

# Yapay Zekâ Uygulamaları

## Ders 4

Ali Mertcan KOSE Msc.  
amertcankose@ticaret.edu.tr

İstanbul Ticaret Üniversitesi



İSTANBUL TİCARET  
ÜNİVERSİTESİ

# Arama İle Problem Çözme

Problem çözme, yapay zeka tabanlı birçok uygulama için temeldir. Gerçek dünyada iki tür problem vardır. Bunlardan birincisi, bir açının sinüsünün ya da bir değerin karekökünün hesaplanması gibi problemlerdir. Bunlar deterministik yöntemler kullanılarak çözülebilir ve başarı garanti edilir. Ancak gerçek dünyada çok az problem basit çözümlere dönüşebilir. Gerçek dünya problemlerinin çoğu ancak bir çözüm arayarak çözülebilir. Aslında yapay zeka bu tür problemlerin çözülmesiyle ilgilenir.

# Arama İle Problem Çözme

*Problem çözme*, gözlenen verilerden çözümler üretme sürecidir. Bu süreçte bir problem bir dizi hedefle karakterize edilir, daha sonra bir dizi nesne ve bir dizi işlem gerçekleştirilerek problem çözülmeye çalışılır. Fakat bu seçilenler kötü tanımlanmış olabilir ya da problem çözme sırasında gelişebilir. Dolayısıyla problem uzayı soyut bir uzaydır.

# Arama İle Problem Çözme

*Bir Problem Uzayı*, herhangi bir nesne kombinasyonu üzerinde herhangi bir işleç kombinasyonunun uygulanmasıyla üretilebilecek tüm geçerli durumları kapsar. Problem uzayı bir ya da daha fazla çözüm içerebilir. Çözüm, hedeflere ulaşan işlemlerin ve nesnelerin bir kombinasyonudur.

# Arama İle Problem Çözme

*Arama*, bir problem uzayında çözüm arama anlamına gelir. Arama farklı türede arama kontrol stratejileriyle ilerler. Derinlik öncelikli arama ve genişlik öncelikli arama iki yaygın arama stratejisidir. Yapay zekadaki akıllı etmenler ya da problem çözme etmenleri, belirli bir sorunu çözmek ve tek sonucu sağlamak için çoğunlukla bu arama stratejilerini ya da algoritmaları kullanırlar. Aslında problem çözme etmenleri hedefe dayalı etmenlerdir ve biçimsel temsile sahiptirler.

# Genel Problem Çözme

Problem çözme yapay zeka için kilit ilgi alanları olmuştur. Problem çözme gözlenen ya da verilen verilerden çözümler üretme sürecidir. Bununla birlikte, doğrudan veriden çözüme gitmek ya da doğrudan yöntemleri kullanmak her zaman mümkün değildir. Bunun yerine, problem çözmede genellikle dolaylı ya da modele dayalı yöntemlerin kullanılması gerekir.

# Problemin Tanımlanması

Bir problem genellikle öğeleri ve ilişkileri ile tanımlanır. Bir problemin biçimsel bir tanımını sağlamak için aşağıdakilerin yapılması gerekir.

a.İlgili nesnelerin bazı olanaksız olanları da dahil olası durumları içeren bir durum uzay tanımlanır.

b.Problem çözme sürecinin başlayabileceği olası durumları tanımlayan bir ya da daha fazla durum belirtilir ve bu durumlara başlangıç durumları denir.

# Problemin Tanımlanması

c.Probleme kabul edilebilir çözüm olabilecek bir ya da daha fazla durum belirtilir ve bu durumlara hedef durumları denir. Mevcut eylemleri açıklayan bir dizi kural belirtilir ve daha sonra problem, bir başlangıç durumundan bir hedef durumuna giden bir yol bulunana kadar problem uzayında ilerlemek için uygun bir kontrol stratejisi ile birlikte kurallar kullanılarak çözülebilir. Genel olarak bu işlem arama olarak bilinir ve problem çözme sürecinin temelidir.

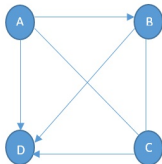


# Problem Uzayı

Problem uzayı aşağıdaki şekilde görüldüğü gibi yönlendirilmiş bir grafikte temsil edilir. Burada düğümler arama durumunu temsil eder ve yollar durumu değiştirmek için uygulanan eylemleri temsil eder. Bir arama algoritmasını basitleştirmek için, problem uzayını şekilde görüldüğü gibi bir ağaç olarak mantıksal ve programlı olarak temsil etmek genellikle uygundur.

# Problem Uzayı

GRAF



AĞAÇ

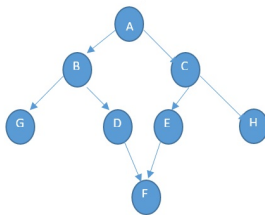


Figure 1: Arama Ağacı

Bir ağaç genellikle bir aramanın karmaşıklığını bir maliyet karşılığında azalatabilir. Burada maliyet GRAF şeklindeki düğüm B ve düğüm D gibi birçok kez bağlanmış olan ağaçtaki bazı düğümlerin çoğaltılmasından kaynaklanmaktadır.

Problem çözme önceden de ifade ettiğimiz gibi; tanımlı hedefe/çözüme ulaşmak için kullanılabilecek olası eylemler kapsamındaki sistematik araştırma olarak ifade edilebilmektedir. Genel ve özel amaçlı yöntemler problem çözme noktasında kullanılan iki farklı uygulama yöntemine tekabül etmektedir.

Bir problem, öğeleri ve ilişkileri ile tanımlanır. Bir problemin her anında, Öğelerin nasıl seçileceğini söyleyen belirli tanımlayıcıları ve ilişkileri vardır. Durum, belirli bir andaki öğelerin temsilidir. Tüm olası durumlar arasında, başlangıç noktasını belirten başlangıç durumu ve hedef durumu niteleyen son durum iki özel durum vardır.

# Problemin çözümü

Durum uzayında, bir çözüm başlangıç durumundan bir hedef durumuna ya da bazen sadece bir hedefe giden yoldur. Her problem çözümünde, çözüm maliyeti işlevi vardır ve her yola sayısal bir maliyet atar. Ayrıca eylemlerin durumlara uygulanmasının maliyetini de verir. Çözüm kalitesi yol maliyet fonksiyonu ile ölçülür ve en uygun çözüm, tüm çözümler arasında en düşük yol maliyetine sahip olan çözümdür. Çözüm olarak herhangi biri, en uygunu ya da tümü kabul edilebilir. Maliyetin önemi, probleme ve problemin çözüm türüne bağlıdır.

# Problemin Tanımlanması

Bir problem dünyanın geçerli durumu, dünyanın bir durumunu diğerine dönüştürebilen eylemler ve dünyanın arzu edilen durumlarının tanımlanmasından oluşur. Bunlar her problem çözümünde benzer bir şekilde tanımlanır.

