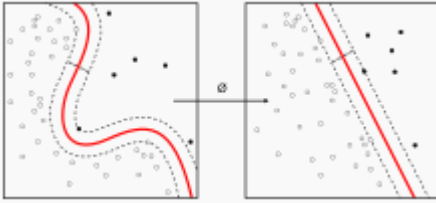


**Yapay sinir ağı** (YSA), insan beyninin bilgi işleme tekniğinden esinlenerek geliştirilmiş bir bilgi işlem teknolojisidir. YSA ile basit biyolojik sinir sisteminin çalışma şekli taklit edilir. Taklit edilen sinir hücreleri nöronlar içerirler ve bu nöronlar çeşitli şekillerde birbirlerine bağlanarak ağı oluştururlar. Bu ağılar öğrenme, hafızaya alma ve veriler arasındaki ilişkiyi ortaya çıkarma kapasitesine sahiptirler. Diğer bir ifadeyle, YSA'lar, normalde bir insanın düşünme ve gözlemlemeye yönelik doğal yeteneklerini gerektiren problemlere çözüm üretmektedir. Bir insanın, düşünme ve gözleme yeteneklerini gerektiren problemlere yönelik çözümler üretebilmesinin temel sebebi ise insan beyninin ve dolayısıyla insanın sahip olduğu yaşayarak veya deneyerek öğrenme yeteneğidir.

Biyolojik sistemlerde öğrenme, nöronlar arasındaki sinaptik (*synaptic*) bağlantıların ayarlanması ile olur. Yani, insanlar doğumlarından itibaren bir *yaşayarak öğrenme* süreci içerisine girerler. Bu süreç içinde beyin sürekli bir gelişme göstermektedir. Yaşayıp tecrübe ettikçe sinaptik bağlantılar ayarlanır ve hatta yeni bağlantılar oluşur. Bu sayede öğrenme gerçekleşir. Bu durum YSA için de geçerlidir. Öğrenme, eğitme yoluyla örnekler kullanarak olur; başka bir deyişle, gerçekleşme girdi/çıkış verilerinin işlenmesiyle, yani eğitme algoritmasının bu verileri kullanarak bağlantı ağırlıklarını (*weights of the synapses*) bir yakınsama sağlanana kadar, tekrar tekrar ayarlamasıyla olur.

## **Makine öğrenmesi ve veri madenciliği**



**Problemler**[göster]

**Gözetimli öğrenme**

[göster]

**Kümeleme**[göster]

**Boyutluluk indirgemesi**[göster]

**Yapılandırılmış tahmin**[göster]

**Anomali tespiti**[göster]

**Sinir ağıları**[göster]

**Pekiştirmeli öğrenme**[göster]

**Teori**[göster]

**Konferanslar ve dergiler**[göster]

G

T

D

YSA'lar, ağırlıklandırılmış şekilde birbirlerine bağlanmış birçok işlem biriminden (nöronlar) oluşan matematikselsistemlerdir. Bir işlem birimi, aslında sık sık transfer fonksiyonu olarak anılan bir denklemdir. Bu işlem birimi, diğer nöronlardan sinyalleri alır; bunları birleştirir, dönüştürür ve sayısal bir sonuç ortaya çıkarır. Genelde, işlem birimleri kabaca gerçek nöronlara karşılık gelirler ve bir ağ içinde birbirlerine bağlanırlar; bu yapı da sinir ağlarını oluşturmaktadır.

Sinirsel (*neural*) hesaplamaların merkezinde dağıtılmış, adaptif ve doğrusal olmayan işlem kavramları vardır. YSA'lar, geleneksel işlemcilerden farklı şekilde işlem yapmaktadırlar. Geleneksel işlemcilerde, tek bir merkezi işlem birimi her hareketi sırasıyla gerçekleştirir. YSA'lar ise her biri büyük bir problemin bir parçası ile ilgilenen, çok sayıda basit işlem birimlerinden oluşmaktadır. En basit şekilde, bir işlem birimi, bir girdiyi bir ağırlık kümesi ile ağırlıklandırır, doğrusal olmayan bir şekilde dönüşümünü sağlar ve bir çıktı değeri oluşturur. İlk bakışta, işlem birimlerinin çalışma şekli yanıltıcı şekilde basittir. Sinirsel hesaplamaların gücü, toplam işlem yükünü paylaşan işlem birimlerinin birbirleri arasındaki yoğun bağlantı yapısından gelmektedir. Bu sistemlerde geri yayılım metoduyla daha sağlıklı öğrenme sağlanmaktadır.

