

Métodos Heurísticos en Inteligencia Artificial

Javier Ramírez-Rodríguez
Ana Lilia Laureano-Cruces

Universidad Autónoma Metropolitana

Métodos Heurísticos en Inteligencia Artificial

- Los problemas de Inteligencia Artificial (IA) generalmente usan un término común llamado *estado*.
- No hay un algoritmo formal que los resuelva, *i.e.*, *dado un estado inicial no se puede decir cual es la secuencia de etapas para llegar a un estado meta.*

- La IA estudia especialmente métodos que permiten resolver problemas en los que no existe el conocimiento sistemático para plantear una solución analítica.
- Estos métodos tienen la ventaja de poderse aplicar en una gran diversidad de clases de problemas.
- Para llevarlos a cabo efectiva y eficientemente, sobre dichos métodos se utiliza el conocimiento *heurístico* disponible.
- Entre menos estados se generen, mejor es el algoritmo utilizado para llegar al estado meta.
- Heurístico: Viene del griego *heuriskein* que significa encontrar o descubrir.

- En **Inteligencia Artificial**, **heurístico** se describe mejor como *método de búsqueda*.
- La **búsqueda heurística** consiste en añadir información, basándose en el espacio estudiado hasta ese momento, de forma que se restringe drásticamente esa búsqueda
- Muchos problemas de IA se pueden modelar como una gráfica en donde hay que buscar recorridos o caminos con ciertas características.

- Para estos modelos se han desarrollado muy diversas heurísticas que han dado muy buenos resultados para encontrar soluciones aproximadas.
- Entre las que se pueden mencionar *algoritmos genéticos, recocido simulado y en general algoritmos evolutivos.*

- La IA estudia especialmente métodos que permiten resolver problemas en los que no existe el conocimiento sistemático para plantear una solución analítica (a lo ancho, en profundidad, hill climbing).
 - Estos métodos, considerados débiles por realizar procedimientos de búsqueda **no informada**, no aprovechan la información relevante del dominio, suelen ser poco eficientes, pero tienen la ventaja de poderse aplicar en una gran diversidad de clases de problemas.
 - Para llevarlos a cabo efectiva y eficientemente, sobre dichos métodos se utiliza el conocimiento *heurístico* disponible.
- La clave para el enfoque de **IA es búsqueda inteligente y emparejamiento de estados**. Entre menos estados se generen, mejor es el algoritmo utilizado para llegar al estado meta.

Búsqueda Heurística

- **George Polya**: Estudio de métodos y reglas para descubrimiento e invención.
- Las heurísticas son formalizadas como reglas para elegir (buscar) las ramas en el espacio de estados que son más probables de llegar a una solución aceptable del problema.

En IA, se emplean heurísticas en dos casos

- Un problema puede **no** contar con **solución exacta** debido a ambigüedades inherentes en el problema o datos disponibles.
 - Diagnóstico médico
 - Visión
- Un problema puede tener solución exacta, **pero el costo computacional por encontrarla puede ser prohibitivo**.
 - Problema del agente viajero
 - Problema de coloración
 - Juego de ajedrez