# Problemin Tanımlanması

- Durum uzayı, açık ya da örtük olarak tanımlanır. Bir durum uzayı, bir sorunu çözmek için gerekli olan her şeyi ve sorunu çözmek için gerekli olmayan hiçbir şeyi tanımlamalıdır.
- Başlangıç durumu, mevcut durumdur.
- Hedef durumu, yerine getirilmesi gereken koşullardır. Dünyanın arzu edilen durumunun tanımını içerir.
- Eylemler, durumu değiştirecek eylemleri ifade eder ve bir durumu diğerine dönüştürebilecek eylemler bulunur. Genel olarak eylemler dünyanın durumunu kısmi açıklayan ve eylemi gerçekleştirmek için doğru olması gereken ön koşullar ve bir sonraki durumun nasıl oluşturulacağını anlatan talimatlar oluşur.



# Problemin Tanımlanması

- Alanın öğeleri, başlangıç noktası bilgisi gibi problemle ilgisi olan bilgilerdir.
- Problem çözümü, başlangıçtaki geçerli durumu bir hedef durumuna dönüştüren sıralı bir eylem dizisi, probleme çözüm bulmaktadır.
- Kısıtlar, çözüm kalitesini belirlemelidir. En kısa diziyi bulma, maliyeti tanımlayan en ucuz diziyi bulma ya da herhangi bir diziyi mümkün olduğunca çabuk bulmayı ifade eder.

Arama, her bir yapay zeka kullanım durumunun temel sürecidir. Belirli bir veri yapısı içinde, bu veri noktasından ilgili bilgileri belirlemek ve ardından çıkarmak için arama algoritmaları kullanılır. Başka bir deyişle arama, durumlardan gelen yolu bulmak için durumların sistematik olarak incelenmesidir ve genellikle bilgi eksikliğinden kaynaklanır. Başlangıç ya da kök durumundan hedef durumuna en iyi cevaba ulaşmak için bilgi alternatiflerini araştırır. Sonuç olarak elde edilen arama algoritması çıktısı bir çözümdür, yani başlangıç durumundan hedef denemesini karşılayan bir duruma giden yoldur. Arama genel amaçlı problem çözme için bir yaklaşımdır ve arama uzayını temsil eden diyagramda belirli özelliklere sahip düğümleri bulmakla ilgilenir.

# Arama Algoritmaları

Yapay zeka uygulamalarında birçok arama algoritmaları kullanılmaktadır. Karmaşık problemler için geleneksel algoritmalar çözümü bazı pratik zaman ve mekan sınırları içinde bulamamaktadır. Sezgisel işlevleri kullanan algoritmalara sezgisel algoritmalar denir. Sezgisel algoritmalar gerçekten akıllı değildir ama daha iyi performans elde ettikleri için akıllı görünürler. Bu algoritmalar, arama yolunu yönlendirmek için verilerden gelen geri bildirimden yararlandıkları için daha verimlidir. Sonuç olarak, sezgisel işlevler kullanılarak bir çok özel teknik geliştirilmiştir.

# Arama Algoritmaları

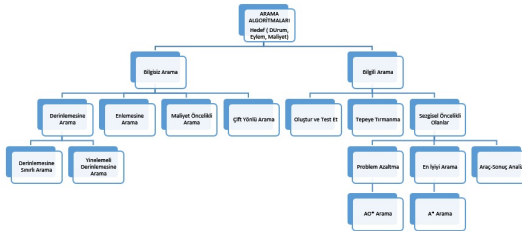


Figure 2: Arama algoritma çeşitleri

# Arama algoritma çeşitleri

Arama algoritmaları, arama teknikleri ya da arama stratejileri olarak da adlandırılmaktadır. Yukarıda da belirtildiği gibi arama algoritmalarını genel olarak iki başlık altında toplamak mümkündür.

# Arama algoritma çeşitleri

**Bilgisiz Arama:** Bilgisiz arama, kör kapsamlı ya da kaba kuvvet arama olarak da adlandırılır. Aramaya rehberlik etmek için problem hakkında hiçbir bilgi kullanmaz ve bu nedenle çok verimli olmayabilir. Bilgisi aramada en çok bilinen derinlemesine Arama ve türleri genellikle son giren ilk çıkar yığın mantığıyla çalışırlar. Derinlemesine Sınırlı Aramada sabit derinlik sınırı uygulanır. Yinelemeli Derinlemesine Aramada sabit derinlik sınırı kademeli olarak arttırılır. Enlemesine Arama, İlk Giren İlk Çıkar Kuyruk modeliyle Çalışır. Maliyet Öncelikli Aramada ise Hedefe yönelik öncelik söz konusudur. Burada  $n$  düğüm sayısıdır.

# Arama algoritma çeşitleri

**Bilgili Arama:** Sezgisel ya da akıllı arama olarak da adlandırılan bilgili aramada, aramaya rehberlik etmek için problem hakkındaki bilgileri kullanır. Genellikle bir hedef durumuna olan mesafeyi tahmin eder ve bu nedenle verimlidir. Ancak arama her zaman mümkün olmayabilir. Bilgili aramalar genellikle sezgisel teknikler kullanırlar. Sezgisel öncelikli algoritmalardan  $A^*$  ve  $AO^*$  arama başarımında sezgisellik ve hedef önceliklidir.

# Arama Algoritmalarının Özellikleri

Arama algoritmalarından istenilen belli başlı özellikler vardır. Bu özellikler aynı zamanda farklı türdeki arama algoritmalarının verimliliğini karşılaştırmak için de kullanılır. Başarılı bir arama algoritmasının temel özellikleri şunları içerir.



# Arama Algoritmalarının Özellikleri

**Tamlık:** Herhangi bir rastgele girdi için en azından herhangi bir çözüm varsa, bir çözüm döndürmeyi garanti ediyorsa, bir arama algoritmasının tamamlanmış olduğu kabul edilir.

**İdeallik:** Bulunan bir çözüm, tanımlanan tüm çözümler arasında en iyisi ise ya da en düşük yol maliyetine sahipse, bu çözüme en uygun çözüm denir. Zaman karmaşıklığı: Bir algoritmanın görevini tamamlaması için geçen süreye zaman karmaşıklığı denir.

# Arama Algoritmalarının Özellikleri

**Uzay Karmaşıklığı:** Algoritmanın arama yaparken herhangi bir zamanda aldığı maksimum depolama ya da bellek alanı uzay karmaşıklığıdır.

# BİLGİSİZ ALGORİTMALAR

Bilgisiz arama, kaba kuvvetle çalışan bir genel amaçlı arama algoritmaları sınıfıdır. Bilgisiz arama algoritmaları, ağacı nasıl geçeceği dışında durum ya da arama alanı hakkında ek bilgiye sahip değildir, bu nedenle kör arama alanı hakkında ek bilgiye sahip değildir, bu nedenle arama ya da kaba kuvvet arama olarak da adlandırılır.

## Derinlemesine Arama

Derinlik öncelikli arama, bir ağaç ya da grafik veri yapısında gezinmek için özyinelemeli bir algoritmadır. Derinlemesine arama Son giren İlk Çıkar (LIFO) stratejisini kullanır ve bu nedenle yığın kullanılarak uygulanabilir. Derinlemesine arama olarak da adlandırılır. Çünkü kök düğümünden başlar ve bir sonraki yola geçmeden önce her yolu en derin düğümüne kadar takip eder.