Çoğu YSA'da, benzer karakteristiğe sahip nöronlar tabakalar halinde yapılandırılırlar ve transfer fonksiyonları eş zamanlı olarak çalıştırılırlar. Hemen hemen tüm ağlar, veri alan nöronlara ve çıktı üreten nöronlara sahiptirler.

YSA'nın ana öğesi olan matematiksel fonksiyon, ağın mimarisi tarafından şekillendirilir. Daha açık bir şekilde ifade etmek gerekirse, fonksiyonun temel yapısını ağırlıkların büyüklüğü ve işlem elemanlarının işlem şekli belirler. YSA'ların davranışları, yani girdi veriyi çıktı veriyeye nasıl ilişkilendirdikleri, ilk olarak nöronların transfer fonksiyonlarından, nasıl birbirlerine bağlandıklarından ve bu bağlantıların ağırlıklarından etkilenir.

Yapay sinir ağlarının üstünlüklerinin yanı sıra bazı sakıncaları da vardır. Bu sakıncalar şu şekilde listelenebilir:

- Sistem içerisinde ne olduğu bilinemez.
- Bazı ağlar hariç kararlılık analizleri yapılamaz.
- Farklı sistemlere uygulanması zor olabilir.

**Makine öğrenimi**, bilgisayarların algılayıcı verisi ya da veritabanları gibi veri türlerine dayalı öğrenimini olanaklı kılan algoritmaların tasarımı ve geliştirme süreçlerini konu edinen bir bilim dalıdır. Makine öğrenimi araştırmalarının odaklandığı konu bilgisayarlara karmaşık örüntüleri algılama ve veriye dayalı akılcı kararlar verebilme becerisi kazandırmaktır. Bu, makine öğreniminin istatistik, olasılık kuramı, veri madenciliği, örüntü tanıma, yapay zekâ, uyarlamalı denetim ve kuramsal bilgisayar bilimi gibi alanlarla yakından ilintili olduğunu göstermektedir.

Makine öğreniminin başlıca uygulamaları makine algılaması, bilgisayarlı görme, doğal dil işleme, sözdizimsel örüntü tanıma, arama motorları, tıbbi tanı, biyoinformatik, beyin-makine arayüzleri ve kiminformatik, kredi kartı dolandırıcılığı denetimi, borsa çözümlemesi, DNA dizilerinin sınıflandırılması, konuşma ve elyazısı tanıma, bilgisayarlı görmede nesne tanıma, oyun oynama, yazılım mühendisliği, uyarlamalı web siteleri ve robot gezisidir.

Makine öğrenimi algoritmaları hedeflenen sonuca göre birkaç sınıfa ayrılabilir:<sup>[1]</sup>

- **Gözetimli öğrenme** - Girdileri hedef çıktılara eşleyen bir işlev üretir.
- **Gözetimsiz öğrenme** - Bir girdi kümesi modeller.
- **Pekiştirmeli öğrenme** - Dünya algısına dayalı bir öğrenme biçimi. Her eylem ortamda bir etki oluşturmakta ve ortam, öğrenme algoritmasına yol gösteren ödülleri biçiminde dönütler vermektedir.
- **Yarı gözetimli öğrenme** - Uygun işlev ya da sınıflandırıcılar oluşturmak için etiketli ve etiketsiz örnekleri birlikte ele alır.
- **Öğrenmeyi öğrenme** - Önceki deneyimlerden yararlanır.

Makine öğrenimi algoritmaları ve bunların başarımına ilişkin berimsel çözümleme berimsel öğrenme kuramı olarak adlandırılan bir kuramsal bilgisayar bilimi dalıdır. Deney kümelerinin sonlu oluşu ve geleceğin tam olarak kestirilememesi nedeniyle öğrenme kuramı söz konusu algoritmaların başarımına ilişkin mutlak güvence verememektedir. Bunun yerine, başarımın olasılıksal sınırları öngörülmeğe çalışılmaktadır.

