

## Tema 1

- **Inteligencia Artificial:**

- “Es la ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes”. John McCarthy (1956).

- **Sistemas Inteligentes:**

- “Un **sistema inteligente** es un programa de computación que reúne características y comportamientos asimilables al de la inteligencia humana o animal.”

- **Test de Turing.**

- Prueba o test de Turing, procedimiento desarrollado por Alan Turing para identificar la existencia de inteligencia en una máquina y comprobar que una máquina puede llegar a pensar al igual que una persona.
- Este test fue expuesto en 1950 en un artículo (Computing machinery and intelligence) para la revista Mind que sigue siendo hoy día una de las puntas de lanza de los defensores de la Inteligencia Artificial.
- Está fundamentado en la hipótesis positivista de que, si una máquina se comporta en todos los aspectos de forma inteligente, entonces debe ser inteligente.

- Básicamente la prueba consiste en un desafío en el cual la máquina debe hacerse pasar por humana en una conversación con un hombre a través de una comunicación de texto en modo chat.
- Al sujeto no se le avisa si está hablando con una máquina o una persona de modo que si el sujeto es incapaz de determinar si la otra parte de la comunicación es humana o máquina, entonces se considera que la máquina ha alcanzado un determinado nivel de madurez: es inteligente.

- Una de las aplicaciones de la prueba de Turing es el control de spam.
- Una forma de determinar si una máquina tiene como objetivo el envío de spam es comprobar, siguiendo la idea expuesta por el test de Turing, si es capaz de realizar una conversación de aspecto humana. Si el resultado del test es negativo, las peticiones de esta máquina serían bloqueadas.



- Un ejemplo muy utilizado es la prueba captcha, en la que un usuario cuando desea enviar un correo a un destinatario concreto debe antes reconocer una serie de imágenes, sonidos, patrones, reproducir un texto distorsionado, etc.
- Basándonos en tests que la mayoría humanos pueden resolver de forma sencilla y que los programas de computadores actuales no podrían en la mayoría de casos.

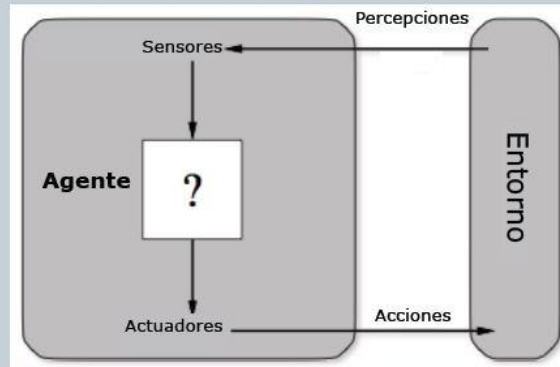


- **Sistemas que piensan como humanos.**- Estos sistemas tratan de emular el pensamiento humano; por ejemplo las redes neuronales artificiales. La automatización de actividades que vinculamos con procesos de pensamiento humano, actividades como la toma de decisiones, resolución de problemas, aprendizaje.
- **Sistemas que actúan como humanos.**- Estos sistemas tratan de actuar como humanos; es decir, imitan el comportamiento humano; por ejemplo la robótica. El estudio de cómo lograr que los computadores realicen tareas que, por el momento, los humanos hacen mejor.
- **Sistemas que piensan racionalmente.**- Es decir, con lógica (idealmente), tratan de imitar o emular el pensamiento lógico racional del ser humano; por ejemplo los sistemas expertos. El estudio de los cálculos que hacen posible percibir, razonar y actuar.
- **Sistemas que actúan racionalmente (idealmente).**- Tratan de emular de forma racional el comportamiento humano; por ejemplo los agentes inteligentes. Está relacionado con conductas inteligentes en artefactos.

- **La Inteligencia Artificial (IA) se divide en dos grupos:**

- **IA fuerte:** Construir programas que emulen el comportamiento inteligente de los humanos como el pensamiento, el aprendizaje, la visión, la resolución de problemas, la creatividad, etc. Ya que estos modos de comportamiento se pueden explicar algorítmicamente en términos de **estados** mentales.
- **IA débil:** Construir máquinas capaces de resolver problemas que requieran de inteligencia. Para ello se construyen programas que tengan un comportamiento inteligente sin importar si emula o no a la inteligencia humana.

