

# ISE 315 Yapay Zekaya Giriş



**Otonom Araç Simülasyonu için Basit Yapay Zekâ Sistemi Geliştirme**

**Öğretim Üyesi: Dr. Öğr. Üyesi Esin Ayşe Zaimoğlu**

**Hazırlayan Grup: Grup 5**

**Aralık 2025**

## Özet

Bu proje, sezgisel olmayan gündelik arama problemlerinden başlayarak bilgisayar bilimlerinde kullanılan dört temel arama algoritmasının (DFS, BFS, Greedy Best-First Search ve A\*) gerçekçi bir otonom araç simülasyonu üzerinde karşılaştırılmasını amaçlamaktadır. Hazırlanan simülasyon ortamı; trafik ışıkları, yaya geçitleri, iki şeritli yollar ve düzensiz şekilde hareket eden diğer araçlar gibi karmaşık şehir içi sürüs koşullarını modellemektedir. Proje kapsamında geliştirilen aracı temsil eden “Agent” nesnesi, verilen bir hedef konuma ulaşmak için söz konusu arama algoritmalarından elde edilen yolları takip etmektedir. Bu çalışma, hem günlük arama davranışlarını algoritmik yaklaşımalarla ilişkilendirmekte hem de arama tekniklerinin performansını 100 farklı senaryo üzerinden deneysel olarak ortaya koymaktadır.

---

## 1. Giriş

Arama algoritmaları, günlük hayatımızda fark etmeden kullandığımız fakat bilgisayar bilimlerinde açık şekilde formüle edilen temel problem çözme yöntemleridir. Kaybolmuş bir nesneyi aramaktan, en kısa rotayı bulmaya veya karmaşık bir sistemde hedefe ulaşmaya kadar çok geniş bir alanda karşımıza çıkarlar. Bu nedenle çeşitli problemleri uygun bir arama alanına dönüştürmek ve doğru algoritmayı seçmek, verimli çözüm üretmenin en kritik basamaklarından biridir.

Bu çalışmada öncelikle gündelik hayatımızdan basit fakat anlamlı bir örnek üzerinden arama davranışlarının sezgisel karşılıkları açıklanmaktadır; ardından geliştirilen otonom araç simülasyonu üzerinde dört temel arama algoritmasının performansları incelenmektedir.

---

## 2. Gündelik Yaşamdan Arama Problemi: “Kaybolan Çorap” Senaryosu

Herkesin her sabah yaşayabildiği “kaybolmuş çorap” problemi, arama algoritmalarını sezgisel olarak betimlemek için oldukça uygun bir analojidir.

### 2.1. DFS Benzeri Davranış

Bazı insanlar bir odayı tamamen aramadan başka bir yere geçmez. Örneğin: Yatağın üstüne bakılır, yastığın altı karıştırılır, komodin çekmeceleri açılır, halının altına bakılır ve ancak tüm bu işlemler tamamlanınca başka bir odaya geçilir. Bu davranış, **derinliği önceleyen arama (DFS)** mantığını temsil eder.

### 2.2. BFS Benzeri Davranış

Bazılı ise tek bir oda üzerinde yoğunlaşmaz; önce odaların yüzeysel alanlarını dolaşır.

Örneğin:

- Önce yatak odasının zeminine göz atar,
- Ardından hızlıca mutfaga geçer,

- Sonra banyoya bakar,
- Tekrar yatak odasına dönerek bu kez daha detaylı bakmaya başlar.

Bu, **genişliği önceleyen arama (BFS)** mantığına karşılık gelir. Tüm “katmanlar” eşit derecede ve sırayla incelenir.

### 2.3. Greedy Benzeri Davranış

Bazı insanlar ilk bakışta en mantıklı görünen yerden başlar. “Mantıken çorap çekmecededir” diyerek direkt oraya yönelir. Buzdolabına bakmaz, banyodaki dolabı açmaz, çünkü oradaki olasılık düşüktür. Bu yaklaşım, **Greedy Best-First Search**'ün yalnızca hedefe yakınlığı (heuristic) dikkate alan davranışını temsil eder.

### 2.4. A\* Benzeri Davranış

Daha sistematik düşünen kişiler ise hem mantıklı görünen yerleri hem de oralara ulaşmanın ne kadar zahmetli olduğunu hesaba katar:

“Çorap çekmecesi yakın ve mantıklı, dolayısıyla önce buraya bakayım.”

Bu düşünme biçimini, hem maliyeti  $g(n)$  hem de hedefe sevgisel uzaklığı  $h(n)$  değerlendiren **A\*** algoritmasının mantığını temsil eder.

Bu örnekler, arama algoritmalarının günlük yaşamla doğrudan ilişkilendirilebileceğini göstermekte ve sunumun sonraki teknik bölümleri için güçlü bir motivasyon oluşturmaktadır.

