

# CENG 415 Evrimsel Hesaplama

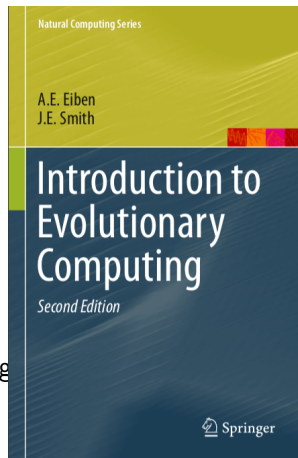
## Bölüm 1: Çözülecek Problemler

Şevket Umut Çakır

Pamukkale Üniversitesi

Hafta 1

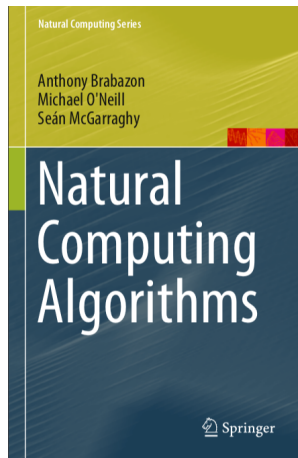
- Eiben, A. E., & Smith, J. E. (2015). Introduction to evolutionary computing. springer.
- Kitap bağlantısı(Üniversite içinden ücretsiz indirilebilir.)
- <http://www.evolutionarycomputation.org> (İngilizce sunumların bulunduğu sayfa)



Şekil: Ders kitabı 1



- Brabazon, A., O'Neill, M., & McGarraghy, S. (2015). Natural computing algorithms (Vol. 554). Berlin: Springer.
- Kitap bağlantısı(Üniversite içinden ücretsiz indirilebilir.)



Şekil: Ders kitabı 2



# Ders Saatleri ve Değerlendirme

- Ders Saatleri
  - ▶ Salı 12:35-15:10
- Değerlendirme yöntemleri
  - ▶ Arasınan %40
  - ▶ Dönem sonu sınavı %60



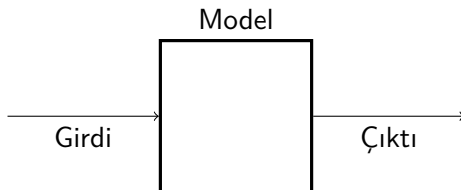
# Çözülecek Problemler

Problemler farklı şekillerde sınıflandırılabilirler.

- Kara kutu modeli
- Arama problemleri
- Eniyileme(optimization) ve kısıt sağlama(constraint satisfaction)
- NP problemler



# Kara Kutu Modeli



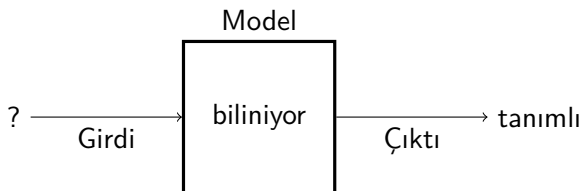
- Kara kutu modeli 3 bileşenden oluşur.
- Bileşenlerden bir tanesi bilinmediğinde: yeni bir problem tipi



# Kara Kutu Modeli

Eniyileme(Optimization)

Model ve arzu edilen çıktı biliniyor, amaç girdileri bulmak



- Örnekler

- ▶ Üniversiteler için ders programı, çağrı merkezleri ve hastaneler için zaman çizelgesi
- ▶ Tasarım özellikleri
- ▶ Gezgin satıcı problemi(TSP)
- ▶ Sekiz vezir problemi vb.