- Un agente es cualquier cosa capaz de percibir su entorno con la ayuda de sensores y actuar sobre dicho entorno mediante actuadores.



- **Conceptos**
  - **Función del agente:** define el comportamiento del agente; proyecta una percepción dada en una acción.
  - **Programa del agente:** implementa la función del agente.
  - **Agente racional:** en cada posible secuencia de percepciones, un agente racional deberá emprender aquella acción que supuestamente maximice su medida de rendimiento, basándose en las evidencias aportadas por la secuencia de percepciones y en el conocimiento que el agente mantiene almacenado.
  - **Autonomía:** se mide por el grado en que su comportamiento viene determinado por su experiencia.
    - ✦ se requiere un balance entre la falta de autonomía y la autonomía absoluta.



Existen diferentes formas de representar problemas para resolverlos de manera automática:

○ Representaciones generales

- ✦ **Espacio de estados:** un problema se divide en un conjunto de pasos de resolución desde el inicio hasta el objetivo
- ✦ **Reducción a subproblemas:** un problema se puede descomponer en una jerarquía de subproblemas

○ Representaciones para problemas específicos

- ✦ **Resolución de juegos:** También es una aproximación específica, en la que el problema se plantea como la competición entre dos o más agentes.
- ✦ **Satisfacción de restricciones:** Esta aproximación es específica para problemas que se puedan plantear como un conjunto de variables a las que se han de asignar valores cumpliendo ciertas restricciones.

## 2.1.2. El espacio de estados.



- Podemos definir un problema por los elementos que intervienen y sus relaciones.
- En cada instante de la resolución de un problema esos elementos tendrán unas características y relaciones específicas.
- Denominaremos **Estado** a la representación de los elementos que describen el problema en un momento.
- Distinguiremos dos estado especiales el **Estado Inicial** (punto de partida) y el **Estado Final** (objetivo del problema).

- Para poder movernos entre los diferentes estados que definen el problema, necesitaremos lo que denominaremos **operadores de transformación**.
  - Un operador es una función de transformación sobre la representación de un estado que lo convierte en otro estado.
  - Los operadores definirán una **relación de accesibilidad** entre estados.
- Los estados y su relación de accesibilidad conforman lo que se denomina **espacio de estados**.

## 2.2. Búsqueda no Informada.



Los algoritmos de búsqueda no informada (también conocidos como algoritmos de búsqueda ciega) no dependen de información propia del problema a la hora de resolverlo.

Esto quiere decir que son algoritmos generales y por lo tanto se pueden aplicar en cualquier circunstancia.

Estos algoritmos se basan en la estructura del espacio de estados y determinan estrategias sistemáticas para su exploración.

Es decir, siguen una estrategia fija a la hora de visitar los nodos que representan los estados del problema.

Se trata también de algoritmos exhaustivos, de manera que pueden acabar recorriendo todos los nodos del problema para hallar la solución.

- Existen básicamente dos políticas de recorrido de un espacio de búsqueda
  - en anchura
  - en profundidad.
- Al ser algoritmos exhaustivos y sistemáticos su coste puede ser prohibitivo para la mayoría de los problemas reales, por lo tanto solo serán aplicables en problemas pequeños.
- Su ventaja es que no nos hace falta obtener ningún conocimiento adicional sobre el problema, por lo que siempre son aplicables.

## 2.3. Búsqueda heurística.

- Es evidente que los algoritmos de búsqueda no informada serán incapaces de encontrar soluciones en problemas en los que el tamaño del espacio de búsqueda sea grande.
- Todos estos algoritmos tienen un coste temporal que es una función exponencial del tamaño de la entrada, por lo tanto el tiempo para encontrar la mejor solución a un problema no es asumible en problemas reales.

## • 2.5. Búsqueda con adversario.

- Identificaremos a cada jugador como el jugador MAX y el jugador MIN.
- MAX será el jugador que inicia el juego, nos marcaremos como objetivo el encontrar el conjunto de movimientos que ha de hacer el jugador MAX para que llegue al objetivo (ganar) independientemente de lo que haga el jugador MIN.