Berimsel öğrenme kuramcıları başarım sınırlarının yanı sıra öğrenmenin zaman karmaşıklığı ve uygulanabilirliği konusunda da çalışmaktadırlar. Berimsel öğrenme kuramında bir berimin uygulanabilir olması için polinomsal zamanda çalışması gerekmektedir. Zaman karmaşıklığı sonuçları iki öbeğe ayrılabilir. Olumlu sonuçlar belirli bir işlev sınıfının polinomsal zamanda öğrenilebileceğini gösterirken olumsuz sonuçlar bu olgunun tam karşıtını ifade etmektedir.

Makine öğrenimi kuramı ve istatistik temelde farklı kavramlar olsalar da birbiriyle yakından ilintilidir.

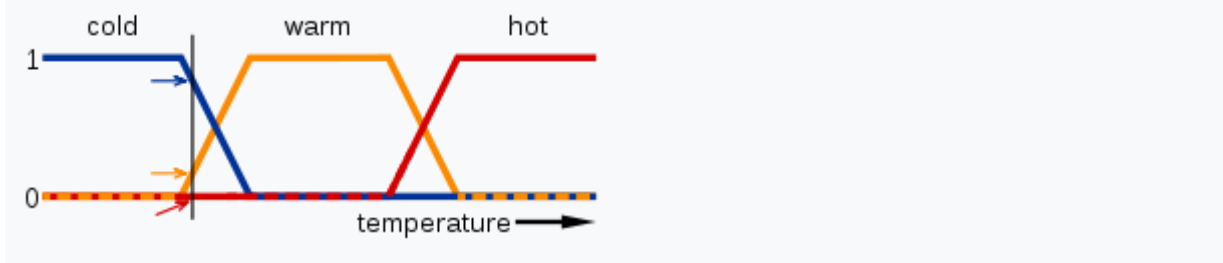
**Derin öğrenme** (aynı zamanda **derin yapılandırılmış öğrenme**, **hiyerarşik öğrenme** ya da **derin makine öğrenmesi**) bir veya daha fazla gizli katman içeren yapay sinir ağları ve benzeri makine öğrenme algoritmalarını kapsayan çalışma alanıdır.

derin öğrenmeyi açıklayacak olursak; yol rengi gri bir renktir çimen rengi yeşildir böyle bakıldığında çimenli alan yoğun bir yeşil bir alan içerir, bu sayede oranın çimenli bölge olduğunu algılar. Nvidia'nın sürücüsüz araç projesi derin öğrenme mantığını anlatan en iyi örnektir arabada bulunan algılayıcılar sayesinde çevrenin üç boyutunu çıkararak yeşil alanların çimenli alan olduğunu algılaması sağlanır alt kısımdaki videoda Nvidia'nın sürücüsüz araç projesinde ilk olarak araba bütün engellere çarpıyor ilk olarak bu veriler kaydediliyor daha sonra araç geçtiği zaman hiç bir engele çarpmadan ilerlediğini görüyoruz ve bu sayede öğrenme gerçekleştiriliyor bunların hepsinin altında derin öğrenme yatmaktadır.

Derin öğrenme ile yapılmış çoğu uygulamayı aslında hayatımızda farkında olmadan kullanmaktayız. Bunlara örnek verecek olursak Apple Siri, Google Bunu mu demek istediniz? , Apple FaceID, Huawei Yapay zeka destekli telefonları. Günümüzde cebimize kadar giren bu kavramın önemi hiçte azımsanmayacak kadar çoktur.

**Bulanık mantık**, **bulanık eseme** ya da puslu mantık, 1961 yılında Lütfü Aliasker Zade'nin yayınladığı bir makalenin sonucu oluşmuş bir mantık yapısıdır.

Bulanık mantığın temeli bulanık küme ve alt kümelere dayanır. Klasik yaklaşımda bir varlık ya kümenin elemanıdır ya da değildir. Matematiksel olarak ifade edildiğinde varlık küme ile olan üyelik ilişkisi bakımından kümenin elemanı olduğunda "1", kümenin elemanı olmadığı zaman "0" değerini alır. Bulanık mantık klasik küme gösteriminin genişletilmesidir. Bulanık varlık kümesinde her bir varlığın üyelik derecesi vardır. Varlıkların üyelik derecesi, (0, 1) aralığında herhangi bir değer olabilir ve üyelik fonksiyonu  $M(x)$  ile gösterilir .