# BİLGİSİZ ALGORİTMALAR

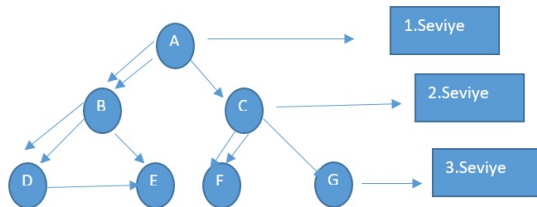


Figure 3: Derinlemesine Arama

# BİLGİSİZ ALGORİTMALAR

Şekilde bir arama ağacı verilmiştir. Bu ağaçta başlangıç durumu A, hedef durumu ise F'dir. Bu durumda A başlangıç durumundan başlar ve ardından B'ye gider ve ardından D'ye gider. D'den B'ye geri dönmesi gerekir. Ancak B zaten ziyaret edildiği için bir sonraki E'ye genişler ve sonra B'ye geri dönmesi gerekir, A'da daha önce ziyaret edildiği için mbce C'ye gidiyor. Sonra F'ye gidiyor ve hedef durumu olduğu için orada duruyor. Sonuç olarak geçiş yolu:  $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow C \rightarrow F$  şeklindedir.

# BİLGİSİZ ALGORİTMALAR

Derinlemesine arama algoritmasının temel özellikleri aşağıdaki gibi verilebilir.

**Tamlık:** Sınırlı bir arama ağacı içindeki her düğümü genişleteceğinden Derinlemesine arama algoritması sonucu durum uzayında tamamlanmıştır. Zaman Karmaşıklığı: Derinlemesine aramanın zaman karmaşıklığı algoritma tarafından geçilen düğüm sayısına eşdeğer olacaktır.

**Uzay karmaşıklığı:** Derinlemesine arama algoritmasının kök düğümünden yalnızca tek bir yolu saklaması gerekir

**İdeallik:** Hedef düğüme ulaşmak için çok sayıda adım ya da yüksek maliyet oluşturabileceğinden, Derinlemesine arama algoritması idealdir.

## Derinlemesine Aramanın Avantajları:

- Derinlemesine arama kök düğümden geçerli düğüme giden yolda yalnızca bir düğüm yığını depolaması gerektiğinden enlemesine arama ya göre çok daha az bellek gerektirir.
- Eğer doğru yolda ilerliyorsa, hedef düğüme ulaşmak enlemesine arama algoritmasına göre daha az zaman alır. Zaman karmaşıklığı daha düşüktür.

## Derinlemesine Aramanın Dezavantajları:

- Pek çok durumun yeniden ortaya çıkma olasılığı olduğu için çözümü bulmanın her zaman garantisi yoktur.
- Derinlemesine arama algoritması derinlerde arama yapar ve bazen sonsuz döngüye gidebilir. Çünkü hesaplama bir şekilde bitmeyebilir ve sınırsızca devam edebilir.



## Derinlemesine Sınırlı Arama

Derinlemesine sınırlı arama algoritması, önceden belirlenmiş bir derinlik seviye sınırı dışında derinlik öncelikli aramaya benzer. Derinlemesine sınırlı arama, derinlemesine öncelikli aramada oluşabilecek sonsuz yolun dezavantajını ortadan kaldırabilir. Bu algoritmada, derinlik sınırındaki düğüm, daha fazla ardıl düğümü yokmuş gibi işlem yapar ve belirtilen derinlik sınırı seviyesinde herhangi bir çözüm bulamazsa uygulamadan çıkar. Derinlemesine sınırlı arama, iki başarısızlık koşuluyla sonlandırılabilir:

# BİLGİSİZ ALGORİTMALAR

**Standart Başarısızlık:** Verilen problemin herhangi bir çözümü olmadığını gösterir. **Kesme Hata Değeri:** verilen limit dahilinde problem için bir çözüm olmadığını gösterir. **Tamlık:** Çözüm derinlik sınırının üzerindeyse Derinlemesine sınırlı algoritması tamamlanır.

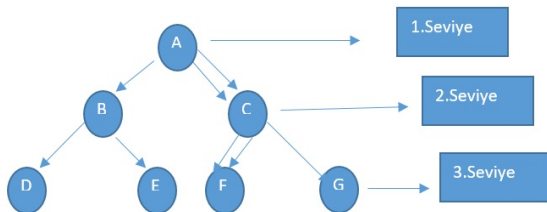


Figure 4: Derinlemesine Sınırlı Arama örneği

# BİLGİSİZ ALGORİTMALAR

A'yı başlangıç düğümü ve C'yi hedef durumu ve derinlik seviye sınırı 2 olarak alındığında geçiş önce A düğümü ile başlar ve daha sonra bir sonraki seviye 2'ye gider ve hedef durumu C'ye ulaşarak durur. Böylece geçişi durdurur. Hedef düğüm olarak C ve derinlik seviye sınırı 1 olarak verildiğinde, hedef düğüm verilen limit dahilinde olmadığı için algoritma herhangi bir yol döndürmeyecektir. Benzer şekilde hedef düğümü F ve derinlik seviye sınırını 2 olarak verilirse, yol A, C, F olacaktır. Şekilde görüldüğü gibi aynı Derinlemesine Sınırlı Arama Algoritması hedef noktası G için, derinlik sınırı 2 iken uygun yol cevabını verirken, derinlik seviye sınırı 1 olduğunda Hedef düğüme uygun yol yok cevabını vermektedir.

## Derinlemesine Sınırlı Arama Algoritmasının Avantajları:

- Derinlemesine sınırlı arama, diğer arama tekniklerine kıyasla daha az bellek gerektirir. Derinlemesine Sınırlı Arama Algoritmasının Dezavantajları:
- Derinlemesine sınırlı aramanın eksik olması en önemli dezavantajıdır.
- Problemin birden fazla çözümü varsa en uygun çözümü vermeyebilir.

## Yinelemeli Derinlemesine Arama

Yinelemeli derinlemesine arama algoritması, derinlik öncelikli arama ve genişlik öncelikli aramanın birleşimidir. Bu arama algoritması, Genişlemesine aramanın hızlı arama ve derinlemesine aramanın bellek verimliliği avantajlarını birleştirir. Bu arama algoritması, belirli bir derinlik sınırına kadar derinlemesine arama gerçekleştirir ve hedef düğüm bulunana kadar her yinelemeden sonra derinlik sınırını artırmaya devam eder. Böylece en iyi derinlik sınırını bulur. Yinelemeli derinlemesine arama algoritması, arama alanı büyük olduğunda ve hedef düğümünün derinliği bilinmediğinde, bilgisiz arama için yararlıdır.

