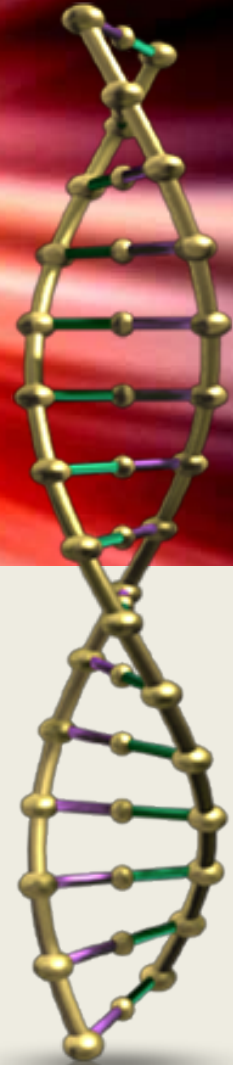


ALGORITMOS GENÉTICOS

“El problema del Viajante”

Prof. Ing. Daniela E. Díaz



*¿Cómo podemos definir la palabra **Búsqueda**?*

Según el Diccionario:

Búsqueda: acción de buscar.

Buscar: hacer lo necesario para encontrar a una persona o cosa.

*¿ Y entonces a qué se llama **Búsqueda**?*

La búsqueda es un método computacional para resolver problemas.



¿Qué problemas se pueden resolver a través de búsquedas?

- single-agent path-finding: único agente encuentra camino
- two-player games: juegos de dos jugadores
- constraint satisfaction: satisfacción de restricciones

Características:

- problemas difíciles: NP-completos
- la solución se calcula por enumeración
- se supone que requieren inteligencia → Inteligencia Artificial



Búsqueda Exhaustiva:

Es una técnica general de resolución de problemas. Se realiza una búsqueda exhaustiva y sistemática en el espacio de soluciones. Por ello, suele resultar ineficiente.

La búsqueda se suele realizar recorriendo un árbol con el que se representan las posibles soluciones.

Hay algunos métodos de recorrido del árbol:

- ❖ **Recorrido en anchura** (consiste en visitar todos los elementos del árbol una sola vez, de la forma en que primero se visitan los elementos del nivel 0, luego los del nivel 1, y así sucesivamente. En cada nivel, se visitan los elementos de izquierda a derecha).
- ❖ **Backtracking** (el algoritmo realiza una búsqueda en profundidad en el árbol de soluciones del problema).
- ❖ **Branch and Bound** (ramificación: se guiará por estimaciones de beneficio que se harán en cada nodo y acotación: para eliminar nodos que no lleven a la solución óptima).



La solución de un problema se puede expresar como una tupla (x_1, x_2, \dots, x_n) , satisfaciendo unas restricciones $P(x_1, x_2, \dots, x_n)$ y tal vez optimizando una cierta función objetivo.

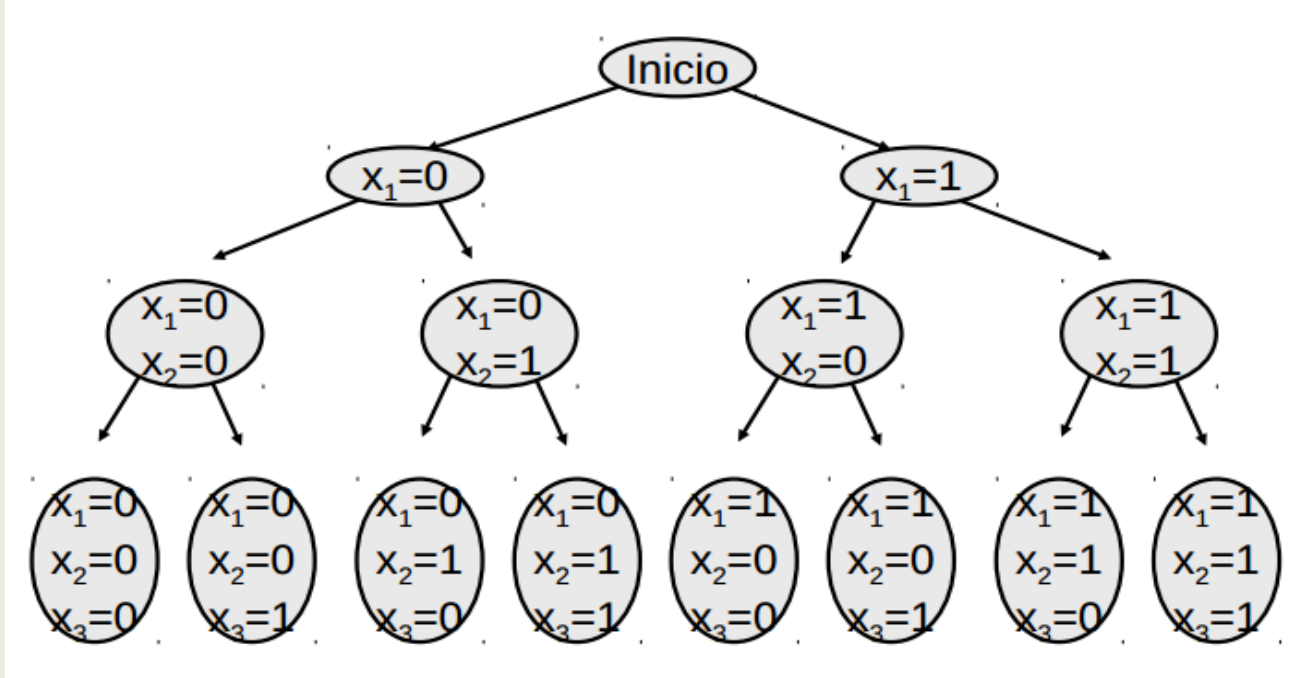
En cada momento, el algoritmo se encontrará en un cierto nivel k , con una solución parcial (x_1, \dots, x_k) .