Örnek olarak normal oda sıcaklığını 23 derece olarak kabul edersek klasik küme kuramına göre 23 derecenin üzerindeki sıcaklık derecelerini sıcak olarak kabul ederiz ve bu dereceler sıcak kümesindeki üyelik dereceleri "1" olur. 23 altındaki sıcaklık dereceleri ise soğuktur ve sıcak kümesindeki üyelik dereceleri "0" olur. Soğuk kümesini temel aldığımızda bu değerler tersine döner. Bulanık küme yaklaşımında üyelik değerleri [0,1] aralığında değerler almaktadır. Örneğin 14 derecelik sıcaklık için üyelik derecesi "0", 23 sıcaklık derecesi için üyelik değeri "0,25" olabilir.

Klasik kümelerin aksine bulanık kümelerde elemanların üyelik dereceleri [0, 1] aralığında sonsuz sayıda değişebilir. Bunlar üyeliğin derecelerinin devamlı ve aralıksız bütünüyle bir kümedir. Keskin kümelerdeki soğuk-sıcak, hızlı-yavaş, aydınlık-karanlık gibi ikili değişkenler, bulanık mantıkta biraz soğuk, biraz sıcak, biraz karanlık gibi esnek niteleyicilerle yumuşatılarak gerçek dünyaya benzetilir. En önemli fark, böyle bir çatıda bilginin kaynağındaki küme üyeliğinin kesin tanımlanmış önkoşullarının olmayışı ve daha çok sorunlarla rastgele değişkenlerin hazır bulunmasındadır.

Bir şeyin varlığı kendisine ait bir isimle doğar. Evrendekilerin tamamı hem (ya) tek (1) hem de (ya da) sonsuz eksi tektir (sonsuz -1).

Klasik mantık ile bulanık mantık arasındaki temel farklılıklar :

Klasik Mantık	Bulanık Mantık
A veya A Değil	A ve A Değil
Kesin	Kısmi
Hepsi veya Hiçbiri Belirli Derecelerde	
0 veya 1	0 ve 1 Arasında Süreklilik
İkili Birimler	Bulanık Birimler

**Bulanık mantık** bir yapay zekâ uygulaması oluşturma prensibidir. Bulanık mantıkta temel olan bir sonuca varmaktır. Normal bir programın yapısı:

- Temel girdiler → Program → Sabit bir sonuç şeklindedir. Oysaki bir bulanık mantık uygulaması:
- Sayısı belli olmayan veri yığını → Program → Girdilere ve varsayıma göre değişken bir veya birden fazla sonuç şeklindedir. Bir bulanık mantık uygulamasındaki sonuç, aynı girdiler olsa bile değişik bir sonuç döndürebilir ve bir öbek halinde veriyi alabilir. Bulanık