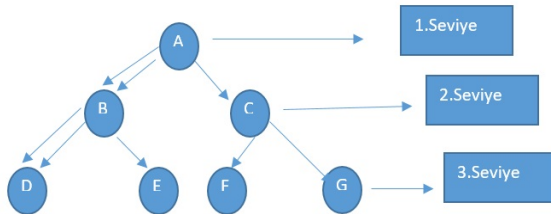


Figure 5: Yinelemeli Derinlemesine Arama Örneği

# BİLGİSİZ ALGORİTMALAR

Şekildeki örnekte ağaçta Başlangıç durumu A düğümü, Hedef durumu E düğümü olsun. Yinelemeli Maksimum derinlik seviye sınır 3 alınarak Yinelemeli derinlemesine arama çalıştırılsın. Bu durumda algoritma çalışmaya başladığında, A başlangıç düğümünden bir sonraki seviyeye geçer ve E'yi arar. Bulamayarak bir sonraki seviye geçer ve E'yi bulur. Geçiş yolu  $A \rightarrow B \rightarrow E$  olur

## Enlemesine Arama

Enlemesine arama ya da başka bir deyişle genişlemesine arama bir ağaç ya da grafi araştırmak için kullanılan en yaygın arama stratejilerden birisidir. Genellikle kök düğümden başlar ve komşu düğümleri inceler ve ardından bir sonraki seviyeye geçer. Dolayısıyla bu algoritma bir ağaç ya da grafikte enine arama yapar, bu nedenle buna genişlemesine arama algoritması denir. Genişlemesine arama algoritması, ağacın kök düğümünden aramaya başlar bir sonraki seviyenin düğümlerine geçmeden önce mevcut seviyedeki tüm ardıl düğümleri genişletir. Çözümüne ulaşmanın en kısa yolunu verdiği için ilk giren ilk çıkar (FIFO) stratejisini kullanır. Bu algoritma genellikle verilen problemin çok küçük olduğu ve uzay karmaşıklığının dikkate alınmadığı durumlarda kullanılır.



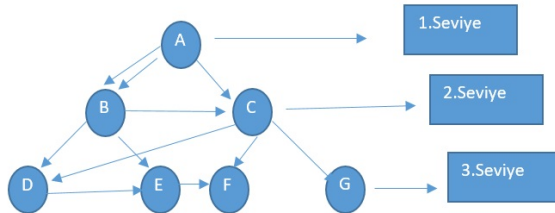


Figure 6: Enlemesine Arama Örneği

# BİLGİSİZ ALGORİTMALAR

Şekildeki arama ağacı üzerinde enlemesine arama algoritması ile arama yapılır. Burada başlangıç durumu olarak A düğümü ve hedef durumu olarak D düğümü alınsın. Enlemesine arama algoritması önce başlangıç durumu ile başlar. Daha sonra bir sonraki seviyeye geçer ve hedef durumuna ulaşana kadar tüm düğümleri ziyaret eder. Bir örnekte, A'dan sonra bir sonraki seviyeye geçtiğinde B ve C düğümlerini ziyaret eder. Ardından bir sonraki seviyeye seyahat eder ve D, E, F ve G düğümlerini ziyaret eder. Ancak burada hedef durumu F düğümü olarak tanımlandığı için F'de duracaktır.

Enlemesine arama algoritmasının avantajları:

- Verilen problem için herhangi bi çözüm varsa, Enlemesine arama algoritması mutlaka bir çözüm döndürecektir.
- Belirli bir problem için birden fazla çözüm varsa, Enlemesine arama algoritması en az adım gerektiren gerektiren minimum çözümü ya da en uygun çözümü verir.

Enlemesine arama algoritması dezavantajları:

- Enlemesine arama algoritması mevcut seviyedeki tüm düğümleri bellekte tutar ve ardından bir sonraki seviyeye geçer. Bu nedenle düğümleri depolamak için çok fazla bellek gerekir
- Hedef durum kök düğümünden uzaktaysa, çözüme ulaşmak enlemesine arama algoritmasında daha fazla zaman alır.

## Maliyet Öncelikle Arama

Maliyet öncelikli arama algoritması, ağırlıklı bir ağaç ya da grafin üzerinden geçmek için kullanılan en iyi arama algoritması olarak kabul edilmektedir. Bu algoritma, her kenar için farklı bir maliyet mevcut olduğunda devreye girer. Maliyet öncelikli arama algoritmasının birincil amacı, en düşük birikimli maliyete sahip hedef düğüme giden bir yol bulmaktır. Bu algoritma, düğümleri kök düğümü oluşturan yol maliyetlerine göre genişletir. En uygun maliyetin talep edildiği herhangi bir grafi ya da ağacı çözmek için kullanılabilir. Öncelik sırası tarafından maliyete en yüksek öncelik verir. Maliyet öncelikli arama algoritması tüm kenarların yol maliyeti aynıysa genişlemesine arama algoritmasına eşdeğerdir.

# BİLGİSİZ ALGORİTMALAR

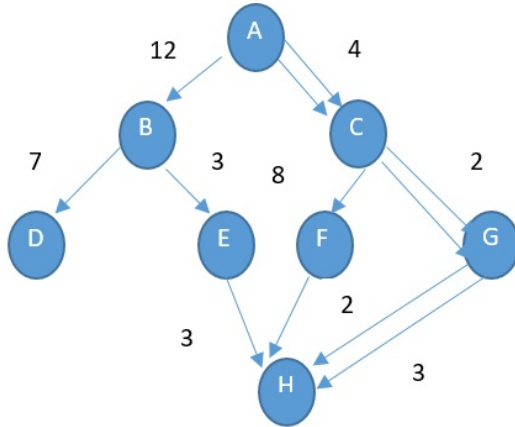


Figure 7: Maliyet öncelikli Arama Örneği

# BİLGİSİZ ALGORİTMALAR

Şekilde maliyet öncelikli arama için örnek bir arama ağacı verilmiştir. Ağaçtaki her düğüm maliyet değeri belirtilmiştir. Burada başlangıç düğümü A ve hedef düğümü de H olarak maliyet öncelikli arama yapılsın. A düğümünden H düğümüne ulaşan yollara ve maliyetlerine bakılırsa:

$$A \rightarrow B \rightarrow E \rightarrow H \rightarrow 18$$

$$A \rightarrow C \rightarrow F \rightarrow H \rightarrow 14$$

$$A \rightarrow C \rightarrow G \rightarrow H \rightarrow 9$$

Maliyet öncelikli arama algoritmasının avantajları:

- Yolların seçimi en düşük maliyete dayandığından bu algoritma en uygundur.

Maliyet öncelikli arama algoritmasının dezavantajları:

- Algoritma, en düşük yola ulaşmak için kaç adım atıldığını dikkate almaz. Bu yüzden sonsuz bir döngüye neden olabilir.



## Çift Yönlü Arama

Çift yönlü arama algoritması, hedef düğümü bulmak için biri ileri arama olarak adlandırılan bir başlangıç durumu ve diğeri geriye doğru arama olarak adlandırılan hedef düğümünden olmak üzere iki eşzamanlı arama yapar. Bunun için çift yönlü arama algoritması tek bir arama ağacını birinin aramayı ilk tepe noktasından, diğerrinin ise hedef tepe noktasından başlattığı iki küçük alt ağaca böler. Bu iki ağaç birbiriyle kesiştiğinde arama durur. Çift yönlü arama, hem başlangıç durumunun ve hem de hedef başlangıcın iyi tanımlanmasını ve dallanma faktörünün iki yönde aynı olmasını gerektirir. Çift yönlü arama algoritmasında enlemesine arama, derinlemesine arama ve derinlemesine sınırlı arama gibi algoritmalar kullanabilir.

