

# Introducción al curso resolución de problemas con algoritmos de búsqueda

Stalin Muñoz Gutiérrez

Centro de Ciencias de la Complejidad  
Universidad Nacional Autónoma de México (UNAM)

## Introducción al curso resolución de problemas con algoritmos de búsqueda

2018-09-04

Introducción al curso resolución de problemas con algoritmos de búsqueda

Stalin Muñoz Gutiérrez

Centro de Ciencias de la Complejidad  
Universidad Nacional Autónoma de México (UNAM)

Bienvenidos al curso de Resolución de problemas por medio de algoritmos de búsqueda.

En el curso daremos a conocer algoritmos de búsqueda que son muy importantes para la Inteligencia Artificial, y que pueden utilizarse para resolver problemas diversos siempre que podamos abstraer el problema a nuestro marco conceptual.

Nos interesará identificar apropiadamente las características del problema a resolver pues no existe un algoritmo universal que pueda desempeñarse bien en todos los escenarios.

El objetivo del curso es que puedas abstraer problemas como grafos de estados-acciones y puedas seleccionar o adaptar el mejor algoritmo de búsqueda para resolverlo.

## Contenido

- Abstracción de un problema como un grafo de estados-acciones.

## Introducción al curso resolución de problemas con algoritmos de búsqueda

└ De que tratará el curso

└ Contenido

¿De qué tratará el curso?

El primer punto a tratar será el de abstraer un problema como un grafo de estados-acciones.

Para poder hacer esto hay que simplificar el dominio de aplicación e identificar tanto los estados como las acciones del agente.

Adicionalmente será necesario establecer ciertas restricciones para que los algoritmos que presentaremos funcionen adecuadamente.

## Contenido

- Abstracción de un problema como un grafo de estados-acciones.
- Criterios de selección de algoritmos.

## Introducción al curso resolución de problemas con algoritmos de búsqueda

└ De que tratará el curso

└ Contenido

La clave del éxito esta en seleccionar el algoritmo idóneo al problema a resolver.

Así como la elección adecuada de sus parámetros.

En el caso de algoritmos informados, tendremos que diseñar funciones que guien la búsqueda hacia una rápida convergencia a la solución.

## Contenido

- Abstracción de un problema como un grafo de estados-acciones.
- Criterios de selección de algoritmos.
- Algoritmos de búsqueda ciega.

## Introducción al curso resolución de problemas con algoritmos de búsqueda

└ De que tratará el curso

└ Contenido

Comenzaremos con un tipo de algoritmos conocido como de búsqueda ciega.

Estos algoritmos son los más básicos y, argumentaremos, los más importantes. Es necesario conocer su comportamiento y saber como adaptarlos para articular búsquedas más inteligentes.

- Abstracción de un problema como un grafo de estados-acciones.
- Criterios de selección de algoritmos.
- Algoritmos de búsqueda ciega.

## Contenido

- Abstracción de un problema como un grafo de estados-acciones.
- Criterios de selección de algoritmos.
- Algoritmos de búsqueda ciega.
- Algoritmos de búsqueda informada.

## Introducción al curso resolución de problemas con algoritmos de búsqueda

### De que tratará el curso

#### Contenido

En ocasiones podemos utilizar conocimiento sobre el dominio del problema para construir ciertas funciones heurísticas que pueden guiar la búsqueda hacia las soluciones de una manera más rápida.

Los algoritmos de búsqueda informada utilizan estas funciones heurísticas.

- Abstracción de un problema como un grafo de estados-acciones.
- Criterios de selección de algoritmos.
- Algoritmos de búsqueda ciega.
- Algoritmos de búsqueda informada.

## Contenido

- Abstracción de un problema como un grafo de estados-acciones.
- Criterios de selección de algoritmos.
- Algoritmos de búsqueda ciega.
- Algoritmos de búsqueda informada.
- Algoritmos metaheurísticos.

## Introducción al curso resolución de problemas con algoritmos de búsqueda

└ De que tratará el curso

└ Contenido

Existe un gran número de algoritmos que toman inspiración en procesos físicos y biológicos.

Presentaremos algunos algoritmos metaheurísticos para solución de problemas.

Estos algoritmos son muy generales y asumen poco, lo que permite aplicarlos en muchos tipos de problemas. Sin embargo suelen ser computacionalmente demandantes.

- Abstracción de un problema como un grafo de estados-acciones.
- Criterios de selección de algoritmos.
- Algoritmos de búsqueda ciega.
- Algoritmos de búsqueda informada.
- Algoritmos metaheurísticos.

