



Satranç Köşesi

Eşref Eşkinat* / eskinat@boun.edu.tr

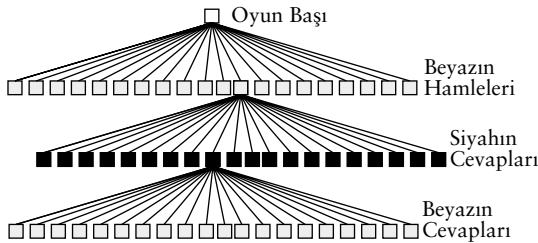


Bilgisayarlar Nasıl Satranç Oynar?

Garry Kasparov'un 1997'de Deep Blue isimli bilgisayarla 6 oyunluk bir maç yapıp yenilmesi uzun süre tartışılmıştı. Bu maçtan sonra önemli oyuncular kuvvetli bilgisayar programlarıyla maçlar yaptılar. Bunların sonuncusu 2003 sonbaharında yine Kasparov'un X3D ile yaptığı ve berabere biten maçı.

Bilgisayar satrancının yaklaşık 50 yıllık bir geçmişi var. İlk zamanlarda donanım yetersizliği yüzünden zayıf oynayan bu programlar bilgisayarların hızlı gelişmesiyle 1990'dan sonra usta, 1995'den sonra ise büyükustalarla başedebilir düzeye geldiler. Artık bir ev bilgisayarında çalışan iyi bir program bile büyükusta düzeyinde oynayabiliyor.

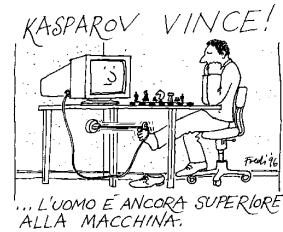
Bilgisayarlar satranç oynarken insanlara benzer bir biçimde düşünmüyorlar. İnsanlar birkaç hamle adayı seçip bunlar üzerinde yoğunlaşırken, bilgisayarlar olası bütün hamleleri belli bir derinliğe kadar değerlendiriyorlar. İnsanlar gibi tercihli arama yapan bazı programlar 1980'lerde (Hans Berliner'in Hitech programı gibi) denendi ama başarılı olamadı. O yüzden bize garip gözükse de bilgisayar programları olası bütün hamlelerden aşağıdaki resimdeki gibi bir hamle arama ağacı oluşturuyorlar.



Bu hamle ağacı çok büyük bir hızla büyür. Beyazın her pozisyonda yaklaşık 20 hamlesi ve siya-

hın da 20 değişik cevabı olduğunu varsayarsak, 5. dereceden bir ağaçta 3.200.000, 10. dereceden bir ağaçta ise 10 trilyon pozisyonun değerlendirilmesi gerekir. İlerde göreceğimiz gibi, bazı tekniklerle bu sayı %90 oranında azaltılabilir ama yine de iyi oynayan bir bilgisayarın her hamlede milyonlarca pozisyonu değerlendirmesi gerekir. Bu durum, uzun süre bilgisayarların iyi oynaması önündeki en önemli engeldi. Bilgisayar donanımı geliştikçe bu bir sorun olmaktan çıktı.

Bilgisayarların her pozisyona bir değer verebilmek için bir değerlendirme fonksiyonu gereklidir. En kaba şekliyle bu fonksiyon, her taş sayısal bir değer verip (örneğin Vezir = 9, Kale = 5, Fil = 3,25, At = 3, Piyon = 1, Şah = 1000) oyundaki taşların ağırlıklı toplamının iki taraf arasındaki farkı alınarak bulunabilir. Haliyle, bu



Tek fikir, üç karikatür...

hem fazla basit hem de fazla materyalist bir fonksiyon olacaktır ama iyi bir hamle arama altprogramıyla, hızlı bir bilgisayarda bayağı iyi oynayan bir program elde etmek için yeterli olabilir. Daha iyi fonksiyonlar için duble piyonlara ceza, açık hatlara ve taş mobilitesine ödül vermek gibi ayrıntıların

* Boğaziçi Üniversitesi Makine Mühendisliği Bölümü öğretim üyesi.



fonksiyonlara eklenmesinin faydalı olduğu görülmüştür.