## • 2.5. Búsqueda local.

- **Una función de calidad**, que nos medirá lo buena que es una solución y nos permitirá guiar la búsqueda. Hemos de tener en cuenta que este valor no nos indica cuanto nos falta para encontrar la solución que buscamos, ya que no representa un coste, es una manera de comparar la calidad entre las soluciones vecinas e indicarnos cual es la dirección que lleva a soluciones mejores.



## 2.5.2. Algoritmo Hill-Climbing.



- **Ascenso de colinas simple:**
  - Consiste en elegir siempre el primer operador que suponga una mejora respecto al nodo actual.
  - De manera que no exploremos todas las posibilidades accesibles, ahorrándonos el explorar cierto número de descendientes.
  - La ventaja es que es más rápido que explorar todas las posibilidades, la desventaja es que hay más probabilidad de no alcanzar las soluciones mejores.

## 2.5.2. Algoritmo Hill-Climbing.



- **Ascenso de colinas por máxima pendiente (steepest ascent hill-climbing).**
  - Esta variante expande todos los posibles descendientes de un nodo y elige el que suponga la máxima mejora respecto al nodo actual. Es decir, se selecciona el mejor movimiento (no el primero de ellos) que suponga mejora respecto al estado actual.
  - Con esta estrategia, suponemos que la mejor solución la encontraremos a través del sucesor que mayor diferencia tenga respecto a la solución actual, siguiendo una política avariciosa.
  - Como veremos más adelante esta estrategia puede ser arriesgada.
  - Algoritmo más utilizado.

## Definición

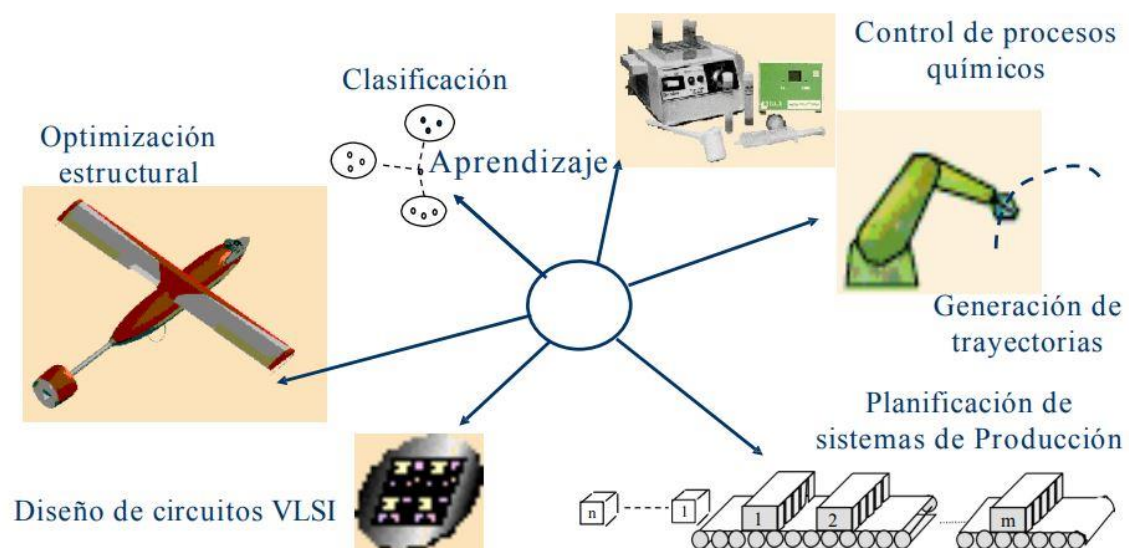
### ■ Ventajas:

- **Algoritmos de propósito general**
- **Gran éxito en la práctica**
- **Fácilmente implementables**
- **Fácilmente paralelizables**

### ■ Inconvenientes:

- **Son algoritmos aproximados, no exactos**
- **Son no determinísticos (probabilísticos)**
- **No siempre existe una base teórica establecida**