## Contenido

- Abstracción de un problema como un grafo de estados-acciones.
- Criterios de selección de algoritmos.
- Algoritmos de búsqueda ciega.
- Algoritmos de búsqueda informada.
- Algoritmos metaheurísticos.
- Implementación de programas en lenguaje Python.

## Introducción al curso resolución de problemas con algoritmos de búsqueda

└ De que tratará el curso

### └ Contenido

Nos interesa que puedas aplicar los algoritmos a problemas concretos.

Te acompañaremos en la implementación de los algoritmos en el lenguaje de programación Python y te mostraremos algunos ejemplos de su aplicación a ciertos problemas modelo.

Al final podrás probar tus algoritmos en un espacio de búsqueda interesante: el resolver el cubo de Rubik.

- Abstracción de un problema como un grafo de estados-acciones.
- Criterios de selección de algoritmos.
- Algoritmos de búsqueda ciega.
- Algoritmos de búsqueda informada.
- Algoritmos metaheurísticos.
- Implementación de programas en lenguaje Python.

### Definición del problema a resolver

- Definición de problema

Partamos de que queremos diseñar un agente racional para solucionar un problema particular.

Decimos que un agente es racional, si actúa de la mejor manera posible de acuerdo a sus preferencias y sujeto a sus limitaciones computacionales. Este tipo de agente exhibe lo que se denomina racionalidad limitada.

En nuestro caso particular nos interesa implantar un algoritmo de búsqueda que permita al agente resolver problemas concretos.





## Definición de problema



Introducción al curso resolución de problemas con algoritmos de búsqueda

└ Definición del problema a resolver

└ Definición de problema

Definición de problema



Los algoritmos que trataremos son aplicables a cierto tipo de problemas.

Comenzaremos por definir que entenderemos por problema.

## Definición del problema

- Comenzamos por identificar los estados relevantes del problema.

## Introducción al curso resolución de problemas con algoritmos de búsqueda

### Definición del problema a resolver

#### Definición del problema

Lo primero que hay que hacer es identificar el espacio de estados.  
Qué variables son relevantes al problema a resolver.

Como ejemplo, consideremos que queremos encontrar la ruta  
carretera que nos lleva de la Ciudad de México a la Ciudad de Colima.

Aquí el lugar donde nos encontramos es un estado relevante.

El lugar al que vamos es un estado relevante.

Los caminos carreteros que existen son relevantes.

Sin embargo tenemos que abstraer, reducir el detalle al máximo.

No tomaremos en cuenta todos los lugares posibles.

Solo las ciudades intermedias por las que tenemos que pasar.

No nos interesará modelar la orografía, las estaciones de gasolina,  
los cruces y desviaciones carreteros.

Para nuestro problema consideraremos únicamente cuando nos  
encontramos en una ciudad. La transición entre ellas será instantánea, no  
nos interesa lo que pase en el camino entre cada ciudad.

## Definición del problema

- Comenzamos por identificar los estados relevantes del problema.
- Después identificamos las acciones del agente.

## Introducción al curso resolución de problemas con algoritmos de búsqueda

### Definición del problema a resolver

#### Definición del problema

Después decidimos cuales son las acciones del agente.

Nos interesa el número mínimo de acciones que puedan articularse en el tiempo para resolver el problema de manera aceptable.

En este ejemplo las acciones serán ir de una ciudad a otra.

Ir de la Ciudad de México a Toluca.

Ir de la Ciudad de México a Queretaro, etc.

- Comenzamos por identificar los estados relevantes del problema.
- Después identificamos las acciones del agente.

## Definición del problema

- Comenzamos por identificar los estados relevantes del problema.
- Después identificamos las acciones del agente.
- Conectaremos los estados con otros estados a través de acciones del agente, formando un [grafo de estados-acciones](#).

## Introducción al curso resolución de problemas con algoritmos de búsqueda

### Definición del problema a resolver

#### Definición del problema

Con los estados y acciones identificados construiremos un grafo en el que los vertices son los estados y las aristas las acciones del agente.

- Comenzamos por identificar los estados relevantes del problema.
- Después identificamos las acciones del agente.
- Conectaremos los estados con otros estados a través de acciones del agente, formando un [grafo de estados-acciones](#).