Oyun sırasında beyaz, değerlendirme fonksiyonunu mümkün olan en yüksek değere ulaştırmaya (yani maksimize etmeye), siyah ise mümkün olan en düşük değere indirmeye (yani minimize etmeye) çalışacaktır.

Bilgisayarın beyazı oynadığını varsayarsak, beyaz, siyahın değerlendirme fonksiyonunu minimize eden hamlelerine rağmen bu fonksiyonu maksimize edecek hamleleri bulmaya çalışacaktır. Bu algoritmaya minimaks algoritması adı verilmektedir. Belli bir hamle ağacı derinliğine kadar uygulanan bu algoritma “en iyi” hamlenin bulunması için bilgisayarın kullandığı başlıca araçtır. Görüldüğü gibi, bilgisayarlar ne kadar iyi oynarlarsa oynasınlar yaptıkları işlemler pek de zekice değil.

Hamle arama ağacında pozisyon sayısının yaklaşık %90 oranında azaltılabileceğinden bahsetmiştik. Bu, **alfa-beta budaması** denilen bir teknikle yapılır. Bu tekniği anlatmak için bir örneğe başvuralım: Diyelim ki rakibinizle bir iddiaya girip kazandınız. Kurallar ise şöyle: Ortada herbirinin içinde 20 tane değişik değerli nesne olan 20 tane torba var. Siz torbayı seçeceksiniz, rakibiniz de size o seçtiğiniz torbadan bir nesneyi seçip verecek. Sizi sevmediği için de en değersiz nesneyi seçeceğine eminsiniz. Rakibiniz size torbadaki en kötü nesneyi vereceği için amacımız en kötüler arasındaki en iyiyi seçmek. Benzer bir biçimde satrançta da rakibinizin size en az kötü cevabı verebileceği hamleyi seçmeye çalışırsınız. İlk torbaya baktınız, bu torbada en az değerli olan nesne bir tencere olsun. Şimdi tencere garanti olarak ikinci torbaya baktınız, ilk olarak elinize bir cezve geldi. Cezve tencereye göre daha az değerli olduğu için, bu ikinci torbadaki diğer nesnelere bakmanıza gerek yok, çünkü bu torbayı seçerseniz rakip size en fazla bu cezveyi verecek, oysa birinci torbada hiç değilse tencere var. Böylece ikinci torbadaki 19 nesneyi değerlendirme ağacımıza katmaktan kurtulduk.

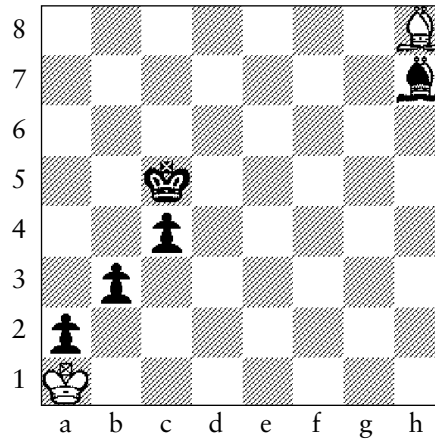
Bu basit gözükten fikir, bütün oyun ağaçlarını budayıp incelemek için çok faydalıdır. Bu sayede hamle ağacını daha derinleştirmek mümkün olabilmektedir. Ayrıca, satrançta aynı pozisyona farklı hamleler yaparak ulaşma olasılığı da oldukça fazladır. Örneğin 1. Ac3 e5 2. e4 Af6 ve 1. e4 e5 2.

Ac3 Af6 hamleleri aynı pozisyona ulaşırlar. Daha önceden değerlendirilen pozisyonların bir çetelesi tutularak değerlendirilmesi gereken pozisyon sayısı bir miktar azaltılabilir. Bu tercih, zamandan tasarruf sağlar ama bilgisayar hafızasında yer tutar.

