

# Algoritmos de Búsqueda Informada

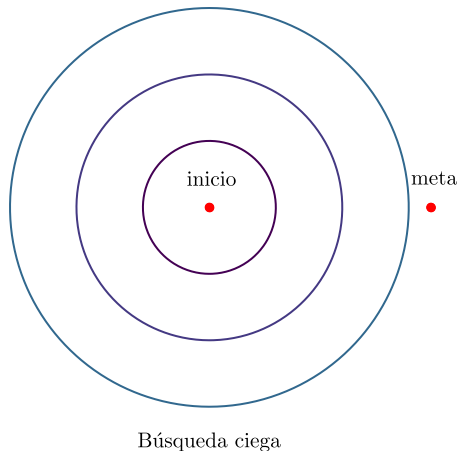
Stalin Muñoz Gutiérrez

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

Hoy presentaremos a los denominados algoritmos de búsqueda informada.

A diferencia de los algoritmos de búsqueda ciega, los algoritmos de búsqueda informada utilizan conocimiento del dominio del problema para orientar la frontera de exploración en la dirección de la meta.

## Búsqueda ciega contra búsqueda informada



## Búsqueda ciega contra búsqueda informada

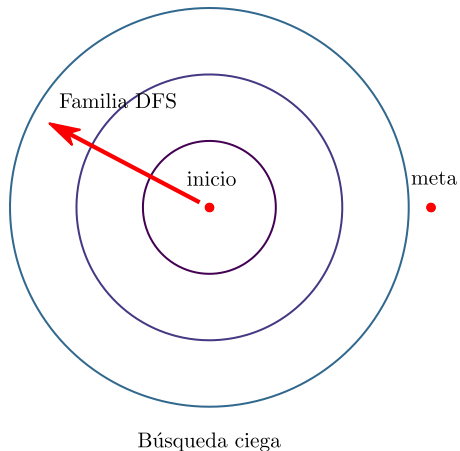


Usaremos un diagrama para ilustrar la diferencia entre los tipos de algoritmos.

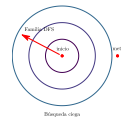
Comencemos con los algoritmos de búsqueda ciega.

Estos algoritmos disponen únicamente de información de los estados que van descubriendo durante la exploración.

## Búsqueda ciega contra búsqueda informada



## Búsqueda ciega contra búsqueda informada

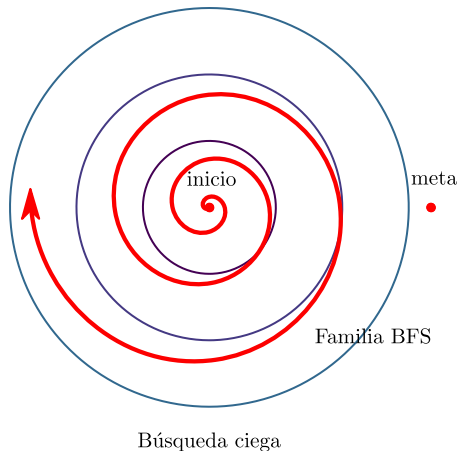


Un algoritmo de la familia de DFS toma una dirección de búsqueda y avanza en la profundidad guiado por esa dirección.

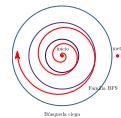
Es muy probable que esa dirección no sea la dirección en la que se encuentra la meta.

De llegar a hacerlo, a esta familia de algoritmos les cuesta mucho tiempo descubrir la meta.

## Búsqueda ciega contra búsqueda informada



### Búsqueda ciega contra búsqueda informada

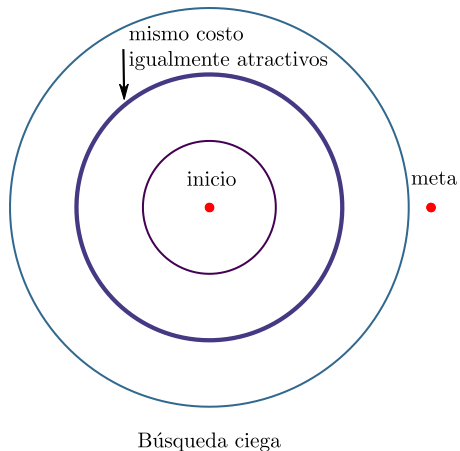


Algoritmos de la Familia BFS realizan exploraciones exhaustivas en todas direcciones.

Estos algoritmos nos dan las soluciones óptimas, pero consumen mucha memoria.