# Kara Kutu Modeli

## Eniyileme Örneđi 1: Üniversite Ders Programı

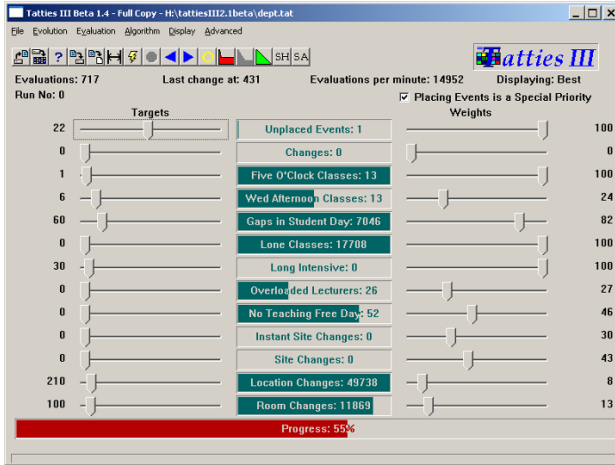
- Çok büyük bir arama uzayına sahiptir
- Ders programları **iyi** olmalıdır
- **İyi** birbiriyle çakışan bir takım kriterler ile tanımlanır
- Ders programı uygulanabilir olmalıdır
- Arama uzayının büyük çoğunluğu uygulanamaz çözümlerden oluşur





# Kara Kutu Modeli

## Eniyileme Örneği 1: Üniversite Ders Programı



Şekil: Tatties III: Evrimsel algoritmalar kullanarak ders programı hazırlayan program



# Kara Kutu Modeli

## Eniyileme Örneği 2: Uydu Yapısı

- NASA'ya titreşim yalıtımını artırmak için eniyilenmiş uydu tasarımları yapılmaktadır
- Geliştirilen: Tasarım
- Uygunluk(Fitness): Titreşim direnci
- Evrimsel yaratıcılık



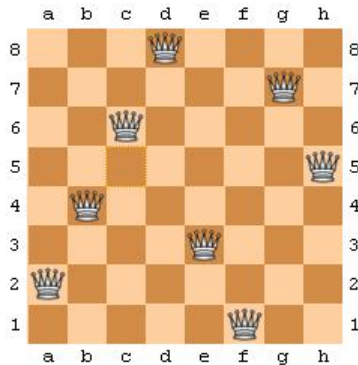
**Şekil:** Evrim stratejisi ile oluşturulmuş uydu anteni



# Kara Kutu Modeli

## Eniyileme Örneği 3: 8 Vezir Problemi

- 8x8'lik bir satranç tahtası ve 8 adet vezir verilir
- 8 adet veziri satranç tahtasına çakışma olmadan yerleştirmek istenir
- Vezirler yatay, dikey ve çapraz yönlerde sonsuz kare ilerleyebilir. Çakışma olmaması için vezirlerin ilerleme yönlerinde başka vezir olmamalıdır
- n-Vezir problemine de dönüştürülebilir ( $n > 8$ )



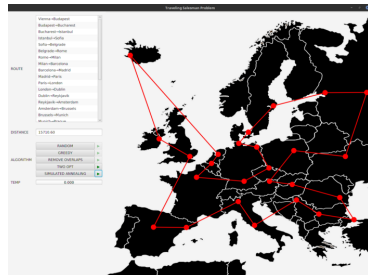
Şekil: 8 vezirin satranç tahtasına örnek yerleştirilmiş hali



# Kara Kutu Modeli

## Eniyileme Örneği 4: Gezgin Satıcı Problemi

- Bir seyyar satıcı var
- Bu satıcı mallarını  $n$  şehirde satmak istiyor
- Satıcı şehirleri mümkün olan en kısa şekilde ve her şehre sadece 1 defa uğrayarak satmak istiyor



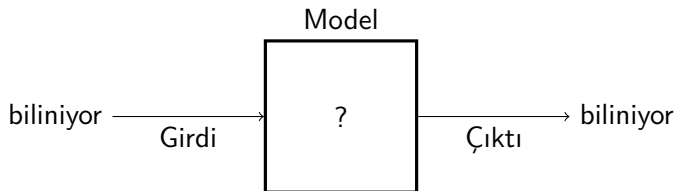
Şekil: Avrupa başkentleri için örnek bir problem



# Kara Kutu Modeli

## Modelleme

Elimizde girdiler ve çıktılar mevcuttur, bilinen girdilere karşılık gelen çıktıları üreten modeli ararız.