Cada conjunto de posibles valores de la tupla representa un nodo del árbol de soluciones.

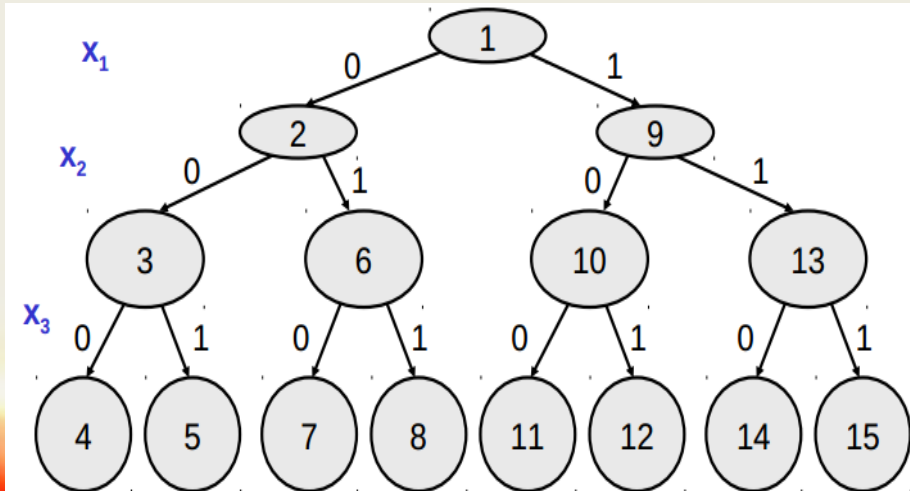
Se sigue hasta que la solución parcial sea una solución completa del problema, o hasta que no queden más posibilidades por probar.

Se recorre un árbol de soluciones. Sin embargo, este árbol es implícito, no se almacena en ningún lugar.





El recorrido se hace en un cierto orden. Por ejemplo, con **backtracking** se hace en profundidad:



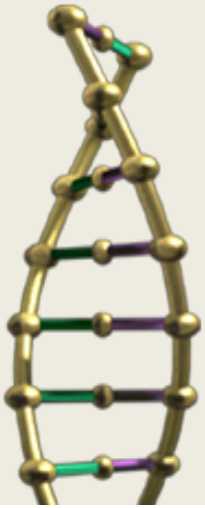


Los algoritmos de **Backtracking** y **Branch and Bound** se suelen aplicar en la resolución de un gran número de problemas, muy especialmente en los de optimización.

Estos algoritmos realizan una búsqueda exhaustiva y sistemática en el espacio de soluciones del problema.