# BİLGİSİZ ALGORİTMALAR

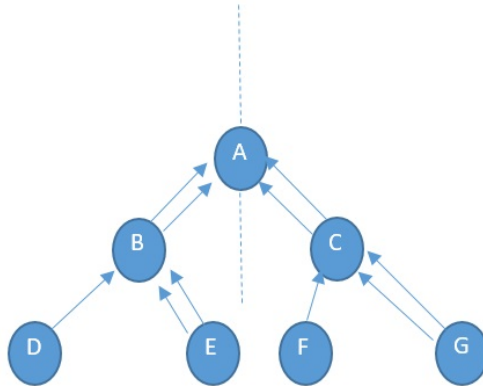


Figure 8: Çift yönlü Arama algoritması örneği

# BİLGİSİZ ALGORİTMALAR

Şekilde çift yönlü arama algoritması için bir örnek arama ağacı verilmiştir. Burada, başlangıç durumu E ve hedef durumu da G dir. Bir alt grafikte arama E düğümünden başlar ve diğerinde arama G düğümünden başlar. E düğümünden B düğümüne ve ardından A düğümüne ulaşılır. G düğümünden C düğümüne gider ve sonra A düğümüne ulaşır. Bu durumda her iki aramada A düğümünde buluşur ve geçiş durarak arama biter. Böylece geçiş yolu

$E \rightarrow B \rightarrow A \rightarrow C \rightarrow G$  olur.

Çift yönlü arama algoritmasının avantajları:

- Çift yönlü arama algoritması hızlıdır.
- Çift yönlü algoritması daha az bellek gerektirir. Çift yönlü arama algoritmasının dezavantajları:
- Çift yönlü arama ağacının uygulanması zordur.
- Çift yönlü aramada, hedef durumu önceden bilmelidir.

# BİLGİLİ ARAMA ALGORİTMALARI

Bilgili arama algoritmasına sezgisel arama da denir. Buraya kadar, arama uzayı hakkında herhangi bir ek bilgiye sahip olmadan, problemin tüm olası çözümleri için arama uzayına bakan bilgisiz arama algoritmalarından bahsedildi. Bilgili arama algoritmaları bilgisiz arama algoritmalarının aksine, hedefe ulaşmak için mesafe, hedefe ulaşmak için atılan adımlar, yolların maliyeti gibi detaylara ihtiyaç duyar ve bütün bunlar algoritmayı daha verimli hale getirir. Bu bilgiler, etmenlerin arama alanında daha az araştırma yapmasına ve hedef düğümü daha verimli bulmasına yardımcı olur.

# BİLGİLİ ARAMA ALGORİTMALARI

Bilgili arama algoritması, geniş arama alanları için daha kullanışlıdır. Bilgilendirilmiş arama algoritması sezgisel fikrini kullanır, bu nedenle sezgisel arama ya da yönlendirilmiş arama olarak da adlandırılır. Sezgisel arama algoritmalarında hedef durumuna mümkün olan en düşük maliyetle ulaşmak için Sezgisel işlev kullanılır. Bu işlev bir durumun hedefe ne kadar yakın olduğunu tahmin eder ve etmenin geçerli durumunu girdi olarak alarak etmenin hedefe ne kadar yakın olduğunun tahminini üretir. Sezgisel yöntem her zaman en iyi çözümü vermeyebilir, ancak makul bir sürede iyi bir çözüm bulmayı garanti eder.

## En iyi Arama

En iyi arama algoritması (Greedy Search) hem derinlemesine arama hem de genişlemesine arama özelliklerini kullanır. Sezgisel işlevi ve aramaları kullanarak her iki algoritmanın da avantajlarından yararlanmamızı sağlar. En iyi arama, o anda en iyi görünen yolu seçerek düğümü geçer ve sezgisel işlev kullanılarak en yakın yol seçilir. Bu algoritma her zaman o anda en iyi görünen yolu seçer. En iyi aramanın yardımıyla her adımda en umut verici düğüm seçilebilir. En iyi arama algoritmasında hedef düğüme en yakın düğümü genişletilir ve en yakın maliyet sezgisel işlevi ile tahmin edilir.

# BİLGİLİ ARAMA ALGORİTMALARI

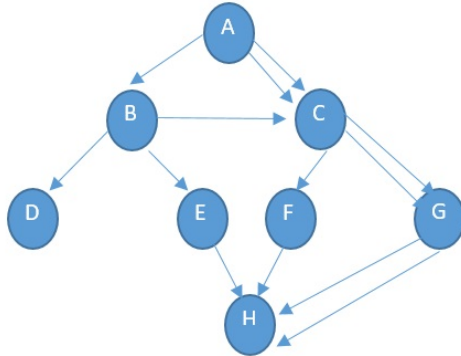


Figure 9: En iyiye arama algoritması için örnek arama ağacı ve sezgisel değerler



# BİLGİLİ ARAMA ALGORİTMALARI

Şekilde örnek arama ağacı ve sezgisel değerleri göz önünde bulundurularak en iyi arama algoritmasının çalışmasını inceleyelim. Burada A başlangıç düğümüdür ve H hedef düğümüdür. En iyi arama önce A düğümü ile başlar ve ardından sonraki komşu B ve C düğümler inceler. Burada B sezgisel değeri 12 ve C düğümünün sezgisel değeri 4'tür. Şu anda en iyi yol C düğümüdür dolayısıyla algoritma C düğümüne gider. C düğümünden komşu düğümler olan F ve G düğümlerini araştırır. F düğümünün sezgisel değeri 8 ve G düğümünün sezgisel değeri 2'dir. Dolayısıyla G düğümüne gider. G düğümünden sezgisel değeri 0 olan H düğümüne gider. Böylece hedef duruma ulaşmış olur.

En iyi arama algoritmasının avantajları:

- En iyi arama algoritması genişlemesine arama ve derinlemesine arama algoritmalarının da avantajlarını elde ederek her algoritma arasında geçiş yapabilir.
- En iyi arama algoritması genişlemesine arama ve derinlemesine arama algoritmalarından daha verimlidir.

# BİLGİLİ ARAMA ALGORİTMALARI

En iyi arama algoritmasının dezavantajları:

- En kötü senaryoda, bilgisiz derinlemesine arama gibi davranabilir.
- Derinlemesine arama algoritmalarında olduğu gibi sonsuz döngüye girebilir.

# BİLGİLİ ARAMA ALGORİTMALARI

Table 1: n iyi arama algoritması için örnek arama ağacı ve sezgisel değerler.

Düğüm	Sezgisel
A	13
B	12
C	4
D	7
E	3
F	8
G	2
H	0

# BİLGİLİ ARAMA ALGORİTMALARI

## A\*Arama

A\* arama algoritması en iyi arama algoritmasının en yaygın olarak bilinen şeklidir. A\* arama algoritması hem maliyet öncelikli arama ve hem de en iyi arama algoritmalarının birleşimidir. Daha iyi bellek kullanımı ile birlikte her ikisinin de avantajlarını kullanır. A\* arama algoritması sezgisel işlev kullanarak arama alanı boyunca en kısa yolu bulur. A\* arama algoritması en iyi yolu bulmak için düğümün maliyetinin ve sezgisel değerlerinin toplamını kullanır.

