

# PREGUNTAS INTELIGENCIA ARTIFICIAL PRIMER PARCIAL

## 1.- El concepto de Agente. Agentes Racionales vs. Agentes Inteligentes. Arquitecturas de Agentes.

Se entiende por **agente** como una entidad que percibe y actúa sobre su entorno, mediante entradas a través de sus sensores, luego de procesar los datos de manera interna da un comportamiento. Hay 3 condiciones que hacen que sea un agente: que perciban y actúen, la proactividad (comportamiento anticipatorio) y que sean autónomos (que el sistema sea capaz de actuar sin la intervención de los humanos).

Un **agente racional** es un agente que actúa con la intención de alcanzar el mejor resultado esperado conforme su entorno (elige la opción mas optima o favorable entre las que tiene para conseguir su objetivo).

El **agente inteligente** es un agente racional que ha conseguido pasar el Test de Turing. Consiste en una prueba de habilidad de una maquina para conocer si tiene un comportamiento inteligente similar al de un ser humano, es decir, no saber diferenciar si estas hablando con una maquina o una persona.

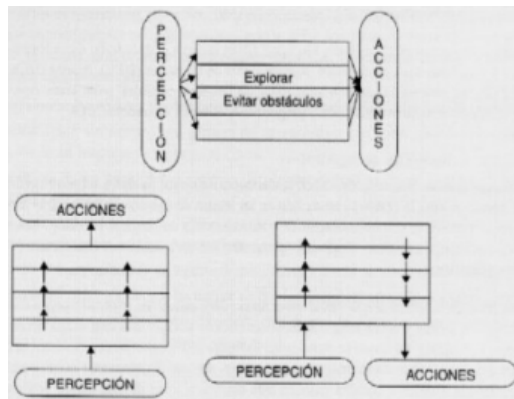
Podemos dividir la arquitectura de agentes en dos tipos: por su topología y su nivel de abstracción.

### Topología:

- **Arquitectura Horizontal:** En esta arquitectura todas las capas tienen acceso a los sensores y a los actuadores. Cada capa actúa por si misma como si fuera un agente, produciendo sugerencias de que acción realizar. Gracias a esto aseguramos la consistencia, pero obtenemos cuello de botella.
- **Arquitectura Vertical:** Solo una capa tiene acceso a los sensores y actuadores. Se va enviando a cada capa (a través de paso de mensajes) hasta llegar a conseguir la acción. El problema de esta arquitectura es que no es tolerante a fallos.
- **Arquitectura Híbrida:** Une las dos arquitecturas anteriores.

### Abstracción:

- **Arquitectura Deliberativa:** Contiene un mundo representado explícitamente de manera lógica, en el cual las decisiones son hechas por medio de un razonamiento lógico.
- **Arquitectura Reactiva:** Se caracteriza porque opera rápidamente y efectivamente sin la necesidad de procesar una representación simbólica del entorno. La decisión que toman esta basada en el presente, sin hacer uso del paso. Su acción es estimulo-respuesta.
- **Arquitectura Híbrida:** Une las dos arquitecturas anteriores.



## 2.- Características de los Agentes reactivos y deliberativos. Similitudes y diferencias. Arquitecturas.

Los **agentes reactivos** siguen un ciclo reacción-acción, a través de sus sensores, y no tienen ningún modelo del mundo, sino que lo van construyendo. Estos agentes se diseñan completamente y es necesario anticipar todas las posibles reacciones para todas las situaciones. Las características principales son que realizan pocos cálculos y que almacenan todo en memoria.

Algunas variantes de los agentes reactivos son: Agentes reactivos con memoria (donde se mejora la precisión teniendo en cuenta la historia sensorial previa) o sistemas basados en pizarras, la cual contiene las soluciones que se esta construyendo conforme al objetivo general del agente mediante Módulos de Conocimiento.