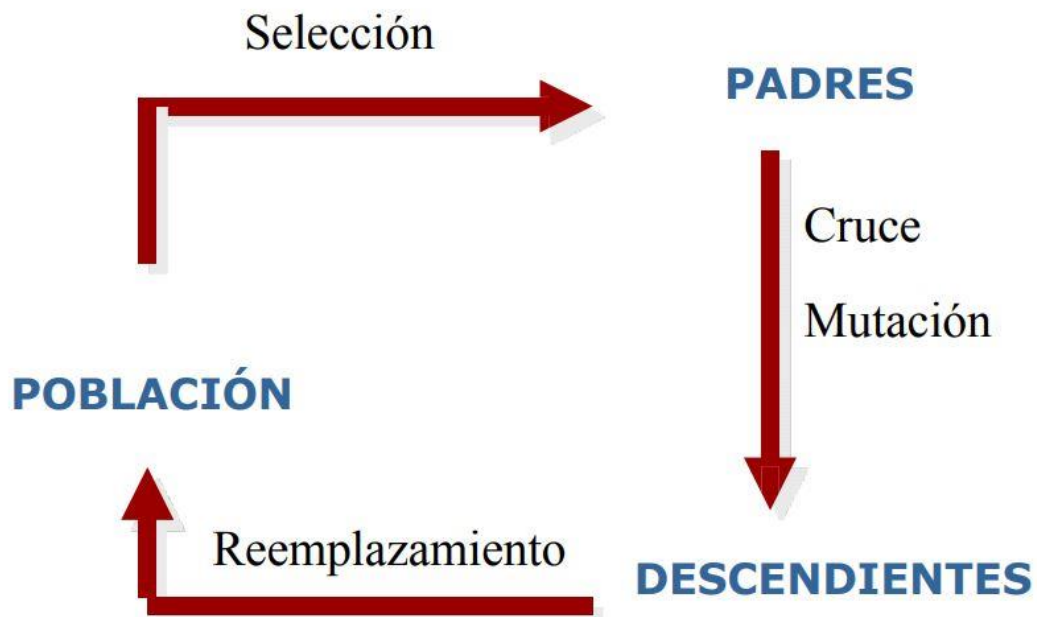
## APLICACIONES





## El ciclo de la Evolución

---



## Algoritmos meméticos

---

- La combinación de Algoritmos **Evolutivos** con Operadores Locales de Búsqueda que trabajan dentro del bucle EA se ha denominado "Algoritmos Meméticos".
- El término también se aplica a los EAs que utilizan el conocimiento específico de la instancia en los operadores.
- Se ha demostrado que los Algoritmos Meméticos son órdenes de magnitud más rápidos y precisos que los EAs en algunos problemas, y son el "estado del arte" en muchos problemas

---

- **Lamarckian**

- los rasgos adquiridos por un individuo a lo largo de su vida pueden transmitirse a su descendencia (actualización del genotipo)
- p. ej., sustituir a una persona por un vecino más adecuado

- **Baldwinian**

- los rasgos adquiridos por el individuo no pueden ser transmitidos a su descendencia (sugiere una nueva búsqueda de dirección)
- Por ejemplo, un individuo recibe la aptitud (pero no el genotipo) de un vecino en forma.

#### Tema 4 Redes Neuronales

- Una red neuronal artificial es una herramienta diseñada para emular la forma en que el cerebro humano funciona

“sistema compuesto por múltiples unidades de proceso que operan en paralelo y cuya funcionalidad depende de la estructura de la red, de la fuerza de las conexiones, y del procesamiento realizado en cada nodo” [DARPA, 98]



### 4.2.3. RED NEURONAL

- Conjunto de neuronas artificiales conectadas entre sí mediante una serie de arcos llamados conexiones
  - Estas conexiones tienen números reales asociados, llamados pesos de la conexión
  - Las neuronas generalmente se distribuyen en capas de distintos niveles, con conexiones que unen las neuronas de las distintas capas (pueden ser también de la misma capa)
- 

### 4.2.4. FUNCIÓN DE ACTIVACIÓN

- La función de activación se utiliza para limitar el rango de valores de la respuesta de la neurona.
- Generalmente los rangos de valores se limitan a  $[0,1]$  o  $[-1,1]$ , sin embargo otros rangos son posibles de acuerdo a la aplicación o problema a resolver.
- Existen diversas funciones de activación y la decisión entre una u otra dependerá nuevamente de la aplicación o problema a resolver.
- Existen funciones de activación comúnmente utilizadas y con las cuales se han obtenido resultados satisfactorios en diversas aplicaciones.