# BİLGİLİ ARAMA ALGORİTMALARI

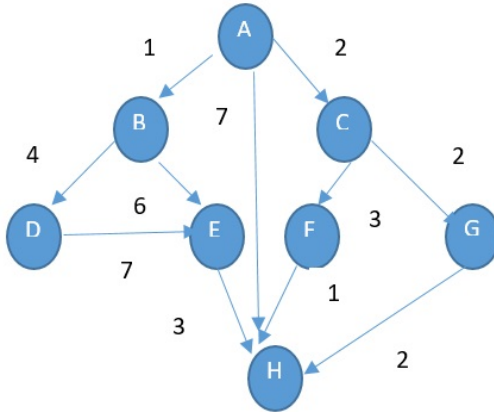


Figure 10: A\* Arama algoritması örneği

# BİLGİLİ ARAMA ALGORİTMALARI

Table 2: A\* Arama algoritması örneği.

Düğüm	Sezgisel
A	5
B	3
C	4
D	2
E	6
F	3
G	1
H	0

# BİLGİLİ ARAMA ALGORİTMALARI

Şekildeki örnek yol ağırlık değerleri olan arama ağacını ve sezgisel değerleri kullanarak  $A^*$  arama algoritmasının çalışması sırasıyla aşağıdaki gibidir. İlk olarak arama algoritması A düğümü ile başlayacak. A düğümünden, B, C ve H düğümlerine gidilebilir. Ancak  $A^*$  aramasının yolu belirlemek için önceliği yol maliyeti ve sezgisel değerin toplamı olacağı için, A düğümünden B düğüme, maliyet ve sezgisel değerlerin toplamı  $1+3 = 4$ , A'dan C'ye  $2+4= 6$  ve A'dan H'ye  $7+0=7$ 'dir. Burada uygunluk sayısı ya da en düşük maliyet 4 olduğu için A düğümünden B düğüme giden yol seçilir. Diğer yollar beklemeye alınacaktır.



# BİLGİLİ ARAMA ALGORİTMALARI

Bu durumda, B düğümünden D ya da E düğümlerine gidilebilir. Uygunluk sayılarına göz atılırsa A'dan B'ye ve oradan D'ye maliyet  $1 + 4 + 2 = 7$ 'dir. A'dan B'ye E'ye  $1 + 6 + 6 = 13$ 'tür. En düşük maliyet  $A > B > D$  arası 7 ile karşılaştırılır. Burada A'dan C'ye giden yol uygunluk sayısı 6 olduğu için daha az maliyetlidir. Bu nedenle A'dan C'ye seçilir ve diğer yollar beklemeye alınır.

# BİLGİLİ ARAMA ALGORİTMALARI

Bu durumda C düğümünden F ya da G düğümlerine gidebilir. A'dan C'ye ve oradan F'ye olan yolun maliyeti  $2+3+3=8$ 'dir. A'dan C'ye ve oradan G'ye olan yolun maliyeti  $2+2+1=5$ 'tir ve en düşük maliyettir. Bu aynı zamanda beklemedeki diğer yollardan daha düşük olduğu için A düğümünden G düğümüne olan yol seçilir.

# BİLGİLİ ARAMA ALGORİTMALARI

G düğümünden uygunluk sayısı  $2+2+2+0=6$  olan H düğümüne gidilebilir. 6 değeri beklemede olan diğer yolların maliyetinden daha düşük olduğu için  $A \rightarrow H$  yolu seçilir. H düğümü aynı zamanda hedef durumu olduğu için algoritma burada sona erecektir. Böylece Başlangıç durumu A düğümünden hedef durumu H düğümüne geçiş yolu  $A \rightarrow C \rightarrow G \rightarrow H$  olur.

A\* arama algoritmasının avantajları:

- A\* arama algoritması diğer arama algoritmalarıyla karşılaştırıldığında en iyi arama algoritmasıdır.
- A\* arama algoritması ideal ve eksiksizdir.
- A\* arama algoritması çok karmaşık problemlerin çözümünde kullanılabilir.

# BİLGİLİ ARAMA ALGORİTMALARI

A\* arama algoritmasının dezavantajları:

- A\* arama algoritması çoğunlukla sezgisel değerlere ve yaklaşıklığa dayalı olduğu için her zaman en kısa yolu üretemeyebilir.
- A\* arama algoritmasının bazı karmaşıklık sorunları vardır.
- A\* arama algoritmasının asıl dezavantajı oluşturulan tüm düğümleri bellekte tuttuğu için büyük boyutlu bellek gereksinimidir. Bu nedenle çeşitli büyük ölçekli problemler için pratik değildir.

# BİLGİLİ ARAMA ALGORİTMALARI

## AO\* Arama

AO\* arama, bilgili bir arama algoritmasıdır ve sezgisel temelli çalışır. AO\* arama algoritması, problemi küçük parçalara ayırarak işlem yapar. Kısaca problem ayrıştırmaya dayalıdır. Bir problem, her alt problemin ayrı ayrı çözülebildiği ve her bir alt problem için bir dizi alt probleme bölünebildiği ya da ayrıştırılabildiği zaman, alt çözüm değerlendirilir ve bu alt çözümlerin kombinasyonu tam bir çözüm olacaktır.

# BİLGİLİ ARAMA ALGORİTMALARI

Yapay zekada problem azaltma, bir problemin çözümünün onun daha küçük alt problemlere ayırarak elde edilmesidir. Başka bir deyişle böl ve yönet stratejisidir. Bu alt problemlerin her biri daha sonra alt çözümünü elde etmek için çözülebilir. Bu alt çözümler daha sonra bir bütün olarak bir çözüm elde etmek için yeniden birleştirilebilir. Bu işleme Sorun Azaltma denir. Bu çözümü temsil etmek için VE-YA DA (AND-OR) grafları ya da VE YA DA ağaçları kullanılır. Bu yöntem VE-YA DA yayları olarak adlandırılan yayı üretir. Bir VE yayı herhangi bir sayıda ardıl düğüme işaret edebilir ve bir yayın bir çözüme işaret etmesi için hepsinin çözülmesi gerekir. VE-YA DA grafi çeşitli karmaşık problem çözümlerini temsil etmek için kullanılır.

# BİLGİLİ ARAMA ALGORİTMALARI

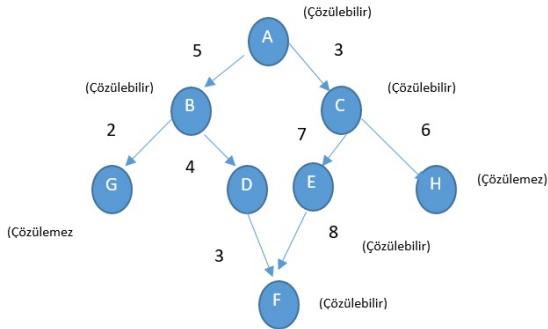


Figure 11: AO\* Arama algoritması örneği



# BİLGİLİ ARAMA ALGORİTMALARI

Şekildeki örnek arama ağacına  $AO^*$  arama algoritmasını adım adım uygularsak çözüm ağacı elde edilmiş olur.