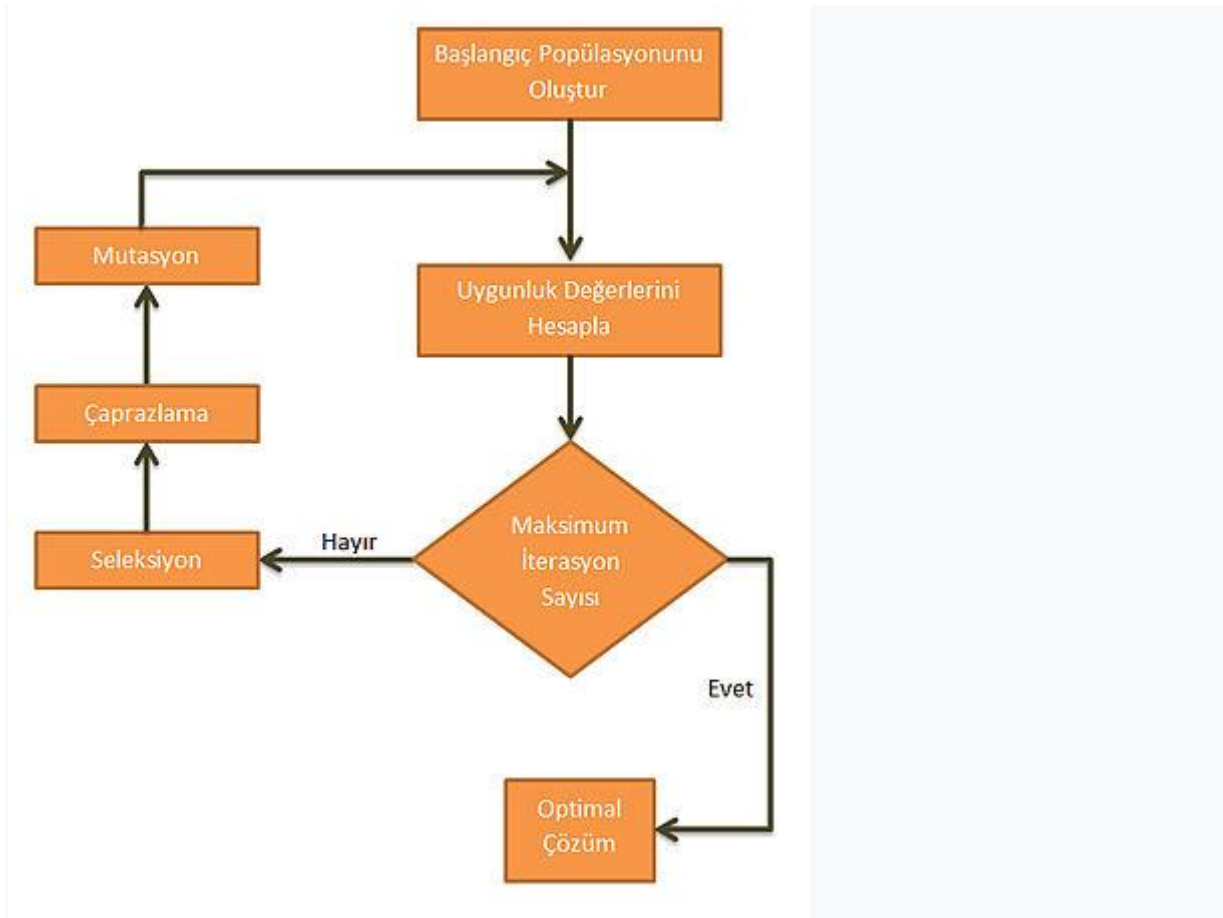
mantıktaki özellik bunun haricinde verilen verilerin örnekleme mantığı ile alınması ve tümü simgelediği varsayımı yapılması ve buna göre bir olasılık değerinin elde edilmesidir.

**Genetik algoritmalar**, doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve eniyileme yöntemidir. Karmaşık çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre bütünsel en iyi çözümü arar<sup>[1]</sup>.

Genetik algoritmaların temel ilkeleri ilk kez Michigan Üniversitesi'nde John Holland tarafından ortaya atılmıştır. Holland 1975 yılında yaptığı çalışmaları "Adaptation in Natural and Artificial Systems" adlı kitabında bir araya getirmiştir. İlk olarak Holland evrim yasalarını genetik algoritmalar içinde eniyileme problemleri için kullanmıştır.

Genetik algoritmalar problemlere tek bir çözüm üretmek yerine farklı çözümlerden oluşan bir çözüm kümesi üretir. Böylelikle, arama uzayında aynı anda birçok nokta değerlendirilmekte ve sonuçta bütünsel çözüme ulaşma olasılığı yükselmektedir. Çözüm kümesindeki çözümler birbirinden tamamen bağımsızdır. Her biri çok boyutlu uzay üzerinde bir vektördür.

Genetik algoritmalar problemlerin çözümü için evrimsel süreci bilgisayar ortamında taklit ederler. Diğer eniyileme yöntemlerinde olduğu gibi çözüm için tek bir yapının geliştirilmesi yerine, böyle yapılardan meydana gelen bir küme oluştururlar. Problem için olası pek çok çözümü temsil eden bu küme genetik algoritma terminolojisinde nüfus adını alır. Nüfuslar vektör, kromozom veya birey adı verilen sayı dizilerinden oluşur. Birey içindeki her bir elemana gen adı verilir. Nüfustaki bireyler evrimsel süreç içinde genetik algoritma işlemcileri tarafından belirlenirler.



