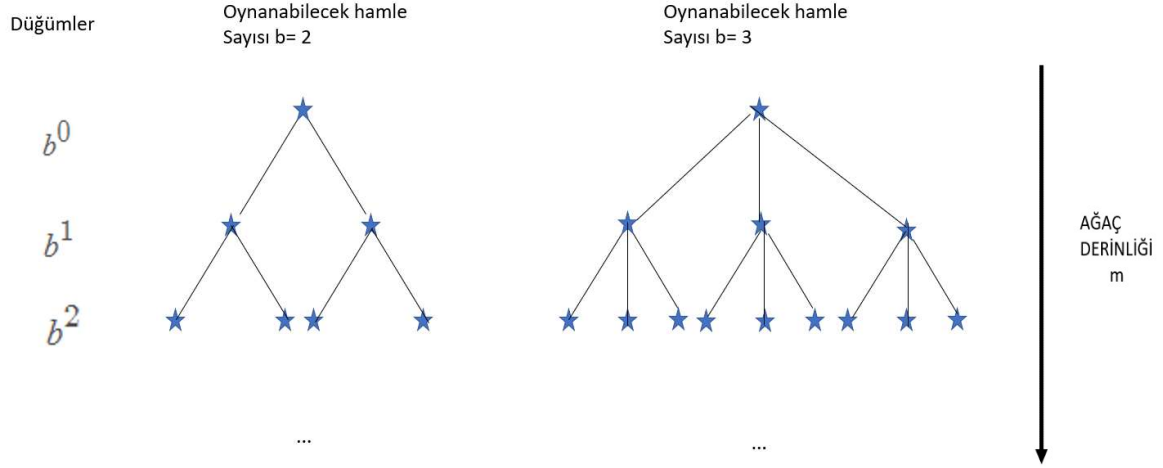
Dezavantajları:

- Hedef durumuna ulaşma sürecini yavaşlatan büyük bir dallanma faktörüne sahiptir.
- Oyun ağacının gereksiz düğümlerinin veya dallarının aranması ve değerlendirilmesi,
- Hem min. hem de max. Oyuncularının karar verecekleri birçok seçeneği vardır.
- Zaman ve mekan

Yer Karmaşıklığı:

- b: Her noktadaki oynanabilecek mümkün hamlelerin sayısı
- m: ağacın maksimum derinliği olmak üzere Minimax algoritmasının yer karmaşıklığı
=>

Zaman Karmaşıklığı:



- Burada b : Her noktadaki oynanabilecek mümkün hamlelerin sayısıdır.
- m : oluşturulan ağacın maksimum derinliğidir.
- Bir düğüm oluşturma maliyetini c gibi zaman sabiti seçersek
 - 0
 - 1
 - 2
 - Bu seviye uzunluğu ağacın maksimum derinliğine kadar olduğundan ağacın en m olur.
 - $c + c*b^1 + \dots + c*b^m$ toplam iş miktarı elde ederiz.
 - Karmaşıklık için en hızlı yükselen elemana bakıldığından ve tüm derinliklerine bakılmasından dolayı zaman karmaşıklığı = $O(b^m)$

Berabere Durumda Zaman Karmaşıklığı:

```
Bilgisayar: 0, Siz: X
Her hareketin numarası
0- 1- 2-
3- 4- 5-
6- 7- 8-
Bilgisayar: 0, Siz: X
Oyna (ilk sıra)(1). ya da (ikinci sıra)(2).?1

  |  |
--+--+
  |  |
--+--+
  |  |

Hareketeniz([0..8]): 0

X |  |
--+--+
  | 0 |
--+--+
  |  |

Hareketeniz([0..8]): 1

X | X | 0
--+--+
  | 0 |
--+--+
  |  |

Hareketeniz([0..8]): 6

X | X | 0
--+--+
0 | 0 |
--+--+
X |  |
```

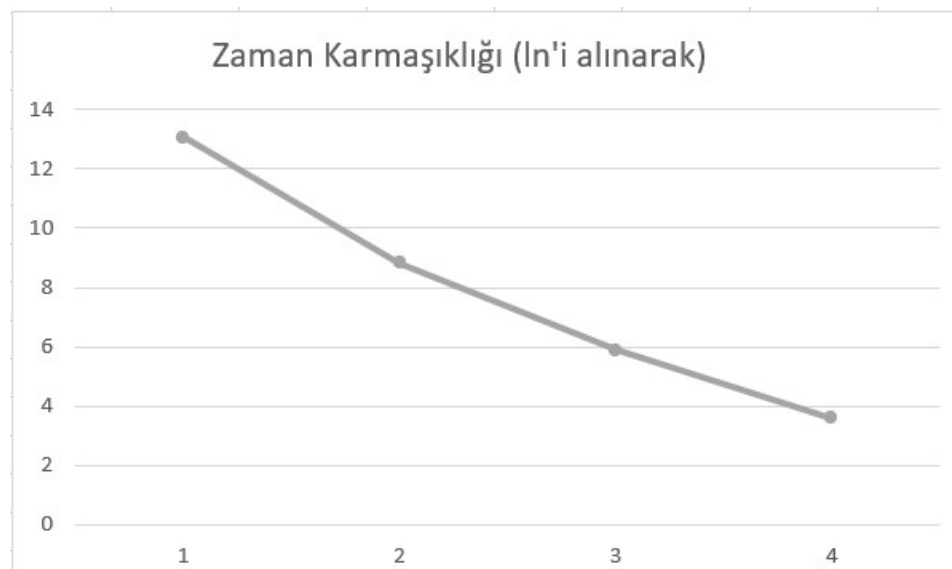
```
Hareketeniz([0..8]): 5

X | X | 0
--+--+
0 | 0 | X
--+--+
X | 0 |

Hareketeniz([0..8]): 8

X | X | 0
--+--+
0 | 0 | X
--+--+
X | 0 | X
BERABERE.
Counter ile Zaman karmasikligi
1.adimda yaklasik zaman karmasikligi= 463752
2.adimda yaklasik zaman karmasikligi= 6853
3.adimda yaklasik zaman karmasikligi= 358
4.adimda yaklasik zaman karmasikligi= 36

Her hamle sonrasi zaman karmasikligi grafigi(ln'i alinarak)
*****
*****
*****
***
```