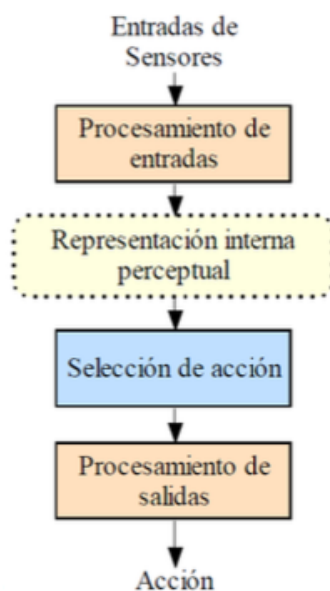
Los agentes reactivos responden directamente a las percepciones mediante técnicas como agentes en una tabla, sistema de producción, redes neuronales, arquitecturas de subsunción (basada en módulos de comportamiento), etc.

Los **agentes deliberativos** son aquellos que tienen un conocimiento del mundo y elaboran un plan según los efectos de sus acciones. Sus decisiones se realizan a través de un razonamiento lógico, razonando sobre los modelos elaborados para decidir que hacer para conseguir un objetivo. No reaccionan a los eventos.

Los agentes reactivos no incluyen ninguna representación interna del entorno, en cambio el deliberativo sí (aunque no siempre, hay a veces que un agente deliberativo no guarda la representación de forma interna). Se podría decir que un agente deliberativo es un agente que conoce todo, y el agente reactivo no, según vaya encontrándose va conociendo su entorno.

Los agentes reactivos suelen tener arquitectura horizontal, es decir, cada acción posible tiene acceso a la percepción que tiene el agente al entorno. En cambio, los agentes deliberativos suelen tener arquitecturas verticales.

Las arquitecturas son la búsqueda en un espacio de estados, búsqueda en grafos, búsqueda primero en profundidad y los problemas descomponibles.



### 3.- Describir brevemente los métodos de búsqueda no informada.

Los métodos de búsqueda no informada son estrategias de búsqueda en las cuales se evalúa el siguiente estado sin conocer a priori si este es mejor o peor que el anterior. No dependen de la información propia del sistema a la hora de resolverlo, sino que proporcionan métodos generales para recorrer los árboles de búsqueda asociados a la representación del problema, por lo que se pueden aplicar en cualquier circunstancia.

Tenemos cuatro búsquedas sin información: anchura, costo uniforme, profundidad y bidireccional.

- **Anchura:** Desde el estado inicial se deben analizar todos los nodos hijos antes de pasar al nivel siguiente en el árbol de búsqueda. ¿Qué quiere decir esto? Si el nodo padre tiene 3 hijos, primero debemos analizar estos 3 hijos antes de bajar a alguno de ellos, si después estos 3 nodos hijos tienen a su vez 5 hijos más tendremos que verlos uno a uno. {A B C D E F G H I J K L M}

Este algoritmo siempre encuentra solución en caso de haberla, pero tarda muchísimo ya que debemos explorar todos los nodos.

- **Profundidad:** Desde el estado inicial debemos ir analizando todos los nodos hijos a partir del nodo elegido. Es decir, si el nodo padre tiene 2 hijos, primero analizaremos el de mayor nivel, y así recursivamente, en caso de no encontrar solución por ese nodo, tomaremos el otro hijo y haremos el mismo proceso. {A B E F J K C G L M D H I}

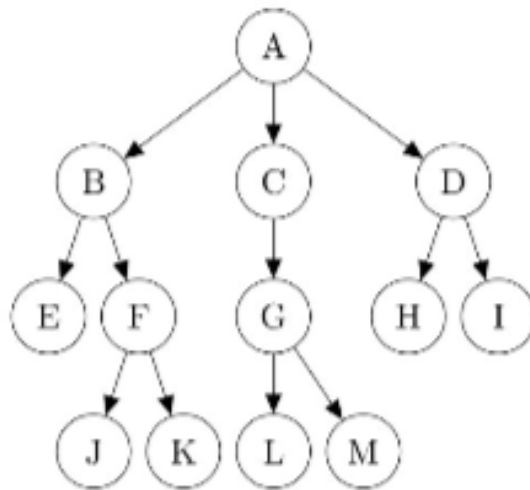
El algoritmo encuentra solución, pero no se garantiza que esta sea óptima ya que la solución encontrada será la primera explorada. Podría ser que por otro hijo se llegará antes. Este método de búsqueda tiene dos variantes más:

**Backtraking:** Es una técnica para resolver problemas de manera recursiva para construir una solución de forma incremental. Si se encuentra alguna alternativa incorrecta durante la búsqueda ésta retrocede hasta el paso anterior y toma la siguiente alternativa.