The map illustrates the proposed Line 8 (Guerrero) in Mexico City. The line is shown as a green route with white circular stations. Key stations include Buenavista, Garibaldi / Lagunilla (marked with a green circle and the number 8), Tepito, Morelos, Candelaria, and Isabel la Católica. The line connects to several existing Metro lines: Line 1 (blue) at Garibaldi / Lagunilla, Line 2 (blue) at Tepito, Line 7 (pink) at Candelaria, and Line 9 (yellow) at Isabel la Católica. Other lines shown include Line 4 (red), Line 5 (orange), and Line 6 (purple). The map also shows major roads like EJE 1 NORTE, EJE 1 OTE, and EJE 2 OTE, and landmarks like the Canal del Norte and the Av. del Tránsito.

### Definición del problema a resolver

└ Grafo de estados-acciones

Como un segundo ejemplo podemos considerar el diseño de un asistente inteligente para que nos guie con la movilidad en la Ciudad de México.



A detailed map of the Toluca Metro system. The 8th station, Garibaldi / Lagunilla, is highlighted in green. The map shows several lines: a green line (Line 8) running horizontally, a blue line (Line 1) running vertically, and a pink line (Line 2) running horizontally. Other stations shown include Buenavista, San Cosme, Hidalgo, Bellas Artes, Lagunilla, Centro, Allende, Juárez, Salto del Agua, San Juan de Letrán, Zócalo, Pino Suárez, Isabel la Católica, Fray Servando, Merced, Candelaria, Tepito, Morelos, Albañiles, and Cana del N. The map also shows the Eje 1 Norte and Eje 2 Ote roads.

## Definición del problema a resolver

└ Grafo de estados-acciones

Partimos de identificar los estados relevantes.

Los lugares, desde luego.

Aquí las estaciones del metro pueden formar parte de los estados.



## Definición del problema a resolver

└ Grafo de estados-acciones

Ahora identificamos las acciones.

En este caso las conexiones que nos llevan de un estado a otro.

En modelos más refinados quizá nos interesa también los transbordos entre líneas.

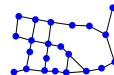


A graph with 18 blue circular nodes and black edges. The nodes are arranged in a complex, interconnected network. The edges connect the nodes in a way that forms several loops and paths. The overall structure is somewhat irregular, with some nodes having multiple connections and others having only one or two.

## Definición del problema a resolver

- Grafo de estados-acciones

Finalmente nuestro grafo de estados-acciones tiene la simplicidad suficiente para poder ser usado para planificar rutas.





## Otros ejemplos

- 1 El rompecabezas del ocho o del quince.

Introducción al curso resolución de problemas con algoritmos de búsqueda

└ Definición del problema a resolver

└ Otros ejemplos

Otros ejemplos

■ El rompecabezas del ocho o del quince.

Para ilustrar algunos de los algoritmos del curso usaremos rompecabezas.

La razón es que son adecuados para explicar los algoritmos.

El espacio de estados es suficientemente grande para ser interesante al comparar el desempeño de los algoritmos.

En el rompecabezas del quince un estado es una configuración.

*AQUI HACEMOS LA EXPLICACION CON EL ROMPECABEZAS*

Las acciones del agente son los movimientos de las fichas.

En esta configuración tenemos varios estados a los que podemos llegar.

De cada uno de estos estados podemos llegar a otros estados.

Así, aunque no construyamos de manera anticipada el grafo.

Podemos entender que existe un grafo de estados-acciones subyacente.

## Otros ejemplos

- 1 El rompecabezas del ocho o del quince.
- 2 El cubo de rubik.

Introducción al curso resolución de problemas con algoritmos de búsqueda

└ Definición del problema a resolver

└ Otros ejemplos

Otros ejemplos

- El rompecabezas del ocho o del quince.
- El cubo de rubik.

Así también para el cubo de Rubik.

*AQUI HACEMOS LA EXPLICACION CON EL CUBO*

Un estado es la configuración de colores de cada sub-cubo.

Hay varias formas de definir las acciones.

Una forma es rotar 90 grados a la izquierda o derecha cada cara.

También se pueden incluir los giros de 180 grados.

Si consideramos este último caso tenemos que para cada configuración tenemos 3 acciones por cara del cubo.

Si tenemos 6 caras en el cubo el número de posibles acciones es  $3 \times 6$  o 18 acciones posibles.

## Otros ejemplos

- 1 El rompecabezas del ocho o del quince.
- 2 El cubo de rubik.
- 3 Planificación automática para un robot.

## Introducción al curso resolución de problemas con algoritmos de búsqueda

### └ Definición del problema a resolver

#### └ Otros ejemplos

- El rompecabezas del ocho o del quince.
- El cubo de rubik.
- Planificación automática para un robot.

También podríamos usarlo para hacer planificación automática en un robot.