## Búsqueda ciega contra búsqueda informada

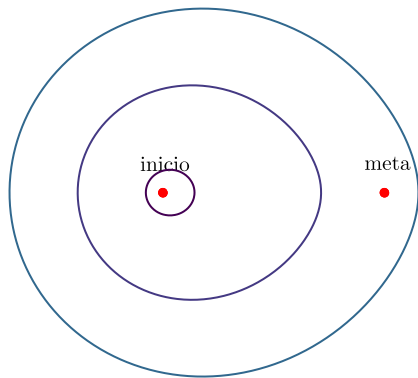


## Búsqueda ciega contra búsqueda informada



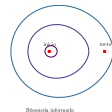
El problema es que desde el punto de vista del agente, todos los nodos a la misma profundidad, o costo en el caso del algoritmo UCS, son igualmente atractivos.

## Búsqueda ciega contra búsqueda informada



Búsqueda informada

### Búsqueda ciega contra búsqueda informada



Búsqueda ciega

La idea principal de los algoritmos informados es dotarlos con información adicional que pueda guiarlos más rápido a encontrar la meta.

En otras palabras, el objetivo es crear un sesgo en las decisiones del agente que lo orienten hacia la meta.

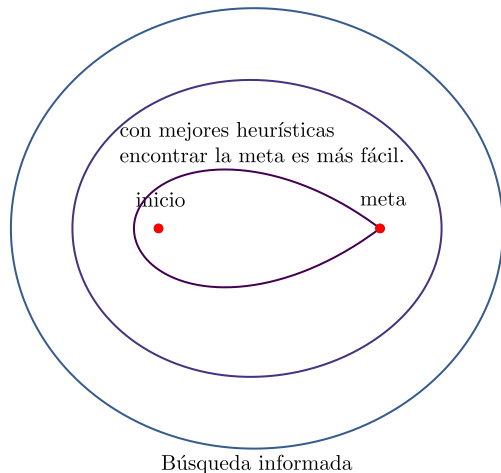
Esto se logra a través de *funciones heurísticas*.

En este diagrama las curvas ilustran puntos que son igualmente atractivos para el algoritmo.

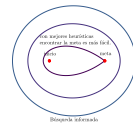
Se construyeron sumando el costo acumulado desde el estado inicial (aquí costo es igual a distancia euclideana) con la distancia al nodo meta multiplicada por un escalar pequeño.

Esta *aproximación* del costo real a la meta sirve para crear un sesgo en la dirección correcta.

# Búsqueda ciega contra búsqueda informada



## Búsqueda ciega contra búsqueda informada



Mejores aproximaciones crean sesgos mas grandes.

Sesgos más grandes son mejores porque permiten a los algoritmos, en especial los tipo BFS, llegar más rápido a la meta, consumiendo con ello menos memoria.

# Funciones heurísticas

## Heurísticas [Judea Pearl 1984]

Las heurísticas son criterios, métodos, o principios para decidir cuál entre múltiples alternativas de líneas de acción promete ser la más efectiva para alcanzar una meta. Representan un balance entre dos requerimientos: la necesidad de hacer los criterios simples y, al mismo tiempo, el deseo de que discriminen correctamente entre elecciones buenas y malas.

## Funciones heurísticas

Para definir las funciones heurísticas citaremos a Judea Pearl. En su libro *Intelligent Search Strategies for Computer Problem Solving*, Pearl las define como criterios discriminatorios para toma de decisiones respecto a líneas de acción en la consecución de una meta.

Algo muy importante a considerar es que deben ser criterios simples.

En los humanos, eso que se conoce como *el sentido común*, *reglas de oro*, *buenas practicas*, etc.

## Ejemplo: Rutas en mapas



## Ejemplo: Rutas en mapas

Consideremos como ejemplo un vehículo autónomo que desea encontrar la ruta óptima para llegar a la Ciudad de Colima desde la Ciudad de México.

Sin mayor información tendría que explorar en todas direcciones, descubriendo las carreteras y sus costos en el camino.



## Ejemplo: Rutas en mapas



└ Ejemplo: Rutas en mapas

Una heurística podría ser implementada con el uso de una brújula. Si el agente ahora sabe que Colima está en el occidente del país y que la Ciudad de México en el centro. La brújula le proporcionaría información valiosa para ir en la dirección correcta.



## Ejemplo: Rutas en mapas



2018-09-27

## Ejemplo: Rutas en mapas

Ejemplo: Rutas en mapas



Aún mejor herística podría definirse utilizando información de un dispositivo GPS.

Este dispositivo le daría la latitud y longitud de cualquier lugar en el que se encuentre.

Si sabemos las coordenadas de Colima, esto puede guiar la búsqueda.

## Ejemplo: Rutas en mapas



## Ejemplo: Rutas en mapas

Una heurística muy buena puede ser construida con la información de coordenadas GPS.