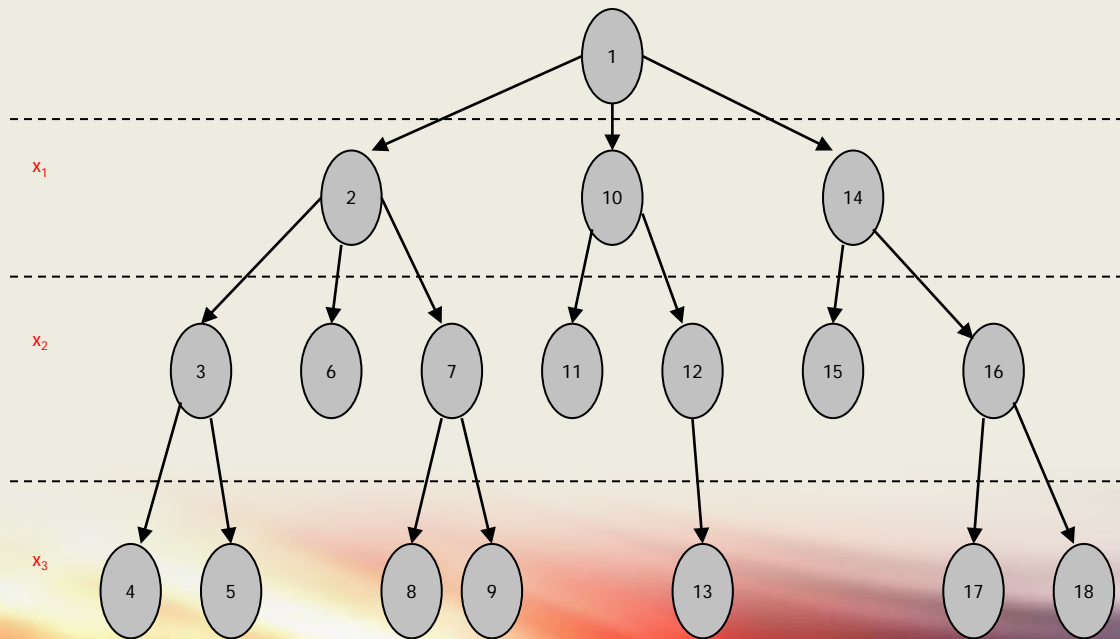
Suelen ser ***muy ineficientes***. Se utilizan para resolver problemas para los que no existe un algoritmo eficiente para resolverlos

Backtracking: es una técnica general de resolución de problemas, que suele aplicarse sobre todo a juegos y problemas de optimización.



El algoritmo realiza una búsqueda en profundidad en el árbol de soluciones del problema.

Por realizar una búsqueda exhaustiva en el espacio de soluciones del problema, los algoritmos de backtracking son bastante ineficientes. Se tienen tiempos con órdenes de complejidad factoriales o exponenciales





Branch and Bound: es una técnica similar a Backtracking. Suele utilizarse en problemas de optimización. Esta técnica puede verse como una mejora del esquema de backtracking. Al igual que en backtracking, el algoritmo realiza una búsqueda sistemática en un árbol de soluciones.

Añade dos características nuevas:

Estrategia de ramificación: La búsqueda se guiará por estimaciones de beneficio que se harán en cada nodo.

Estrategia de poda: Para eliminar nodos que no lleven a la solución óptima.



El esquema de branch and bound suele obtener mejores resultados que el de backtracking, sin embargo sigue siendo muy ineficiente.



Búsqueda Heurística



Heurística: procedimiento: criterio que puede resolver un problema pero que no hay garantía de que siempre lo resuelva.

Función: estimación del coste necesario para alcanzar una solución desde el estado actual.

Los métodos de búsqueda heurística disponen de alguna información sobre la proximidad de cada estado a un estado objetivo, lo que permite explorar en primer lugar los caminos más prometedores.

Son características de los métodos heurísticos:

- ☐ No garantizan que se encuentre una solución, aunque existan soluciones.
- ☐ Si encuentran una solución, no se asegura que ésta tenga las mejores propiedades (que sea de longitud mínima o de coste óptimo).
- ☐ En algunas ocasiones (que, en general, no se podrán determinar a priori), encontrarán una solución (aceptablemente buena) en un tiempo razonable.



Búsqueda Heurística

En general, los métodos heurísticos son preferibles en la solución de problemas difíciles a aquellos en los que utilizar una búsqueda exhaustiva necesitaría un tiempo demasiado grande. Esto cubre prácticamente la totalidad de los problemas reales que interesan en Inteligencia Artificial.

"Un proceso que puede resolver un problema dado, pero que no ofrece ninguna garantía de que lo hará, se llama una heurística para ese problema".