# BİLGİLİ ARAMA ALGORİTMALARI

Adım 1: düğümler A,B,C,D,E,F ve çözülemeyen düğümler G, H'dir.  $A^*$ 'yı başlangıç düğümü olarak alınır. Öyleyse A Açık dizine yerleştirilir.

Adım 2: A düğümünün alt ögeleri B ve C çözülebilir düğümlerdir. Bu nedenle B ve C düğümleri Açık dizisine alınırken, A düğümü Kapalı dizisine yerleştirilir.

Adım 3: Bu durumda B ve C düğümleri işlenecektir. B düğümünün alt düğümleri G ve D ile C düğümünün alt düğümleri E ve H düğümleri başlangıçta Açık dizine alınır. B ve C düğümleri Kapalı dizine taşınır.

Adım 4: G ve H düğümleri çözülemez olduğu için doğrudan Kapalı dizine alınır. D ve E düğümleri üzerinden işlem yapılır.

# BİLGİLİ ARAMA ALGORİTMALARI

Adım 5: Böylece hedef durum olan F düğümüne ulaşılmıştır. F düğümü kapalı dizine alınır.

Adım 6: Kapalı dizisindeki çözülebilir ağacında seçilenler alınarak AO\* grafi elde edilir. Başarı sağlanarak algorithmadan çıkılır.

AO\* arama algoritmasının avantajları:

- İdeal bir algoritmadır.
- Düğümlerin sırasına göre hareket ederse hem YA DA hem de VE grafları için kullanılabilir.

AO\* arama algoritmasının dezavantajları:

- Bazen çözülemeyen düğümler için en uygun yolu bulamaz.
- Karmaşıklığı diğer algoritmalarından daha fazladır.

## Araç-Sonuç Analizi

Araç-sonuç analizi, problem çözmede böl ve yönet yöntemlerinin bir başka versiyonudur. Araç-sonuç analizi, Newell ve Simon'ın problem çözme konusundaki çalışmalarından ortaya çıkan bir problem çözme stratejisidir. Araç-sonuç analizinde, ilk problem durumu ile çözüm/hedef durum arasında yer alan engeller dikkate alınarak problem çözülür. Bu engellerim ortadan kaldırılması ve yinelemeli olarak bu engellerin ortadan kaldırılmasının önündeki engellerin kaldırılması ile ulaşılabacak daha basit alt hedefler olarak tanımlanabilir.

# BİLGİLİ ARAMA ALGORİTMALARI

Tüm alt hedeflere ulaşıldığında ya da tüm engeller ortadan kalktığında asıl ilgi amacına ulaşmış demektir. Alt hedefler bu ana hedefi çözme ihtiyacı tarafından çağrıldığından, araç-sonuç analizi, problem çözmeye rehberlik etmek için uzun vadeli hedefin her zaman akılda tutulduğu bir arama stratejisi olarak görülebilir.

# BİLGİLİ ARAMA ALGORİTMALARI

Tepe tırmanışında böl ve yönetin tamamen özyinelemeli olmasıdır. Böylece çözülen alt problemler her zaman aynı türedendir. Araç-sonuç analizi daha esnektir ve daha az özyinelemelidir. Çünkü çözüm için tanımlanan alt problemlerin hepsinin aynı tipte olması gerekmez. Geçerli durum GEÇERLİ ve Hedef durum HEDEF olarak alınarak Araç-sonuç analizi algoritmasında gösterilmektedir.



## Tepe Tırmanma

Tepe Tırmanma arama algoritması yapay zeka kapsamında optimizasyon problemleri için işe koşulan, basit bir sezgisel arama çözümüdür. Büyük bir girdi seti ve iyi bir amaç fonksiyonu (matematiksel model) verildiğinde, probleme yönelik uygun (optimum) bir çözüm bulmaya çalışır. Tepe tırmanma algoritması, dağın zirvesini ya da probleme en iyi çözümü bulmak için sürekli artan yükseklik/değer yönünde hareket eden yerel bir arama algoritmasıdır.

# BİLGİLİ ARAMA ALGORİTMALARI

Yerel arama algoritmaları, aday çözümler arasından kriterleri maksimize eden bir çözüm bulmaya çalıştığı karmaşık optimizasyon problemlerinde kullanılır. Aday çözüm, bir problemin tüm işlevsek bölgesindeki tüm olası çözümlerin kümesi olarak kabul edilir. Tepe tırmanma algoritması da yalnızca yakın komşu durumuna baktığı için aynı zamanda açgözlü yerel arama olarak da adlandırılır ve genel olarak aşağıdaki özellikleri verilebilir:

# BİLGİLİ ARAMA ALGORİTMALARI

- Bir tepe tırmanma algoritması düğümü durum ve değer olmak üzere iki bileşene sahiptir.
- Tepe Tırmanışı çoğunlukla iyi bir sezgisel değer mevcut olduğunda kullanılır.
- Bu algoritmada, yalnızca tek bir geçerli durumu koruduğu için arama ağacını ya da grafiğini korumak ve işlemek gerekmez.

# BİLGİLİ ARAMA ALGORİTMALARI

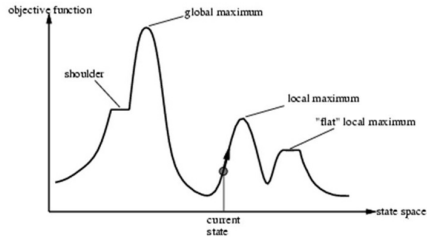


Figure 12: Tepe Tırmanma örneği

# BİLGİLİ ARAMA ALGORİTMALARI

Tepe Tırmanmanın Durum-uzay Şeması Şekilde durum-uzay görünüşü, çeşitli algoritma durumları ve amaç fonksiyonu/maliyet arasındaki bir grafiği gösteren tepe tırmanma algoritmasının grafiksel bir temsilidir. Y ekseninde amaç fonksiyonu ya da maliyet fonksiyonu olabilir ve x ekseninde durum uzayı olur. Y eksenindeki fonksiyon maliyet ise, aramanın amacı küresel minimum ve yerel minimumu bulmaktır. Y ekseninin işlevi amaç işlevi ise, aramanın amacı küresel maksimum ve yerel maksimumu bulmaktır.

# BİLGİLİ ARAMA ALGORİTMALARI

**Yerel Maksimum:** Diyagramda da görüleceği üzere yerel maksimum (lokal maksimum), komşu durumlardan biraz daha iyi olan ancak her zaman en yüksek durumdan daha düşük olan durumdur. **Küresel Maksimum:** Küresel maksimum (global maksimum), durum uzayının en yüksek durumudur ve maliyet fonksiyonunun en yüksek değerine sahiptir. **Geçerli Durum:** Aktif bir etmenin varlığını içeren durumdur. **Düz yerel maksimum:** Komşu, hepsinin aynı değere sahip olduğunu belirtiyorsa, düz bir boşlukla temsil edilebilirler.