En planificación automática se busca encontrar una secuencia de acciones que nos lleve de un estado inicial a uno final.

Los estados pueden representarse con variables Booleanas.

Las acciones del agente pueden cambiar el estado de dichas variables.

Para ello se puede usar una descripción lógica del efecto de las acciones, así como restricciones sobre su aplicabilidad.

Un ejemplo de este tipo de construcción es el algoritmo denominado GRAPH-plan, donde el problema de planificación se convierte en un grafo de estados acciones.

Un algoritmo de búsqueda puede resolver el problema de encontrar la articulación de las acciones en el tiempo.

# Complejidad e Inteligencia Artificial

## Complejidad Computacional

En Inteligencia Artificial los problemas a resolver tienen la característica de que son **computacionalmente complejos**.

Esto quiere decir que no se conocen algoritmos que puedan resolver los problemas de manera eficiente.

Los problemas dejan de ser materia de la Inteligencia Artificial cuando algoritmos eficientes se desarrollan.

Introducción al curso resolución de problemas con algoritmos de búsqueda

└ Complejidad e Inteligencia Artificial

└ Complejidad e Inteligencia Artificial

La complejidad computacional clasifica los problemas de acuerdo a su complejidad inherente.

Cuando un problema es fácil significa en términos computacionales que puede resolverse con pocos recursos computacionales.

Los recursos computacionales son la memoria y el tiempo o número de pasos necesario para solucionarlo.

La mayoría de los investigadores considera a los problemas de la Inteligencia Artificial como de alta complejidad computacional.

## Análisis asintótico de algoritmos

- En el análisis asintótico de algoritmos nos interesa poder acotar los requerimientos de un programa ya sea memoria o tiempo con una función.
- Analizamos el peor de los escenarios para un algoritmo, es decir, cuando este consume el máximo de recursos computacionales.

Decimos que un algoritmo que consume un recurso a razón de una función  $T(n)$  con el parámetro  $n$ , es  $O(f(n))$  si existen  $n_0, k \in N$  y  $f(n)$  tal que:

$$T(n) \leq kf(n) \text{ para toda } n > n_0.$$

## Introducción al curso resolución de problemas con algoritmos de búsqueda

### └ Complejidad e Inteligencia Artificial

#### └ Análisis asintótico de algoritmos

##### Análisis asintótico de algoritmos

- En el análisis asintótico de algoritmos nos interesa poder acotar los requerimientos de un programa ya sea memoria o tiempo con una función.
- Analizamos el peor de los escenarios para un algoritmo, es decir, cuando este consume el máximo de recursos computacionales.

Decimos que un algoritmo que consume un recurso a razón de una función  $T(n)$  con el parámetro  $n$ , es  $O(f(n))$  si existen  $n_0, k \in N$  y  $f(n)$  tal que:  
 $T(n) \leq kf(n)$  para toda  $n > n_0$ .

Para poder comparar el desempeño de los algoritmos y decidir sobre que algoritmo nos conviene más usaremos la notación de O grande.

Esta notación indica que el recurso computacional que consume un algoritmo, ya sea tiempo o espacio puede acotarse por una función.

Al usar esta caracterización no nos interesa conocer a detalle el consumo exacto de recursos, sino simplemente ver como crece este consumo con el tamaño del problema.

# Análisis asintótico de algoritmos

- Algoritmo para encontrar el valor máximo en una lista de números no ordenados.

```

1  $m \leftarrow -\infty$ 
2 for  $i \in \text{lista}$ 
    1 if  $i > m : m \leftarrow i$ 
3 return  $m$ 
```

En este caso decimos que el algoritmo es  $O(n)$ .

## Introducción al curso resolución de problemas con algoritmos de búsqueda

### Complejidad e Inteligencia Artificial

#### Análisis asintótico de algoritmos

##### Análisis asintótico de algoritmos

```

■ Algoritmo para encontrar el valor máximo en una lista de
números no ordenados.
1  $m \leftarrow -\infty$ 
2 for  $i \in \text{lista}$ 
    1 if  $i > m : m \leftarrow i$ 
3 return  $m$ 
```

En este caso decimos que el algoritmo es  $O(n)$ .

Por ejemplo en este algoritmo que encuentra el máximo de una lista no ordenada, para encontrar el máximo hay que recorrer todos los elementos de la lista, por lo tanto, si  $n$  es el tamaño de la lista entonces el número de pasos puede acotarse con una línea recta.

En otras palabras el algoritmo es  $O(n)$ , es decir crece linealmente con  $n$ .