**Profundización Iterativa:** En este método de profundidad lo que hacemos es ir avanzando por el árbol, pero explorando nodo hijo a nodo hijo, es decir, si nuestro padre tiene dos hijos, verá primero uno y luego otro, si estos dos hijos tienen dos hijos más, veremos primero los dos hijos de uno y después los otros dos, pero donde el límite 'L' va incrementando gradualmente hasta encontrar el objetivo.

- **Bidireccional:** En esta búsqueda se ejecutan de forma simultánea dos búsquedas. Una empieza desde el estado inicial, y otra desde el objetivo. Se dice que se ha encontrado solución cuando las dos búsquedas se encuentran entre sí.

El problema de esta búsqueda es que necesitamos saber en todo momento donde se encuentra el objetivo. Y a veces esto no es así, sabemos que está ahí, pero no dónde exactamente. {A L B G C}



#### **4.- El concepto de heurística. Como se construyen las heurísticas. Uso de las heurísticas en IA.**

Una heurística es un criterio, método o principio para decidir cual, entre una serie de alternativas acciones, promete ser la más efectiva a la hora de lograr alguna meta. Es importante que la heurística elegida funcione bien ya que una mala decisión puede causar un gran problema para el funcionamiento del algoritmo.

La construcción de funciones heurísticas normalmente es a través de modelos simplificados. Es difícil encontrar una buena nada más empezar, ya que la iremos descubriendo poco a poco. Para construir una heurística para un problema, esta debe de cumplir varias condiciones: La heurística debe de resolver dicho problema, aunque la solución que devuelven no siempre es óptima. Debe tener en cuenta la información que le ha sido otorgada para la búsqueda de la solución. Debe ser simple, computable y manejable.

La Inteligencia Artificial se basa mayormente en el uso de algoritmos de búsqueda los cuales usan heurísticas para que funcionen de forma correcta. Aunque es cierto que no garantizan la solución óptima siempre produce resultados que satisfacen el problema. Incluso en ciertas ocasiones la heurística puede dar soluciones óptimas.

Las heurísticas se utilizan en Inteligencia Artificial para guiar las búsquedas ya que esto es esencial en todo método en Inteligencia Artificial. Por ejemplo, en algoritmos greedy se utiliza el criterio de selección para saber por donde ir para encontrar una solución, este criterio también es una heurística.

## 5.- Los métodos de escalada. Caracterización general. Variantes.

Los métodos de escalada son técnicas de búsqueda con información, en el cual buscan en un entorno local del nodo en curso. Tratan de elegir en cada paso un estado cuyo valor heurístico sea mayor que el del estado activo en ese momento.

Tenemos diferentes métodos de escalada: escalada simple, escalada por la máxima pendiente.

- **Escalada Simple:** En este método, en el momento que se encuentra un nodo más favorable que el que se está expandiendo, dicho nodo es devuelto sin generar el resto de nodos hijos. Este nodo más favorable pasa a ser el nodo en curso y sigue el proceso con normalidad.
- **Escalada por la máxima pendiente:** En este caso, se generan todos los hijos de un nodo, y se calculan sus valores heurísticos. Se determina uno de los nodos de mejor valor, y se compara dicho valor con el del nodo explorado. Si es mejor, éste pasa a ser el nodo en curso y sigue el proceso.

No es un algoritmo completo, es rápido y útil si la función es monótona. El problema de los métodos anteriores es la posibilidad de encontrarse con máximos/mínimos locales, etc.

Y además variantes del método de escalada como: estocástica, primera opción y reinicio aleatorio.

- **Estocástica:** Escoge aleatoriamente entre los sucesores con mejor valoración que el estado actual.
- **Primera opción:** Se generan aleatoriamente sucesores, escogiendo el primero con mejor valoración que el estado actual.
- **Reinicio aleatorio:** Se repite varias veces la búsqueda partiendo cada vez desde un nodo inicial distinto, generado aleatoriamente. El mejor nodo inicial es guardado y si posteriormente hay un nodo mejor que el guardado, lo reemplaza.