# BİLGİLİ ARAMA ALGORİTMALARI

Omuz bölgesi: Kenarı yukarı doğru olan bir bölgedir ve tepe tırmanma algoritmalarında da problem den biri olarak kabul edilir. Algoritmanın uygulanmasında bu durumlarda oluşabilecek bazı sıkıntılarda vardır ve bunların çözülmesi içinde bazı uygulamalar yapılabilir. Örneğin: Yerel maksimum durumunda, tepe tırmanma algoritması her zaman en iyi olan bir durum bulur ancak bir yerel maksimum biter. Çünkü komşu durumlar geçerli duruma göre daha kötü değerlere sahiptir ve tepe tırmanma algoritmaları açgözlü bir yaklaşım izlediği için sona erme eğilimindedir.

# BİLGİLİ ARAMA ALGORİTMALARI

Bu tür sıkıntıların üstesinden gelmek için, algoritmanın ziyaret ettiği her durumun değerlerini hatırlaması gereken durumlarda geri izleme tekniği kullanılabilir.

Plato ya da omuz bölgesi durumunda, tüm komşular aynı değeri içeriyor gibi görünüyor ve bu da uygun bir yön seçmeyi zorlaştırır. Bu tür sıkıntıların üstesinden gelmek için algoritma, geçerli durumdan uzakta rastgele bir durum seçtiği rastgele tanımlanan bir süreç izleyebilir. Bir çözüm ayrıca bir etmenin plato olmayan bir bölgeye düşmesine neden olabilir.



## Tepe Tırmanmanın Türleri

### 1 Basit Tepe Tırmanışı

Basit tepe tırmanma tek bir seferde yalnızca komşu düğüm durumunu dikkate alıp değerlendirir ve mevcut çözümü optimize eden ilk komşuyu seçer. Sadece bir ardıl durumu olup olmadığına bakar ve eğer geçerli durumdan daha iyi bir komşu bulursa, o zaman o komşuya geçiş sağlanır. Bu algoritma genel olarak şu özelliklere sahiptir.

# BİLGİLİ ARAMA ALGORİTMALARI

- Daha az zaman alır.
- Daha az ideal çözümdür.
- Çözüm garanti değildir.

## ② En dik-Yükseklik Tepesi Tırmanma

En dik-yükseliş algoritması, geçerli durumun bütün komşu düğümlerini inceler ve hedef duruma en yakın olan bir komşu düğümü seçer. Bu algoritma, birden fazla komşu ararken daha fazla zaman harcar.

## ③ Rastgele tepe tırmanma

Rastgele tepe tırmanma, komşu düğümleri rastgele, komşu düğümleri rastgele seçtiği ve rastgele hareket etmeye ya da başka bir tane seçmeye karar verdiği için diğer iki algoritmadan farklıdır. Bu algoritma değer iki algoritmaya göre çok daha az kullanılmaktadır. Rastgele tepe tırmanma, hareket etmeden önce tüm komşularını incelemeyiz. Bunun yerine, bu arama algoritması rastgele bir komşu düğümü seçer ve onu mevcut bir durum olarak değerlendirir ya da başka bir durumu inceler. Bu algoritmanın başlıca özellikleri aşağıda verilmiştir.

# BİLGİLİ ARAMA ALGORİTMALARI

- Herhangi bir yöne bakmaksızın maliyet fonksiyonunu azaltabilen durumları bulmaya devam ederken açgözlü bir yaklaşım kullanılır.
- Beklenen çözümlerin üretilmesinde ve test algoritmasında bir değişken olarak kabul edilir. Önce çözümler üretmeye çalışır ve bunun beklenip beklenmediğini değerlendirir. Beklendiği gibi bulunursa durur, yoksa yine çözüm bulmaya çalışır.
- Önceki alanı hatırlamak için bir hafıza içermediği için geri izleme yaklaşımı gerçekleştirmez.

# BİLGİLİ ARAMA ALGORİTMALARI

**Oluştur ve Test Et** Oluştur ve test et arama algoritması tüm yaklaşımların en basitidir. Oluştur ve test et arama algoritması sistematik olarak yapıldığında ve bir çözüm bulunduğunda, uygun çözümü bulmayı garanti eden geri izlemeli derinlemesine aramaya dayalı sezgisel bir arama tekniğidir. Bu teknikte tüm çözümler üretilir ve en iyi çözüm için test edilir. Oluşturulan tüm olası çözümlere karşı en iyi çözümün kontrol edilmesini sağlar.

# BİLGİLİ ARAMA ALGORİTMALARI

Tüm çözümler oluştur ve test et algoritmasında sistematik olarak üretildiğinden, değerlendirme buluşsal fonksiyon tarafından gerçekleştirilir. Ancak sonuca götürme olasılığı en düşük olan bazı yollar dikkate alınmaz. Sezgisel işlev bunu tüm alternatifleri sıralayarak yapar ve bunu yaparken genellikle etkilidir. Sistematik oluştur ve test et, karmaşık sorunları çözerken etkisiz olabilir. Ancak, arama alanını azaltmak için oluştur ve test aramasını diğer tekniklerle birleştirerek karmaşık durumları da iyileştirebilecek bazı teknikler vardır.

# BİLGİSİZ VE BİLGİLİ ARAMA ALGORİTMALARININ KARŞILAŞTIRILMASI

Bilgisiz arama kör arama olarak da bilinirken, bilgili arama da sezgisel arama olarak adlandırılır. Bilgisiz arama fazla bilgi gerektirmezken, bilgili arama, alana özgü ayrıntılar gerektirir. Bilgisiz arama ile karşılaştırıldığında, bilgili arama teknikleri daha verimlidir ve bilgisiz arama tekniklerinin zaman karmaşıklığı daha fazladır. Bilgili arama, problemi kör armadan daha iyi ele alır. Birçok problem, graflarda yol bulma problemi olarak özetlenebilir. Bilgili ve bilgisiz arama algoritmalarının temel özellikleri aşağıdaki gibi özetlenebilir.



# BİLGİSİZ VE BİLGİLİ ARAMA ALGORİTMALARININ KARŞILAŞTIRILMASI

- Genişlemesine ve derinlemesine arama algoritmaları, grafin ötesinde herhangi bir ek bilgi olmadan graflardaki yolları bulabilir.
- A\* arama algoritması bir düğümden hedefe giden maliyet tahmin eden bir sezgisel işlevi kullanabilir. Bu tahmin gerçek maliyeti olduğundan az tahmin ederse, A\* algoritmasının önce en düşük maliyetli yolu bulması garanti edilir.

# BİLGİSİZ VE BİLGİLİ ARAMA ALGORİTMALARININ KARŞILAŞTIRILMASI

- Yinelemeli derinlemesine ve derinlemesine öncelikli dal ve sınır aramaları, birden çok yolu depolayan  $A^*$  gibi yöntemlerden daha az belleğe ihtiyaç duyar ve en düşük maliyetli yolları bulmak için kullanılabilir.
- Graflar küçük olduğunda her bir düğümden hedefe giden en düşük maliyetli yolun gerçek maliyetini kaydetmek için dinamik programlama kullanılabilir, bu da ideal bir yoldaki sonraki kolu bulmak için kullanılabilir.