#### 4. 5 aplicaciones donde se puedan usar redes neuronales

- Entradas formadas por vectores de valores discretos o reales
- Pueden ser problemas de clasificación o regresión
- Puede producir vectores de valores
- Los usuarios finales no quieren obtener explicaciones: no producen conocimiento inteligible (caja negra)
- Los datos de entrenamiento pueden contener errores

**Elitismo**→Elitismo es una tendencia o ideología que defiende la superioridad o el dominio de un grupo selecto o élite sobre el resto de la sociedad. Esta élite puede ser determinada por razones económicas, políticas, educativas, culturales, etc. El elitismo puede manifestarse en diversos ámbitos, incluyendo la política, la economía, la educación, la cultura, entre otros, y se caracteriza por promover la exclusión de aquellos considerados "inferiores" o "no aptos" para formar parte de la élite.

Genotipo→ Elitismo es una técnica utilizada en algoritmos genéticos y evolutivos en la cual se seleccionan los individuos más aptos de una población para reproducirse y pasar sus genotipos a la siguiente generación.

Fenotipo→ El fenotipo es el conjunto de características observables de un organismo, como su apariencia, comportamiento y habilidades. El genotipo, por otro lado, es el conjunto de información genética de un organismo, es decir, el conjunto de todos sus genes. En un sistema inteligente, el fenotipo podría incluir características como la capacidad de aprendizaje, la capacidad de tomar decisiones y la capacidad de procesar información, mientras que el genotipo podría incluir información sobre los algoritmos y los parámetros utilizados para implementar esas características.



## **Tema 1**

### Test de Tourin

Procedimiento desarrollado para identificar la existencia de inteligencia en una máquina y comprobar que una máquina puede llegar a pensar al igual que una persona.

Fundamentado en la hipótesis de que, si una máquina se comporta en todos los aspectos de forma inteligente, debe ser inteligente.

Consiste en un desafío en el cual la máquina debe hacerse pasar por humana en una conversación con un hombre a través de una comunicación de texto en modo chat. Al sujeto no se le avisa si está hablando con un hombre o una persona de modo que, si el sujeto es incapaz de determinarlo, se considera que la máquina es inteligente.

### Agentes

Un agente es cualquier cosa capaz de percibir su entorno con la ayuda de sensores y actuar sobre dicho entorno mediante actuadores.

### IA Fuerte IA Débil

- **IA fuerte:** Construir programas que emulen el comportamiento inteligente de los humanos como el pensamiento, el aprendizaje, la visión, la resolución de problemas, la creatividad, etc. Ya que estos modos de comportamiento se pueden explicar algorítmicamente en términos de estados mentales.
- **IA débil:** Construir máquinas capaces de resolver problemas que requieran de inteligencia. Para ello se construyen programas que tengan un comportamiento inteligente sin importar si emula o no a la inteligencia humana.

## Tema 2

### Mirar un poco el espacio de estados

Algoritmos de búsqueda ciega:

- No tienen en cuenta el coste de la solución en la búsqueda
- Su funcionamiento es sistemático, siguen un orden de visitas y generación de nodos establecido por la estructura del espacio de búsqueda.
- Anchura prioritaria, profundidad prioritaria, profundidad iterativa.

4

---

Algoritmos heurísticos:

- Utilizan una estimación del coste de la solución para guiar la búsqueda.
- No siempre garantizan el óptimo, ni una solución.

### Algoritmos de búsqueda ciega o no informada

Basados en el espacio de estados. Sistemáticos y exhaustivos. Puedes acabar recorriendo todos los nodos para resolver problema. Los usas cuando son problemas pequeños. Se pueden usar para cualquier problema.

Hay 3 tipos:

Anchura prioritaria

Profundidad prioritaria

Profundidad iterativa

### Heurísticos o búsqueda informada

Necesitan conocimiento para reducir el coste de la búsqueda de la solución.