## 6.- Características esenciales de los métodos “primero el mejor”.

Este método, llamado “Primero el Mejor” es una de las técnicas de búsqueda con información. Avanzan a través del mejor nodo encontrado hasta el momento.

Son algoritmos basados en las variantes del algoritmo de Dijkstra, en vez de generar todos los mejores caminos nos quedamos con el primer nodo encontrado hasta ese momento.

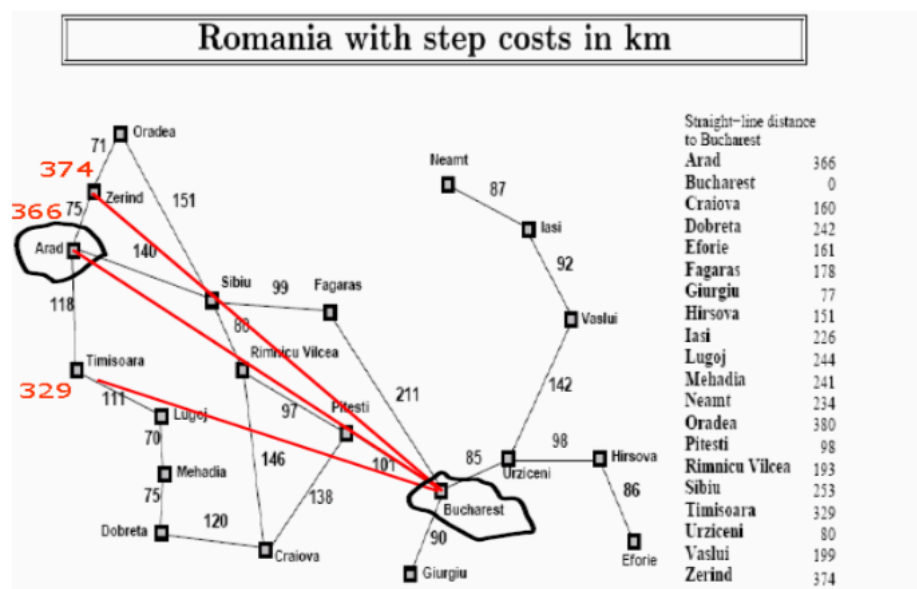
Utilizamos dos listas, una de nodos abiertos y otra de nodos cerrados.

La **lista de abiertos** son los nodos que se han generado y a los que se le ha aplicado la función heurística pero que aún no han sido explorados, hay que destacar que esta lista se ordena según el valor de la heurística. Puede considerarse como una cola de prioridades en la que los elementos con mayor prioridad son los que tienen los valores más prometedores, dados por la función heurística.

La **lista de cerrados** son aquellos nodos que han sido ya explorados. Es necesaria para que no exploremos un nodo que hemos explorado con anterioridad (se encontrará en esta lista), así la búsqueda es en un grafo y no en un árbol.

La selección de nodos se basa en una función de evaluación  $f(n)$ , consiste en evaluar los nodos del árbol dada una función heurística  $h(n)$ , seleccionar y expandir los nodos de valor más favorable. La función  $h(n)$  es el coste estimado del camino más barato desde el nodo  $n$  a un nodo objetivo. Como se ha comentado anteriormente estos algoritmos suelen implementarse con una cola con prioridad, donde se almacenan los datos en orden ascendente de  $f$ -valores, de esta manera se evalúa los nodos utilizando la función heurística  $f(n) = h(n)$ .

Se selecciona el mejor nodo de la lista de abiertos, se generan sus hijos y según la heurística llevada a cabo, se le calculan sus hijos a cada uno de ellos. Finalmente, se añaden a la lista de abiertos de forma ordenada según el valor de la heurística calculada. Tienen una característica y es que en cuanto aparece un nodo objetivo, se acaba pero puede NO ser el mejor camino.