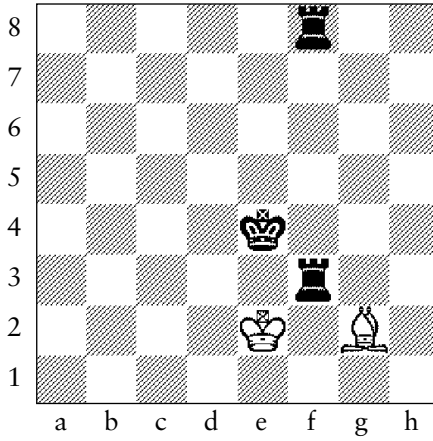
Satranç oynayan programlarda bulunan diğer birimler de açılış ve oyun sonu veritabanlarıdır. Standart açılışlar bir veritabanına yüklenerek bu aşamanın hızlı ve hatasız geçilmesi sağlanır. Satranç oyuncuları bazen bilgisayarı veritabanından ayırmak için garip gözükten hamleler yaparlar. Veritabanından ayrılan bilgisayar programları açılışta çok anlaşılmasız hamleler yapabiliyorlar. Bunun geçenlerde gördüğüm bir örneği şöyle : 1. e4 Ac6 2. d4 d5 3. e5 Kb8?!

Bilgisayarların satranç teorisine önemli bir katkıları 5 taşlı oyun sonlarının (mesela Ş+K+P – Ş+K) tamamen çözülmesi olmuştur. Bu tür oyun sonlarının artık kusursuz oynanması mümkündür. Bundan faydalanan bazı bilgisayar programları oyun sonu veritabanları kullanırlar. Böyle programlarla oynarken 5 taşlı bir oyun sonuna geldiğinizde program birdenbire “23 hamlede mat var” diye bir bildirimde bulunursa şaşırmayın, bu pozisyon veritabanında var demektir.

Bilgisayarlar satrançta insandan farklı biçimde düşünseler de kendilerine özgü bir stilleri var. İnsanların aksine, hiç bariz hata yapmıyorlar, ancak buna karşılık hâlâ stratejik düşünme becerisine sahip değiller. İnsan oyuncular için çok açık olan bazı pozisyonları bilgisayarlar değerlendirmekte zorlanıyorlar. Örnek olarak aşağıdaki pozisyonları verebiliriz.



Bu pozisyon beraberektir. Siyah Şb4 sonrasında piyonunu c3'e sürerse Fxc3 Şxc3 ile pat olur. Bilgisayarlar hamle ağacını ne kadar genişletirlerse genişletsinler bunu göremiyorlar.



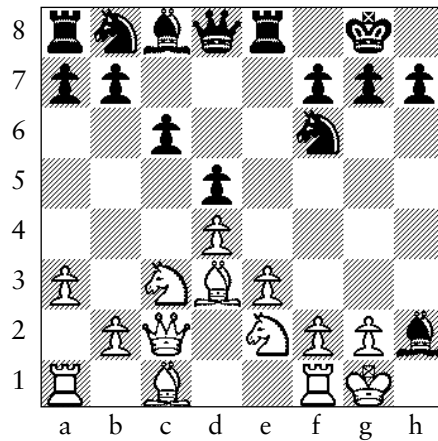
Bu pozisyon da beraberdir. Beyaz sürekli Fg2-h1-g2 oynar, siyah kale de f hattında kalmak zorundadır. Oysa 20 hamlelik bir ağaçtan sonra bile bilgisayarlar bu pozisyonu siyah kazanır olarak değerlendiriyorlar.

Biraz deneyimli bir oyuncu taraflardan birinin bilgisayar olduğu bir oyuna bakınca hangi tarafın bilgisayar olduğunu kolayca anlayabilir. Bu anlamda bilgisayarlar insanlardan daha iyi oynasalar da hâlâ Turing testini geçemediler¹. Bunun aksi örnekleri de olmuyor değil. Aşağıda verdiğimiz oyunda Deep Junior programı tamamen hesaplamadan ve insanların yaptığına çok benzer bir fil fedası ile Kasparov'u şaşırtmış ve beraberliğe zorlamıştı :

Kasparov, G. - Deep Junior

FIDE İnsan-Makine Maçı, New York City, 05. 02. 2003, Nimzo-Hint Açılışı

1. d4 Af6 2. c4 e6 3. Ac3 Fb4 4. e3 0-0 5. Fd3 d5 6. cxd5 exd5 7. Age2 Ke8 8. 0-0 Fd6 9. a3 c6 10. Vc2 Fxh2+!