---

## 3. Proje Simülasyon Ortamı

Bu projede şehir içi trafiğini temsil eden bir simülasyon ortamı geliştirilmiştir. Harita iki şeritli yollar, çeşitli yönlerde uzanan bağlantılar, trafik ışıkları ve yaya geçitlerinden oluşmaktadır.

### 3.1. Rastgele Davranışlı Araçlar

Simülasyonda normal araçları temsil eden nesneler bulunmaktadır. Bu araçlar:

- Yalnızca önlerindeki 5 kareyi sensörleriyle algılar,
- Hızlanma/yavaşlama kurallarına uyar,
- Kırmızı ışıkta durur,
- Yaya geçidinde yayalara yol verir,
- Gelişigüzel bir rota boyunca şehir içinde dolaşır.

Bu davranışlar tamamen kural tabanlıdır; araçlar bağımsız karar vermez.

### 3.2. Arama Yapan “Agent” Aracı

Asıl inceleme konusu olan Agent ise bir hedef koordinatına **arama algoritmasının ürettiği rotaya göre** ilerleyen özel bir araçtır. Agent:

- Normal araçlarla aynı kurallara uyar,
- Ancak ilerlemesi için dışarıdan verilen bir **path** listesine ihtiyaç duyar,
- Bu path listesi, ilgili arama algoritmasının **search()** fonksiyonu tarafından oluşturulur.

Bu yapı sayesinde dört algoritmanın davranışları hem görsel hem ölçülebilir şekilde test edilmektedir.

---

## 4. Kullanılan Arama Algoritmaları

Tüm algoritmalar ortak bir üst sınıf olan **SearchVisualizer** üzerinden türetilmiştir. Her biri sadece kendi arama stratejisine özgü **search()** fonksiyonunu tanımlar.

### 4.1. DFS (Depth-First Search)

- Bir dalı sonuna kadar takip eder, çıkmazsa geri döner.
- Hızlı sonuç verebilir fakat **optimal değildir**.
- Bazı hedef konumlarında çok verimsiz olabilir.

### 4.2. BFS (Breadth-First Search)

- Haritayı halka halka tarar.
- **Daima en kısa yolu bulur** (birim ağırlıklı graf).
- Fakat tüm dalları genişletecek ilerlediği için **çok yavaştır ve yüksek bellek kullanır**.

### 4.3. Greedy Best-First Search

- Sadece hedefe sezgisel uzaklığı (Manhattan Uzaklığı) göre ilerler.
- Çoğu durumda çok hızlıdır.
- Fakat **yanlış bir yönlendirici sezgi** durumunda gereksiz sapmalara neden olabilir.

### 4.4. A\*

- Hem maliyet hem sezgiyi toplar:

$$f(\mathbf{n}) = g(\mathbf{n}) + h(\mathbf{n})$$

- Greedy kadar hızlı, BFS kadar güvenilir olabilir.
- Sezgi fonksiyonu uygun seçildiğinde en ideal yöntemdir.

## 5. Algoritmaların Gerçek Zamanlı Gösterimi

Simülasyon sırasında Agent belirli bir başlangıç konumundan hedefe ulaştırılmak istenir. Kullanıcı arayüzünden sırasıyla DFS, BFS, Greedy ve A\* çalıştırılabilir.

### 5.1. DFS Denemesi

- Coğu durumda hızlıca bir yol bulur.
- Ancak yol optimal değildir.
- Gidilemeyecek kadar derin dallara sapabilir.

### 5.2. BFS Denemesi

- Her zaman optimal yolu bulur.
- Ancak tüm haritayı neredeyse tamamen taradığı için gereksiz zaman ve hafiza kullanır.

### 5.3. Greedy Denemesi

- Hedef yönüne doğru “çekilen” bir davranış sergiler.
- Genellikle hızlı ve hafiftir.
- Fakat engelle karşılaşınca çok dolambaçlı bir arayış sürecine girebilir.

### 5.4. A Denemesi\*

- Greedy'nin hızını, BFS'nin optimalitesini birleştirir.
- Hem sezgiye hem de aşılmış mesafeye bakar.
- Pratikte çoğu zaman **en dengeli algoritmadır**.

---

## 6. Dinamik Ortam ve Yeniden Planlama

Simülasyon statik değildir; yol ortasında bir engel belirdiğinde (örneğin kullanıcı bir yol parçasını silerse) Agent'in mevcut rotası geçersiz hale gelir.

Bu durumda:

- Agent durur,
- Konumunu yeniden algılar,
- Seçili algoritmayı tekrar çalıştırarak **yeni bir rota** oluşturur.

Bu özellik, gerçek dünya otonom sürüsistemlerindeki yeniden planlama mantığının sadeleştirilmiş bir karşılığıdır.