Búsquedas Heurísticas y Sistemas Expertos

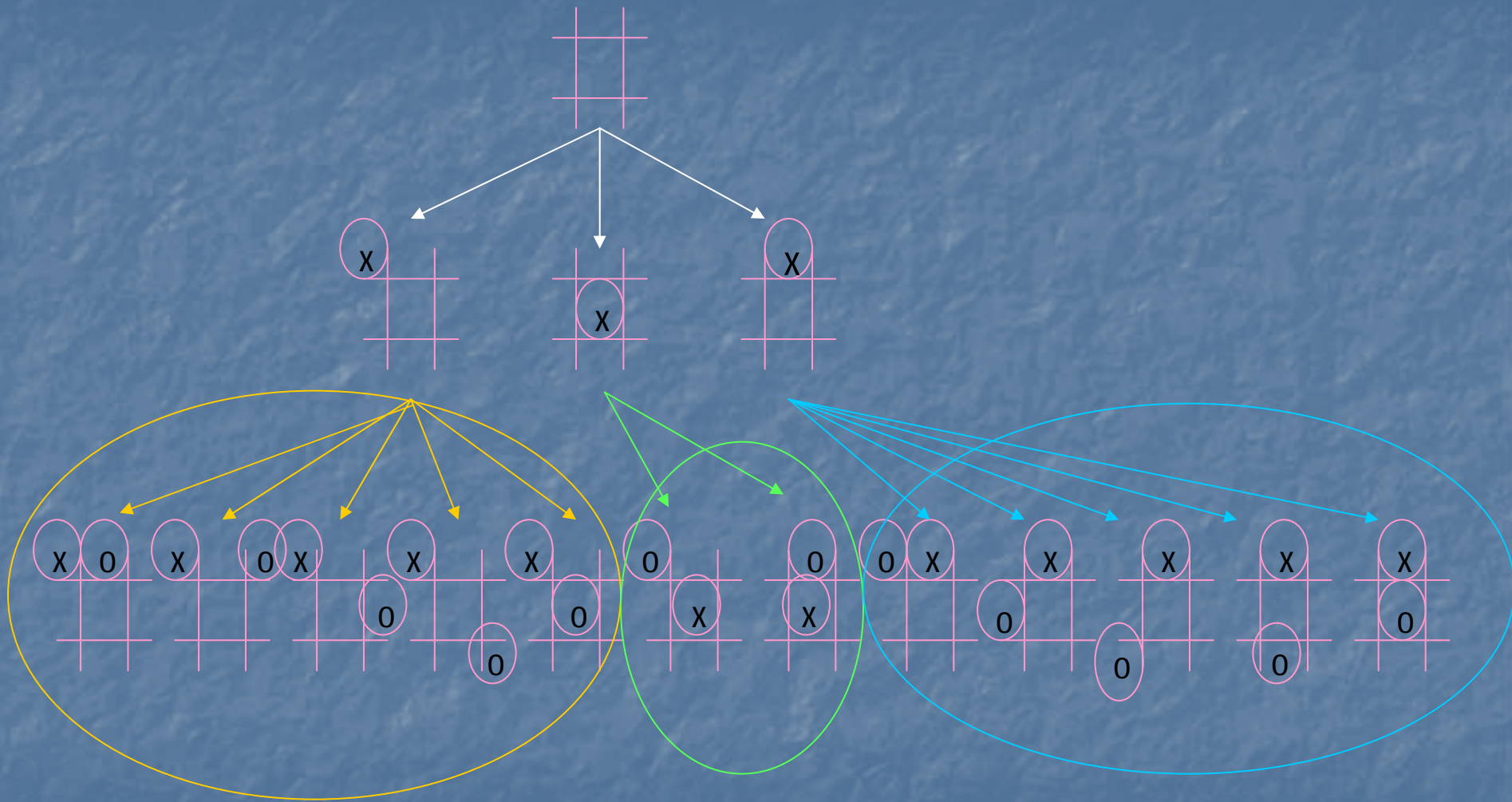
- **Juegos simples** son vehículos ideales para explorar el diseño y comportamiento de algoritmos de búsqueda heurística debido a que
 - El **espacio de búsqueda** es suficientemente grande que requiere **poda heurística**.
 - Son **suficientemente complejos** para sugerir una amplia **variedad de evaluaciones heurísticas**.
 - Generalmente admiten **representaciones sencillas**.
 - Cada nodo del espacio de estados tiene una representación común, una heurística puede aplicarse a través del espacio de estados.

Las Heurísticas

- Atacan la complejidad guiando la búsqueda por los caminos **más prometedores** en el espacio de búsqueda.
- Eliminan estados no prometedores y sus descendientes
- Encuentran soluciones aceptables.
- Desafortunadamente las reglas son falibles, pues usan información limitada.
- **Juegos y probar teoremas** son dos de las más antiguas aplicaciones en IA.
 - No es posible examinar cada inferencia que puede ser hecha en un dominio matemático, o cada movimiento que puede ser hecho en un tablero de ajedrez.
 - De aquí que las funciones heurísticas sean un componente esencial para resolver este tipo de problemas.

Un Algoritmo heurístico consta de **dos partes**

1. La medida heurística.
 2. Un algoritmo que la usa para buscar en el *espacio de estados*.
- Ejemplo: Considerar el **juego del GATO**.
 - Un simple análisis permite determinar el número de estados que necesitan considerarse en una búsqueda exhaustiva en $9 \times 8 \times 7 \times \dots \circ 9!$
 - Una **reducción simétrica** puede disminuir un poco el espacio de búsqueda.
 - Tomando en consideración el punto anterior, sólo hay **3 movimientos iniciales**.



Razonamiento

- Si usamos reducciones por simetría, en un segundo **nivel** de estados el número de posibles caminos a través del espacio de estados se convierte en **$12 \times 7!$**
- Una heurística simple casi puede eliminar la búsqueda total, si se intenta *colocar a X* sólo a estados donde esta tiene más posibilidades de ganar.
 - El algoritmo selecciona y se mueve al estado **con el valor más alto de la heurística**, con lo que se podan hasta **$2/3$ del espacio de estados**.
 - La **heurística se utilizará en cada turno de X**.