El coste del camino es el coste del camino recorrido más el coste del camino a recorrer.

$$F(n) = g(n) + h(n)$$

Búsqueda voraz el primero mejor utiliza una cola con prioridad.  
No se garantiza la solución óptima

Si el heurístico es admisible se asegura la optimalidad.

### Búsqueda local

Los algoritmos heurísticos o de búsqueda ciega nos sirven siempre que podamos plantear el problema como la búsqueda de un camino en un espacio de estados.

En este tipo de problemas puede ser relativamente fácil hallar una solución inicial, aunque no sea demasiado buena.

El saber cuándo hemos acabado la búsqueda dependerá totalmente del algoritmo, que tendrá que decidir cuando ya no es posible encontrar una solución mejor. Por lo tanto, no tendremos un estado final definido.

No hay un final definido. Puede haber una solución que sea una óptima, pero puede ser que el algoritmo no devuelva esta solución.

Función de calidad.

- Hill Climbing

---

Tenemos colina simple que coge el primer operador que suponga una mejora.

Tenemos máxima pendiente que de todos los operadores que tengo opciones a alcanzar cojo el que mejor resultado me dé.

Cuando usas estos algoritmos, al elegir un camino ya no hay vuelta atrás, por lo que se proponen las siguientes soluciones:



- Reiniciar la búsqueda en otro punto buscando mejorar la solución actual (Random Restarting Hill Climbing).
- Hacer backtracking a un nodo anterior y seguir el proceso en otra dirección (solo posible limitando la memoria para hacer el backtracking, Beam Search).
- Aplicar dos o más operaciones antes de decidir el camino
- Hacer Hill-Climbing en paralelo (p.ej. Dividir el espacio de búsqueda en regiones y explorar las más prometedoras, posiblemente compartiendo información).
- Algoritmos simulated annealing

Se utilizan para problemas de optimización combinatoria y continua.  
Para problemas grandes con óptimos rodeados de óptimos locales.

En primer lugar, denominaremos **función de energía** a la función heurística que nos mide la calidad de una solución.

Tendremos un parámetro de control que denominaremos **temperatura**, que nos permitirá controlar el funcionamiento del algoritmo.

El último elemento es la estrategia de **enfriamiento**. El algoritmo realizará un número total de iteraciones fijo y cada cierto número de ellas el valor de la temperatura disminuirá en cierta cantidad.

### Búsqueda con adversario

Dos agentes que juegan alternativamente, el max y el min, y comienza el max. El objetivo es encontrar un conjunto de movimientos accesibles que den como ganador al max. Se da por hecho que ambos jugadores tienen un conocimiento perfecto del juego. Juegan alternativamente y la accesibilidad de los estados depende de los movimientos del adversario.

Tampoco van a dar un óptimo porque todos se consideran igual de buenos.  
No podemos explorar todo el espacio.

Aproximación heurística: hay que definir una función que nos indique lo cerca que estamos de la jugada ganadora.

### Algoritmo Minimax

Aproximación heurística. Coges un árbol vas hasta el fondo y coges una jugada que sea un max.

### **Tema 3**

#### **Metaheurística**

Son de propósito general y de optimización. Los usamos cuando no hay un método exacto o que utilizamos es ineficiente o cuando no se necesita una solución óptima y queremos una buena solución.

Para obtener buenas soluciones, cualquier algoritmo de búsqueda debe establecer un balance adecuado entre dos características contrapuestas del proceso:

Intensificación: cantidad de esfuerzo empleado en la búsqueda en la región actual (explotación del espacio)

Diversificación: cantidad de esfuerzo empleado en la búsqueda en regiones distantes del espacio (exploración)

Clasificación: basadas en métodos constructivos, basadas en trayectorias y basadas en poblaciones.

Computación evolutiva: técnicas de optimización probabilística. Compuesta por modelos de evolución basados en poblaciones cuyos elementos representan soluciones a problemas.

Evolución artificial, 4 paradigmas:

- Algoritmos genéticos: operadores genéticos sobre cromosomas.
- Estrategias de evolución: cambios de comportamiento individuos.
- Programación evolutiva: cambios de comportamiento especies.
- Programación genética: árboles.

Aplicaciones de la computación evolutiva: Clasificación, aprendizaje, optimización estructural, control de procesos químicos, diseño de circuitos VLSI.