Este algoritmo suele ser óptimo para la búsqueda de soluciones, pero consume mucho espacio en memoria. Un claro ejemplo de algoritmo “primero el mejor” es el algoritmo A\*

## 7.- Elementos esenciales del algoritmo A\*.

El algoritmo A\* pertenece a los métodos “primero el mejor” cuyo objetivo es el de obtener la ruta de un origen a un destino. Sus elementos esenciales son:

- Función de evaluación  $f(n)$ : es el mínimo coste calculado cuyo valor es la suma de  $g(n)$  y  $h(n)$  donde:

-  $h(n)$  es la distancia estimada desde la posición inicial a la posición de destino final. En ese caso se usa una función heurística para calcular su valor.

-  $g(n)$  es el coste que se toma de ir desde la posición inicial a la posición actual, es decir, la suma del coste de todo el recorrido realizado hasta el momento.

Para realizar dicho algoritmo necesitamos también una lista de **nodos Abiertos**, implementado como una cola de prioridad (ordenada por el valor  $f(n)$  de cada nodo). En esta lista se guardarán todos los nodos que se han identificado como posibles movimientos, pero aún no han sido evaluados. También una lista de **nodos Cerrados**: en esta lista se guardarán todos los nodos evaluados y descartados por no ser nodos válidos.

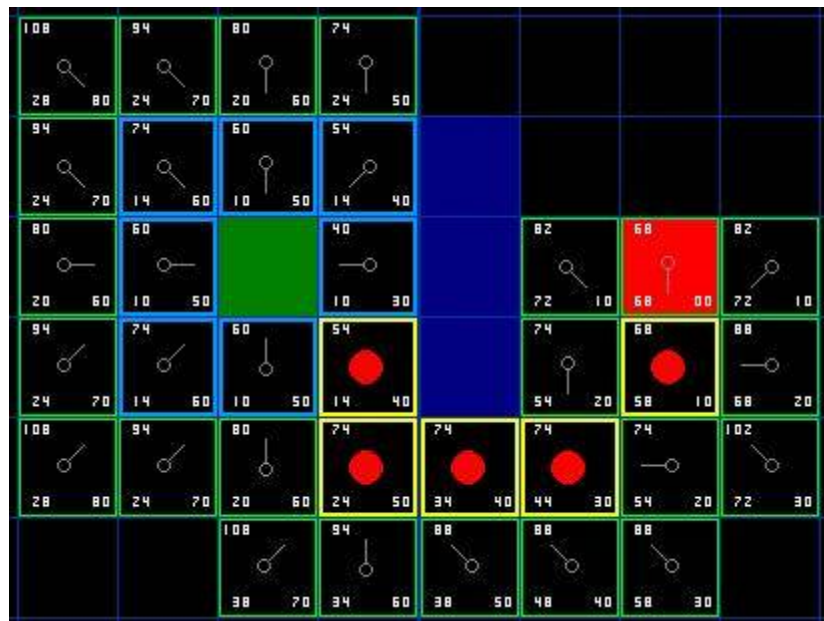
En cada paso del algoritmo, se expande el nodo que esté primero en abiertos, y en caso de que no sea un nodo objetivo, calcula la  $f(n)$  de todos sus hijos, los inserta en abiertos, y pasa el nodo evaluado a cerrados.

El algoritmo A\* selecciona el nodo más prometedor que se haya generado hasta ese momento (se encuentre en la lista de abiertos). A continuación, explora el nodo elegido generando así a todos sus nodos hijos. Si alguno de ellos es meta (ósea, el objetivo), el proceso acaba. De no ser así, el algoritmo continúa.

En sí el A\* es una búsqueda en anchura, si en una rama por la que está explorando no aparece la solución podemos abandonar la rama y explorar la nueva. Sin embargo, la vieja rama no se olvida, su último nodo se almacena en el conjunto de nodos generados sin explorar.