Oyuncunun Kaybetmesi Durumunda Zaman Karmaşıklığı:

```
Bilgisayar: O, Siz: X
Her hareketin numarasi
0- 1- 2-
3- 4- 5-
6- 7- 8-
Bilgisayar: O, Siz: X
Oyna (ilk sıra)(1). ya da (ikinci sıra)(2).?2

 0 |  |
---+---
  |  |
---+---
  |  |

Hareketeniz([0..8]): 1

 0 | X |
---+---
 0 |  |
---+---
  |  |

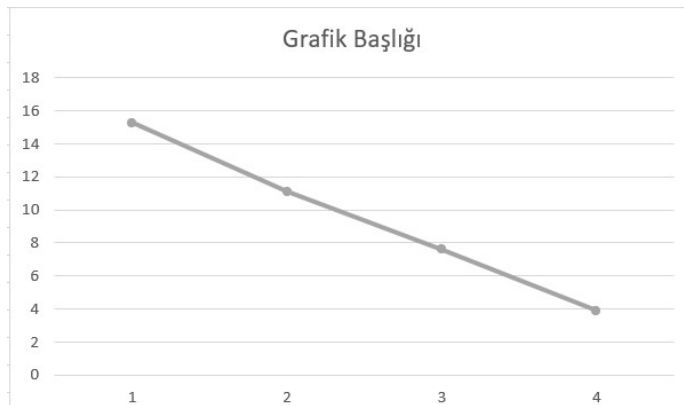
Hareketeniz([0..8]): 6

 0 | X |
---+---
 0 | O |
---+---
 X |  |

Hareketeniz([0..8]): 2

 0 | X | X
---+---
 0 | O | O
---+---
 X |  |
KAYBETTİNİZ
Counter ile Zaman karmasikligi
1.adimda yaklasik zaman karmasikligi= 4276161
2.adimda yaklasik zaman karmasikligi= 66325
3.adimda yaklasik zaman karmasikligi= 1999
4.adimda yaklasik zaman karmasikligi= 49

Her hamle sonrasi zaman karmasikligi grafigi(ln'i alinarak)
*****
*****
*****
***
```



- Bu 2 sonuçta görüldüğü üzere her adımda oynanabilecek mümkün hamlelerin sayısı ve ağacın bundan sonraki derinliği azaldığından zaman karmaşıklığı azaldığı sonucuna varabiliriz.

Satranç için Sınırlar:

- Satranç oyunu için Minimax algoritmasını ele alalım.
- Arama uzayının büyüklüğü (32^{40})
 -
 - Her adımda yapılabilecek farklı hamle sayısı ortalaması = 32
 - $32^{40} = 2^{200} \sim 10^{60}$
- Saniyede 3 milyar ($3 \cdot 10^9$) durum işlersek
 - Bir yıldaki saniye sayısı $\sim 32 \cdot 10^6$
 - Bir yılda işlenebilecek durum sayısı $\sim 10^{17}$
 - Bu durumda tüm durumların değerlendirilmesi $\sim 10^{43}$ yıl sürer.
 - Ama evrenin yaşı $\sim 10^{10}$ yıl..
- Minimax algoritmasının satranç gibi oyun ağacının çok çok büyük olduğu oyunlarda işe yaramadığını gördük. Peki ne yapabiliriz ?

Minimax Algoritması Rakibi Alpha- Beta ($\alpha - \beta$) Budaması:

- Oyun arama ağacının her düğümüne bakmadan bir Minimax kararı vermeye çalışır. Görünüşe göre birçok dal göz ardı edilebilir(budanabilir). Arama sırasında iki değer oluşturulur;
 - **Alfa**: Maksimum düğümlerle ilişkilendirilir.
 - **Beta**: Minimum düğümlerle ilişkilendirilir.
- Bu şekilde tüm oyun ağacının incelenmeden; nihai kararı etkilemeyen dalları budayarak:
 - En kötü durum performansında $\rightarrow \underline{d}$
 - En iyi durum performansında $\rightarrow \overline{b}$
- Sonuç olarak Minimax algoritmasına göre hesaplama ve arama miktarı azalır.

C Dilinde Kodu

KODLAR SILINMISTIR :)

Kaynaklar:

- <https://cs.stanford.edu/people/eroberts/courses/soco/projects/2003-04/intelligent-search/minimax.html>
- https://tr.esc.wiki/wiki/Alpha-beta_pruning
- <http://www.cs.trincoll.edu/~ram/cpsc352/notes/minimax.html#:~:text=Alpha%2Dbeta%20pruning%20is%20a,in%20a%20depth%2Dfirst%20fashion.>
- <https://www.professional-ai.com/minimax-algorithm.html>
- <https://www.geeksforgeeks.org/minimax-algorithm-in-game-theory-set-1-introduction/>
- https://www3.ntu.edu.sg/home/ehchua/programming/java/javagame_tictactoe_ai.html