El problema del Viajante

El problema del viajante (también conocido como problema del viajante de comercio o por sus siglas en inglés: TSP (Traveling Salesman Problem), es uno de los problemas más famosos (y quizás el mejor estudiado) en el campo de la optimización combinatoria computacional.

A pesar de la aparente sencillez de su planteamiento, el TSP es uno de los más complejos de resolver .


DEFINICIÓN:

Sean N ciudades de un territorio.

La distancia entre cada ciudad viene dada por la matriz D : $N \times N$, donde $d[x,y]$ representa la distancia que hay entre la ciudad X y la ciudad Y .

El objetivo es encontrar una ruta que, Comenzando y terminando en una ciudad concreta, pase una sola vez por cada una de las ciudades y minimice la distancia recorrida por el viajante.





El **TSP** está entre los problemas denominados NP-Complejos, esto es, los problemas que no se pueden resolver en tiempo polinomial en función del tamaño de la entrada (en este caso el número N de ciudades que el viajante debe recorrer).

La solución más directa es la que aplica la **fuerza bruta**: evaluar todas las posibles permutaciones y quedarse con la mejor.

No obstante, el número de posibles ciclos viene dado por el factorial del número de ciudades ($N!$) y esto hace que la solución por **fuerza bruta** sea **impracticable para valores de N incluso moderados**.



Por ejemplo, si un ordenador fuese capaz de calcular la longitud de cada ciclo en un microsegundo, tardaría algo más de 3 segundos en resolver el problema para 10 ciudades, algo más de medio minuto en resolver el problema para 11 ciudades y. 77.146 años en resolver el problema para sólo 20 ciudades.

Queda demostrado que el requerimiento de volver a la ciudad de partida no cambia la complejidad computacional del problema.

El problema tiene considerables aplicaciones prácticas, a parte de las más evidentes en áreas de logística de transporte. Por ejemplo, en robótica, permite resolver problemas de fabricación para minimizar el número de desplazamientos para conseguir realizar un número determinado de perforaciones en una plancha o en un circuito impreso.

Ejercicios:

1. Hallar la ruta de distancia mínima que logre unir todas las capitales de provincias de la República Argentina, utilizando un **método exhaustivo**. ¿Puede resolver el problema? Justificar de manera teórica.
2. Realizar un programa que cuente con un menú con las siguientes opciones:
 - a) Permitir ingresar una provincia y hallar la ruta de distancia mínima que logre unir todas las capitales de provincias de la República Argentina partiendo de dicha capital utilizando la siguiente **heurística**: ***“Desde cada ciudad ir a la ciudad más cercana no visitada.”*** Recordar regresar siempre a la ciudad de partida. Presentar un mapa de la República con el recorrido indicado. Además indicar la ciudad de partida, el recorrido completo y la longitud del trayecto. El programa deberá permitir seleccionar la capital que el usuario desee ingresar como inicio del recorrido.
 - b) Encontrar el recorrido mínimo para visitar todas las capitales de las provincias de la República Argentina siguiendo la heurística mencionada en el punto a. Deberá mostrar como salida el recorrido y la longitud del trayecto.
 - c) Hallar la ruta de distancia mínima que logre unir todas las capitales de provincias de la República Argentina, utilizando un **algoritmo genético**.



Recomendaciones para el algoritmo:

N = 50 Número de cromosomas de las poblaciones.

M = 200 Cantidad de ciclos.

Cromosomas: permutaciones de 23 números naturales del 1 al 23 donde cada gen es una ciudad.

Las frecuencias de crossover y de mutación quedan a criterio del grupo.
Se deberá usar crossover cíclico.

Comparar los resultados obtenidos entre la resolución a través de heurísticas y con algoritmos genéticos a través de una conclusión que deberá anexarse al informe.

Agregar en el informe un apartado final denominado «Aportes Prácticos del TSP» donde se expliquen algunas aplicaciones en las que actualmente se use el problema del viajante. Tomar por lo menos dos y explicarlas.



Observaciones

- El programa debe tener las siguientes opciones:

Resolución por Búsqueda heurística:

- Mostrar el recorrido mínimo, nombrar todas las ciudades que lo componen y la cantidad de kms recorridos.
- Permitir seleccionar una ciudad y mostrar el recorrido partiendo de la misma y volviendo a ésta habiendo visitado una vez todas las ciudades. También mostrar la cantidad de kms recorridos.

Resolución por AG:

- Mostrar el recorrido mínimo, nombrar todas las ciudades que lo componen y la cantidad de kms recorridos.





¿Preguntas?