### Algoritmo:

1. Crear un grafo de búsqueda **G**. Inicializar **G** con **I**.
2. Crear ABIERTOS con **I**, crear CERRADOS vacío.
3. Si ABIERTOS está vacía terminar con FALLO.
4. Tomar primero nodo, **N**, de ABIERTOS e insertarlo en CERRADOS.
5. Si  $N = O$  entonces ÉXITO. Devolver **N** y el camino de **I** a **N** (usando el árbol de búsqueda dado por los punteros que se construyen en el paso 7).
6. Expandir **N** y crear el conjunto **M** con todos sus sucesores. Eliminar de **M** aquellos nodos que sean ancestros de **N** en **G**. Añadir **M** a los sucesores de **N** en **G**.
7. Establecer un puntero a **N** desde aquellos nodos de **M** que no estén ni en ABIERTOS ni en CERRADOS (no han sido visitados en **G**). Insertar estos elementos en ABIERTOS. Para cada nodo de **M** que esté en ABIERTOS o en CERRADOS modificar el puntero para que este apunte a **N** siempre que el mejor camino encontrados hasta el momento hacia dicho nodo pase por **N**.  
Para cada nodo de **M** que ya esté en CERRADOS, modificar los punteros de cada uno de sus descendientes de modo que apunten hacia atrás a los mejores caminos encontrados hasta el momento a esos descendientes.
8. Reordenar ABIERTOS según  $f(n)$  de menor a mayor. En caso de empate emplear el criterio de profundidad en el árbol de búsqueda.
9. Volver al paso 3.



Uno de los problemas del A\* es que necesita mucho espacio para que sea ejecutado, y el rendimiento de los algoritmos depende de la calidad de la función heurística.



## 8.- Elementos esenciales de un algoritmo genético.

Los algoritmos genéticos son métodos adaptativos que pueden usarse para resolver problemas de búsqueda y optimización. Están basados en el proceso genético de los organismos vivos. A lo largo de las generaciones, las poblaciones evolucionan en la naturaleza y por ello, dichos algoritmos genéticos van creando las soluciones basadas en el mundo real.

Un algoritmo genético es una variante de la búsqueda de haz estocástica en la que los estados sucesores se generan combinando dos estados padres. Aunque en la búsqueda de haz estocástica en vez de elegir los K mejores sucesores, se escoge a K sucesores aleatoriamente.

**Inicialización:** Se genera aleatoriamente la población inicial, que está constituida por un conjunto de cromosomas los cuales representan las posibles soluciones del problema.

**Evaluación:** A cada uno de los cromosomas de esta población se le aplicará la función de aptitud para saber qué tan “buena” es la solución que se está codificando.

**Selección:** Después de saber la aptitud de cada cromosoma, se procede a elegir los cromosomas que serán cruzados en la siguiente generación. Los cromosomas con mejor aptitud tienen mayor posibilidad de ser seleccionados.

**Recombinación:** La recombinación es el principal operador genético, representa la reproducción sexual, coge dos padres seleccionados y corta sus ristas de cromosomas en una posición escogida al azar, para producir dos subristas iniciales y dos subristas finales. Después se intercambian las subristas finales, produciéndose dos nuevos cromosomas completos donde se combinan las características de ambos cromosomas padres.

**Mutación:** Modifica al azar parte del cromosoma de los individuos, y permite alcanzar zonas del espacio de búsqueda que no estaban cubiertas por los individuos de la población actual.

**Reemplazo:** Una vez aplicados los operadores genéticos, se seleccionan los mejores individuos para conformar la población de la generación siguiente.

**Condición de término:** El AG se deberá detener cuando alcance la solución óptima, pero ésta generalmente se desconoce, por lo que se deben utilizar otros criterios de parada. Normalmente se usan 2 criterios:

- Correr el algoritmo un número máximo de iteraciones (generaciones).
- Detenerlo cuando no haya cambios en la función objetivo. Mientras no se cumpla la condición de término, se hace lo siguiente:

### **Algoritmos genéticos:**

Cromosoma: Vector representación de una solución al problema

Gen: Característica/Variables/Atributo concreto del vector de representación de una solución