- Genellikle dinamik ortamlar geliştirilirken “... olursa ne olur” sorusuna cevap olması için kullanılır
  - ▶ Evrimsel ekonomi, yapay yaşam
  - ▶ Hava tahmin sistemleri
  - ▶ Yeni vergilerin etki analizi için



# Kara Kutu Modeli

Modelleme Örneği: Kredi Başvuru Güvenirliği

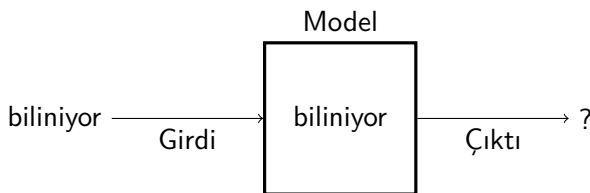
- Bankalar yeni kredi başvurularında güvenirliliği öngörmek için modeller geliştirmektedir
- Geliştirilen: Kestirim/öngörü modeli
- Uygunluk: Geçmiş veri üzerinde modelin doğruluğu



# Kara Kutu Modeli

## Benzetim(Simulation)

Elimizde girdiler ve çıktılar mevcuttur, bilinen girdilere karşılık gelen çıktıları üreten modeli ararız.



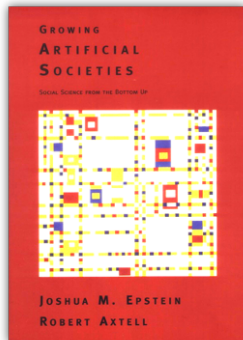
- Örnekler
  - ▶ Evrimsel makine öğrenmesi
  - ▶ Borsa fiyatlarının tahmini
  - ▶ Akıllı evler için ses kontrol sistemleri
- Not: Modelleme problemleri eniyileme problemlerine dönüştürülebilir.



# Kara Kutu Modeli

Benzetim Örneği: Yapay Toplamlar Geliřtirmek

- Modelleri ayarlamak için ticaretin ve ekonomik rekabetin benzetilmesi
- Modeller strateji ve politikaları eniyilemek için kullanılır





# Arama Problemleri

- Benzetim(simulation), eniyileme(optimization) ve modellemeden farklıdır
- Eniyileme ve modelleme problemleri çok büyük olasılık uzaylarında arama yaparlar
- **Arama uzayı:** İstenilen çözümün de bulunduğu ilgili bütün nesnelerin koleksiyonudur
- **Soru:** n adet şehrin farklı sıralarda gezilmesi durumunda arama uzayının büyüklüğü nedir?
- Bu problemleri sınıflandırmanın faydası: arama problemleri(arama uzaylarını tanımlar) ile problem çözücüler(arama uzayında nasıl ilerlenmesi/hareket edilmesi gerektiğini belirler) arasındaki farkın belirlenmesidir



# Eniyilemeye(Optimization) karşı Kısıt Sağlama(Constraint Satisfaction)

- **Amaç fonksiyonu:** Bir çözümün kalitesini sayısal bir değer ile ilişkilendirmenin bir yoludur
  - ▶ Çakışmayan vezir sayısı(ençoklama/maximize)
  - ▶ Şehirlerin kümesini dolaşırken katedilen yol(enazlama/minimize)
- **Kısıt:** Bir gereksinimin gerçekleşip gerçekleşmediğinin ikili değerlendirmesi(evets/hayır)
- Herhangi iki vezirin birbiri ile çakışmadığı bir konfigürasyon bul
- X şehrinden sonra Y şehrinin dolaşıldığı en küçük mesafeye sahip bir tur bul