Genetik Algoritmanın Genel Akış Şeması

Problemin bireyler içindeki gösterimi problemden probleme deęişiklik gösterir. Genetik algoritmaların problemin çözümündeki başarısına karar vermedeki en önemli faktör, problemin çözümünü temsil eden bireylerin gösterimidir. Nüfus içindeki her bireyin problem için çözüm olup olmayacağına karar veren bir uygunluk fonksiyonu vardır. Uygunluk fonksiyonundan dönen değere göre yüksek değere sahip olan bireylere, nüfustaki diğer bireyler ile çoęalmaları için fırsat verilir. Bu bireyler çaprazlama işlemi sonunda çocuk adı verilen yeni bireyler üretirler. Çocuk kendisini meydana getiren ebeveynlerin (anne, baba) özelliklerini taşır. Yeni bireyler üretilirken düşük uygunluk değerine sahip bireyler daha az seçileceğinden bu bireyler bir süre sonra nüfus dışında bırakılırlar. Yeni nüfus, bir önceki nüfusta yer alan uygunluğu yüksek bireylerin bir araya gelip çoęalmalarıyla oluşur. Aynı zamanda bu nüfus önceki nüfusun uygunluğu yüksek bireylerinin sahip olduęu özelliklerin büyük bir kısmını içerir. Böylelikle, pek çok nesil aracılığıyla iyi özellikler nüfus içerisinde yayılırlar ve genetik işlemler aracılığıyla da diğer iyi özelliklerle birleşirler. Uygunluk değeri yüksek olan ne kadar çok birey bir araya gelip, yeni bireyler oluşturursa arama uzayı içerisinde o kadar iyi bir çalışma alanı elde edilir. Probleme ait en iyi çözümün bulunabilmesi için;

- Bireylerin gösterimi doğru bir şekilde yapılmalı,
- Uygunluk fonksiyonu etkin bir şekilde oluşturulmalı,
- Doğru genetik işlemciler seçilmeli.

Bu durumda çözüm kümesi problem için bir noktada birleşecektir. Genetik algoritmalar, diğer eniyileme yöntemleri kullanılırken büyük zorluklarla karşılaşılan, oldukça büyük arama uzayına sahip problemlerin çözümünde başarı göstermektedir. Bir problemin bütünsel en iyi çözümünü bulmak için garanti vermezler. Ancak problemlere makul bir süre içinde, kabul edilebilir, iyi çözümler bulurlar. Genetik algoritmaların asıl amacı, hiçbir çözüm teknięi bulunmayan problemlere çözüm aramaktır. Kendilerine has çözüm teknikleri olan özel problemlerin çözümü için mutlak sonucun hızı ve kesinlięi açısından genetik algoritmalar kullanılmazlar. Genetik algoritmalar ancak;

- Arama uzayının büyük ve karmaşık olduęu,
- Mevcut bilgiyle sınırlı arama uzayında çözümün zor olduęu,
- Problemin belirli bir matematiksel modelle ifade edilemedięi,
- Geleneksel eniyileme yöntemlerinden istenen sonucun alınmadıęı alanlarda etkili ve kullanışlıdır.

Genetik algoritmalar; Parametre ve sistem tanılama, kontrol sistemleri, robot uygulamaları, görüntü ve ses tanıma, mühendislik tasarımları, planlama, yapay zeka uygulamaları, uzman sistemler, fonksiyon ve kombinasyonel eniyileme problemleri aę tasarım problemleri, yol bulma problemleri, çizelgeleme problemleri, sosyal ve ekonomik planlama problemleri için diğer eniyileme yöntemlerinin yanında başarılı sonuçlar vermektedir.

#### Diğer yöntemlerden farkı

1. Genetik algoritmalar problemlerin çözümünü parametrelerin değerleriyle değil, kodlarıyla arar. Parametreler kodlanabildięi sürece çözüm üretilebilir. Bu sebeple genetik algoritmalar ne yaptıęı konusunda bilgi içermez, nasıl yaptıęını bilir.
2. Algoritmalar aramaya tek bir noktadan değil, noktalar kümesinden başlar. Bu nedenle çoęunlukla yerel en iyi çözümde sıkışıp kalmazlar. Ancak bazı durumlarda genetik algoritmalar yerel en iyi çözüme, genel en iyi çözümden daha çabuk ulaşırlar ve algoritma bu noktada sonuçlanır. Bu durumla karşılaşıldığında, bu durumu önlemek

için (genel en iyi çözüme ulaşabilmek için) ya arama uzayındaki çeşitlilik artırılır ya da uygunluk fonksiyonu her üreme aşamasında değiştirilir.

3. Genetik algoritmalar türev yerine uygunluk fonksiyonunun değerini kullanır. Bu değer kullanılması ayrıca yardımcı bir bilginin kullanılmasını gerektirmez.
4. Genetik algoritmalar gerekirci kuralları değil olasılıksal kuralları kullanır.