Población: Conjunto de soluciones al problema

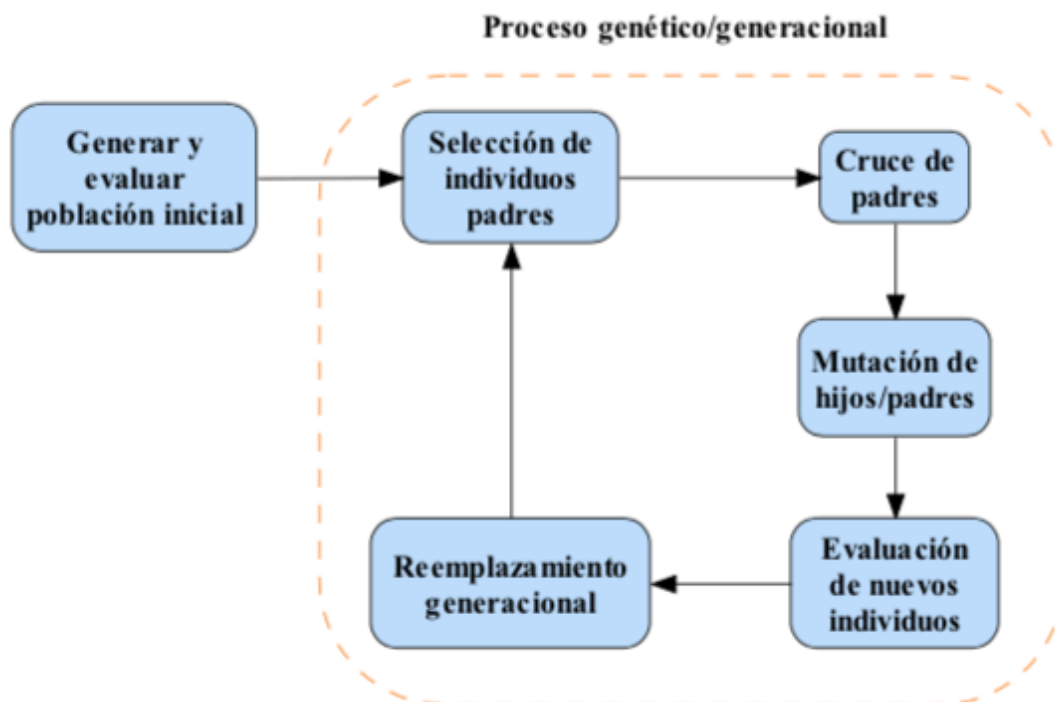
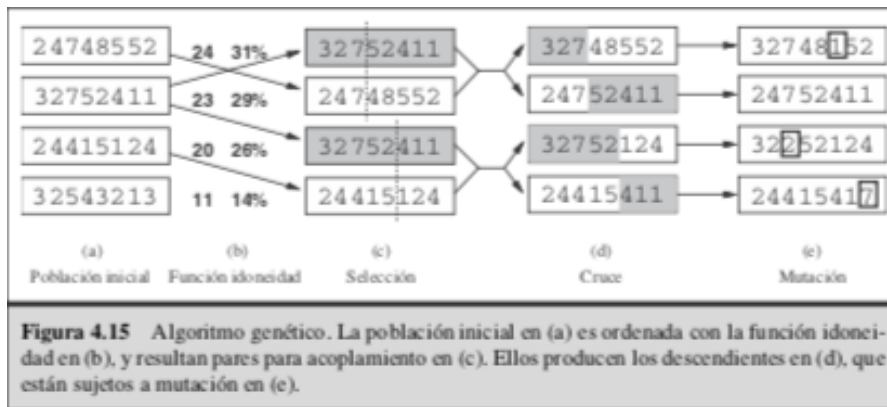
Adecuación al entorno: Valor de función objetivo

Selección natural: Operador de selección

Reproducción sexual: Operador de cruce

Mutación: Operador de mutación

Cambio generacional: Operador de reemplazamiento



Este algoritmo se basa en la reproducción sexual y el cruce. El proceso del algoritmo genético consiste en generar y evaluar la población inicial, pasando a la selección de los individuos padres, los cuáles cruzaremos, dando paso a una mutación de padres e hijos. Este proceso de cruce se puede realizar con muchos métodos, con el fin de evaluar los nuevos individuos, hacemos un reemplazamiento generacional, tras el que se reanuda el bucle, empezando de nuevo con la selección de los individuos padres.