Algoritmos para búsqueda heurística

- La forma más simple de implementar una búsqueda heurística es mediante el procedimiento llamado **Hill climbing (escalando una colina)**:
 - ✓ no se corrigen fallas en la estrategia
 - ✓ tendencia a caer en óptimo local (**Hill climbing-RecSimul**).
- Mejor primera búsqueda (best-first-search) (MBH)
 - Usar listas para mantener estados: **abiertos y cerrados (algoritmo A*)**.
 - ✓ Etapa adicional de este ultimo, es ordenar los estados en **abiertos** de acuerdo a alguna estimación heurística de su **cercanía** al objetivo.
 - ✓ En cada iteración se considera el estado más prometedor de la lista de **abiertos**
 - ✓ Actualizando la historia de los ancestros de nodos en **abierto** y **cerrado** cuando se redescubren es más probable que el algoritmo encuentre el camino más corto al objetivo.
 - ✓ Cuando **abierto** se mantiene como una lista clasificada se llama *fila prioritaria*.

FUNCIONES DE EVALUACIÓN HEURÍSTICA

- A continuación se evalúa el desempeño de varias heurísticas para el rompecabezas de las 9 celdas y 8 azulejos:
 - Piezas fuera de lugar, *con respecto al estado meta*.
 - Suma de distancias fuera de lugar, *con respecto al estado meta*.
 - Número de regresos directos por 2. *Con respecto al estado meta*

<table border="1"> <tr><td>2</td><td>8</td><td>3</td></tr> <tr><td>1</td><td>6</td><td>4</td></tr> <tr><td></td><td>7</td><td>5</td></tr> </table>	2	8	3	1	6	4		7	5	5	6	0
2	8	3										
1	6	4										
	7	5										
<table border="1"> <tr><td>2</td><td>8</td><td>3</td></tr> <tr><td>1</td><td></td><td>4</td></tr> <tr><td>7</td><td>6</td><td>5</td></tr> </table>	2	8	3	1		4	7	6	5	3	4	0
2	8	3										
1		4										
7	6	5										
<table border="1"> <tr><td>2</td><td>8</td><td>3</td></tr> <tr><td>1</td><td>6</td><td>4</td></tr> <tr><td>7</td><td>5</td><td></td></tr> </table>	2	8	3	1	6	4	7	5		5	6	0
2	8	3										
1	6	4										
7	5											
Heurísticas	(1) Piezas fuera de lugar	(2) Suma de distancias fuera de lugar	(3) 2 x no. Regresos directos									

Meta

1	2	3
8		4
7	6	5

Ejemplo de la contabilización en un regreso directo

2	1	3
8		4
7	5	6

1	2	3
8		4
7	6	5

Un estado con una meta y *dos regresos directos*: 1 y 2, 5 y 6

ejemplo

2	8	3
1	6	4
7		5



2	8	3
1	6	4
	7	5

2	8	3
1		4
7	6	5

2	8	3
1	6	4
7	5	

meta

1	2	3
8		4
7	6	5

$g(n) = 0$

2	8	3
1	6	4
7		5

$g(n) = 1$

2	8	3
1	6	4
	7	5

2	8	3
1		4
7	6	5

2	8	3
1	6	4
7	5	

Valores de $h(n)$

6

4

6

1	2	3
8		4
7	6	5

$f(n) = g(n) + h(n)$, donde:

$g(n)$ = distancia de n al estado inicial (no. de niveles)

$h(n)$ = número de distancias fuera de lugar / heurística (2)