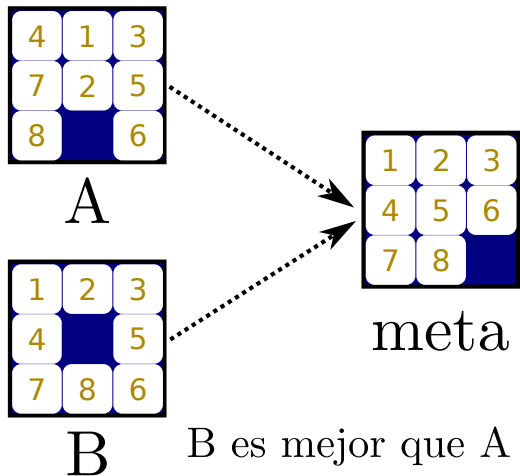
La distancia en línea recta puede servir para crear una preferencia de dirección.

El agente trataría de minimizar la distancia a la meta.

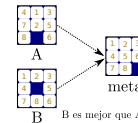




## Ejemplo: el juego del 15

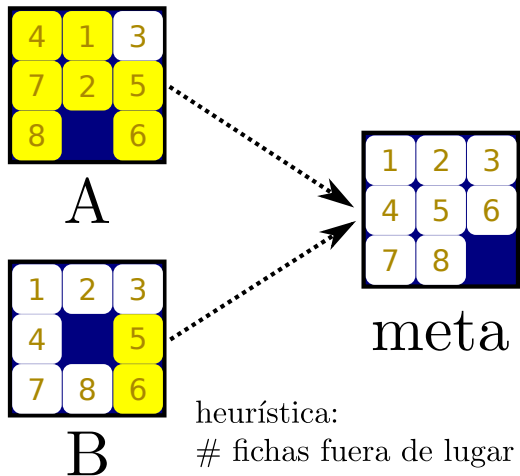


## Ejemplo: el juego del 15

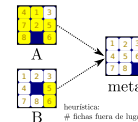


Pensemos ahora en el juego del 15.  
 Si conocemos la configuración de la meta,  
 podemos dar preferencia a una configuración sobre otra dependiendo  
 de qué tan *parecida* es la configuración a la configuración deseada.  
 En este caso la configuración B se parece más que la A.

## Ejemplo: el juego del 15



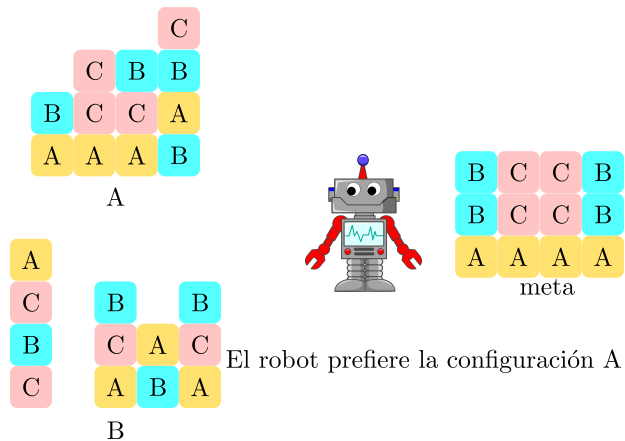
## Ejemplo: el juego del 15



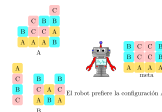
En este caso una función heurística que es informativa para la discriminación de buenas y malas configuraciones esta dada por el número de fichas fuera de su lugar, respecto del objetivo.

Vemos que la configuración A tiene 7 fichas fuera de su lugar, mientras que para la configuración B se tienen sólo 2 fichas fuera de lugar.

## Ejemplo: mundo de bloques



## Ejemplo: mundo de bloques



En este ejemplo, conocido como el mundo de bloques, el robot tiene que mover los bloques hasta conseguir la configuración objetivo.

Es posible definir una heurística que nos permita discriminar estados más cercanos a la meta que otros.

En este caso A es más cercano que B.

## La función heurística

### Costo aproximado a la meta

La función heurística es una estimación del costo que tendrá llegar a la meta desde un nodo particular.

La denotaremos con la letra  $h$ ,  $h : S \rightarrow \mathbb{R}$ .

Estrictamente esta función depende del estado meta, pero no hacemos esta relación explícita.

## La función heurística

Costo aproximado a la meta  
La función heurística es una estimación del costo que tendrá llegar a la meta desde un nodo particular.  
La denotaremos con la letra  $h$ ,  $h : S \rightarrow \mathbb{R}$ .  
Estrictamente esta función depende del estado meta, pero no hacemos esta relación explícita.

Para los algoritmos que presentaremos podemos pensar a la función heurística como una estimación del costo de un estado dado al estado meta.

Como recomienda Pearl, la función debe poder calcularse de manera eficiente y debe ser informativa para la correcta discriminación de estados.