¹ Bilgisayar programının aldığı kararlar insanların aldıkları kararlardan ayırdedilebiliyorsa, o zaman bilgisayar programı "Turing testini" geçememiş demektir.

Bu çok "insanvari" bir hamledir. Ortada bariz bir kazanç olmadığı halde Deep Junior atak için fil feda ediyor. Kasparov bu hamleyi görünce çok şaşırılmış ve bilgisayarlar sonunu görmeden feda yapmadıkları için kaybettiğini düşünmüş. Ortada açık bir kazanç olmadığını görünce rahatlamış. Bu hamle, bilgisayarın değerlendirme fonksiyonuna hücum pozisyonlarına ekstra puan verilerek sağlanmış.

11. Şxh2 Ag4+ 12. Şg3 Vg5 13. f4 Vh5 14. Fd2 Vh2+ 15. Şf3 Vh4 16. Fxh7+

Beyaz 30 dakikalık bir düşünmeden sonra beraberliğe giden bir yola yöneliyor. Kasparov, 16. g3 devam yolunun bilgisayara karşı çok riskli olduğuna karar veriyor.

16. ... Şh8 17. Ag3 Ah2+ 18. Şf2 Ag4+ 19. Şf3 Ah2+ 1/2-1/2

Bu yazıyı bilgisayarların insanla savaşı henüz kazanmadıklarını gösteren bir örnekle bitiriyoruz. Stratejik ve uzun erimli hücumlar karşısında en güçlü bilgisayar programları bile tamamen çaresiz kalabiliyor. Bu oyunda bilgisayarı boğarak yenen Kasparov insanlığın onurunu kurtarıyor.

Kasparov-X3D Fritz, Oyun 3, New York City, 16. 11. 2003, Slav Savunması

1. Af3 Af6 2. c4 e6 3. Ac3 d5 4. d4 c6 5. e3 6. c5 Bilgisayarların kapalı pozisyonları iyi oynayamadığını bilen Kasparov oyunu kapatıyor.

Abd7 7. b4 a5 8. b5 e5 9. Va4 Vc7 10. Fa3 e4 11. Ad2 Fe7 12. b6! Vd8 13. h3 O-O 14. Ab3 Fd6?! Sadece bilgisayarlar böyle hamleler yapar.

15. Kb1 Fe7 16. Axa5 Ab8 17. Fb4 Vd7

Siyahın Vezir kanadı atağına karşı yapabileceği tek şey şah kanadında f5-f4 sürmek. Ancak X3D bunu "göremiyor". Bu tür pozisyonlarda bilgisayarlar oluşturamıyorlar.

18. Kb2

Bu zekice hamle f2 karesini korumuş gibi yapıp olası bir f5-f4 sürüşünü önüyor. X3D iyi korunduğunu sandığı f2 karesine saldırıyor.

18. ... Ve6 19. Vd1 Afd7 20. a3 Vh6 21. Ab3 Fh4 22. Vd2 Af6 23. Şd1! Fe6 24. Ş c1 Kd8 25. Kc2 Abd7 26. Sb2 Af8 27. a4 Ag6 28. a5 Ae7 29. a6! bxa6 30. Aa5 Kdb8 31. g3 Fg5 32. Fg2 Vg6 33. Şa1 Şh8 34. Aa2 Fd7 35. Fc3 Ae8 36. Ab4 Fc8 37. Ka2 Şg8 38. Kb1 Fh6 39. Ff1 Ve6 40. Vd1 Af6 41. Va4 Fb7 42. Axb7 Kxb7 43. Axa6 Vd7 44. Vc2 Şh8 45. Kb3 1-0 ♠