## 7. Deneysel Değerlendirme (100 Senaryo)

Proje kapsamında Agent'ın başlangıç ve hedef konumlarının rastgele seçildiği **100 farklı senaryo** oluşturulmuştur.

Her algoritma için ölçülen temel metrik:

### 7.1. “Keşfedilen Kare Sayısı” (Search Cost)

Bu değer, algoritmanın arama sırasında ziyaret ettiği toplam harita karesini ifade eder. Gerçek bellek karmaşıklığının birebir karşılığı değildir ancak **arama maliyeti** açısından en mantıklı ölçümdür.

### 7.2. Elde Edilen Yol Uzunluğu (Final Solution Cost)

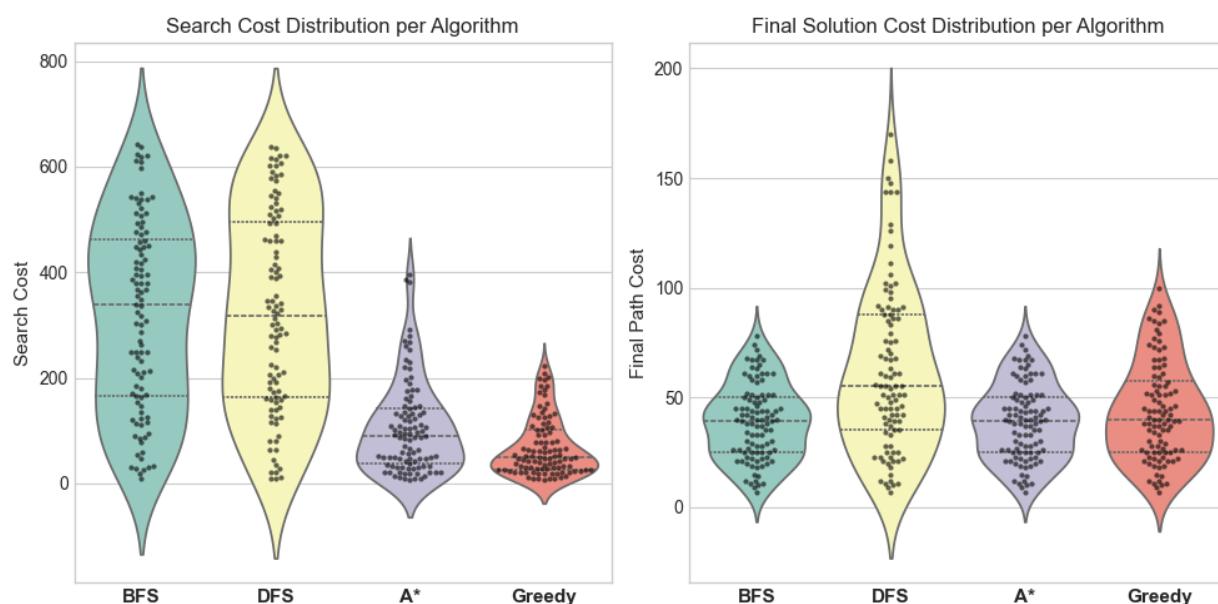
Algoritmanın bulduğu yolun optimum olup olmadığını gösterir.

---

## 8. Sonuçlar

100 deneye ilişkin grafiksel özet raporu aşağıda verilen **Şekil 1**'de sunulmaktadır.

- BFS ve DFS en fazla kareyi ziyaret edenler, dolayısıyla en maliyetli algoritmalarıdır.
- Greedy ve A\* ortalama keşif sayısında en verimli yöntemlerdir.
- DFS bazen hızlı olsa da optimal yolu garanti etmediği için dengesiz sonuçlar üretmiştir.
- A\* genel performans açısından en tutarlı ve dengeli çözümdür.



**Şekil 1: Dört arama algoritmasının 100 deney üzerindeki arama maliyetleri (kare sayısı) ve yol uzunlukları karşılaştırması**

## **9. Kaynak Kod ve Proje Reposu**

Projenin tamamı GitHub üzerinden erişime açiktır:

**GitHub Repusu:** <https://github.com/MamoMGD1/AutonomousAI>

---

## **10. Genel Değerlendirme**

Bu çalışma, arama algoritmalarının hem sezgisel hem de formel yönlerini gerçekçi bir otonom araç simülasyonunda bir araya getirmiştir. Gündelik bir problemi arama perspektifinden ele almak hem sezgisel motivasyon sağlamış hem de algoritmaların davranışlarını kavramayı kolaylaştırmıştır. Gerçek zamanlı görselleştirme sayesinde arama stratejilerinin güçlü ve zayıf yönleri açıkça gözlemlenebilmiştir.

Yapılan deneysel analizler, teorik bilginin pratikle örtüştüğünü göstermiş ve özellikle A\* algoritmasının gerçekçi uygulamalarda neden yaygın şekilde tercih edildiğini açıkça ortaya koymuştur.