# Eniyilemeye(Optimization) karşı Kısıt Sağlama(Constraint Satisfaction)

	Amaç fonksiyonu	
Kısıtlar	Evet	Hayır
Evet	Kısıtlı eniyileme problemi(constrained optimisation problem)	Kısıt sağlama problemi(constraint satisfaction problem)
Hayır	Bağımsız eniyileme problemi(free optimisation problem)	Problem yok

- Verilen örnekler hangisine uyar?
- **Not:** Kısıt problemleri eniyileme problemlerine dönüştürülebilir
- **Soru:** 8-vezir problemini KEP/KSP ve BEP şeklinde nasıl temsil edebiliriz?



# NP Problemler

- Sadece problemleri sınıflandırdık, problem çözücüleri incelemedik
- Bu sınıflandırma şeması problem çözücünün özelliklerine ihtiyaç duyar
- Bu şemanın faydası: Problemin ne kadar zor olduğunu söyleyebiliriz



# NP Problemler

## Anahtar kavramlar

- **Problem büyüklüğü(Problem size):** Eldeki problemin boyutunu ve problem değişkenleri için mümkün değer sayısını temsil eder
- **Çalışma zamanı(Running time):** Algoritmanın sonlanması için gerekli işlem sayısı
  - ▶ Problem büyüklüğünün bir fonksiyonu olarak en kötü durum(worst case)
  - ▶ Polinomsal(polynomial), süper-polinomsal, üssel(exponential)
- **Problem indirgeme(Problem reduction):** Mevcut problemi izdüşüm yaparak başka bir probleme dönüştürme



# NP Problemler

## Sınıf

Problemlerin zorlukları aşağıdaki şekilde sınıflanabilir

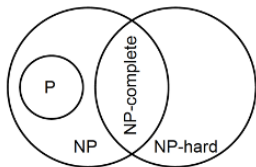
- **P Sınıfı:** Algoritma problemi polinomsal zamanda çözebilir( $n$  büyüklüğündeki bir problem için en kötü durumda çalışma zamanı  $F(n)$ 'den azdır,  $F$  polinomsal bir fonksiyon)
- **NP Sınıfı:** Problem çözülebilir ve herhangi bir çözüm başka bir algoritma tarafından polinomsal zamanda doğrulanabilir( $P$ ,  $NP$ 'nin alt kümesidir)
- **NP-Tam Sınıfı:** Problem  $NP$  sınıfına aittir ve  $NP$  sınıfındaki başka bir problem bu probleme polinomsal zamanda çalışan bir algoritma tarafından indirgenebilir
- **NP-Zor Sınıfı:** Problem en az  $NP$ -tam sınıfındaki bir problem kadar zordur fakat çözüm polinomsal zamanda doğrulanmak zorunda değildir



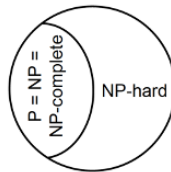
# NP Problemler

## Sınıflar Arasındaki Farklar

- P, NP-zor'dan farklıdır
- P'nin NP'den farklı olup olmadığı bilinmemektedir

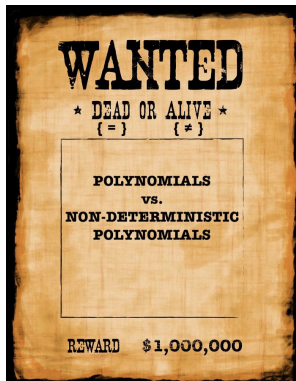


Şekil:  $P \neq NP$



Şekil:  $P = NP$

# NP Problemler



**Şekil:**  $P = ? NP$  sorusu; 24 Mayıs 2000 tarihinde, Clay Matematik Enstitüsü (CMI) tarafından hazırlanan 7 soruluk “Milenyum Problemleri” listesinin ilk sırasında yer almaktadır. Milenyum problemlerinin her birinin ilk doğru yanıtının ödülü 1.000.000 \$ ‘dır.