2	8	3
1	6	4
7		5

1	2	3
8		4
7	6	5

$h(n) = 6 \quad 4 \quad 6$



2	8	3
1	6	4
	7	5

2	8	3
1		4
7	6	5

2	8	3
1	6	4
7	5	

$h(n) = 5 \quad 3 \quad 5$



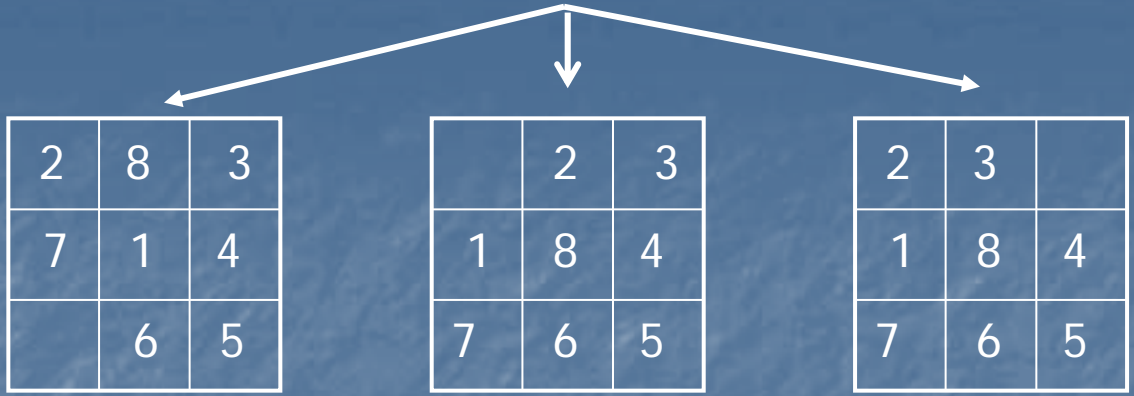
2	8	3
	1	4
7	6	5

2		3
1	8	4
7	6	5

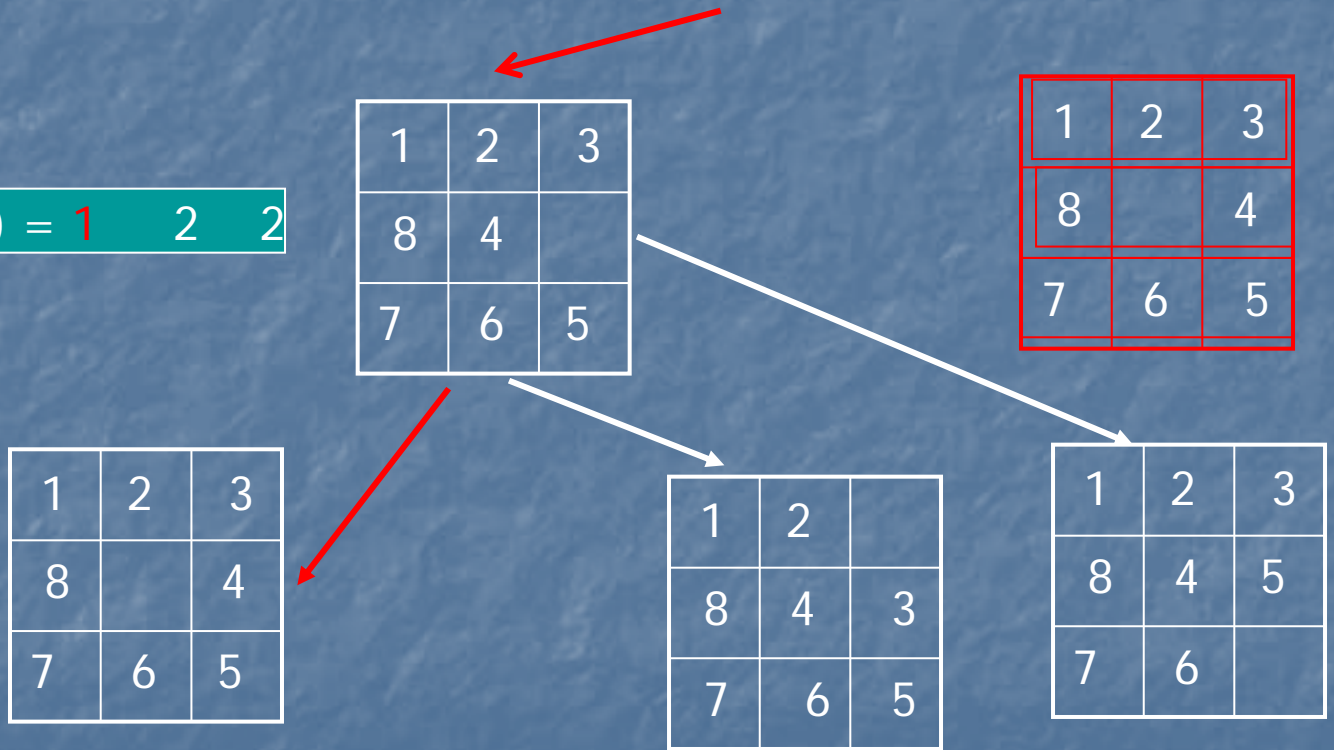
2	8	3
1	4	
7	6	5



$h(n) = 6 \quad 2 \quad 3$



$h(n) = 1 \quad 2 \quad 2$



Análisis de las heurísticas

1. La heurística *más simple* cuenta los azulejos que están fuera de lugar, cuando se compara con el estado meta.
2. Una heurística *mejor* sería la que suma todas las distancias por las cuales los azulejos están fuera de lugar.

Ambas desconocen las dificultades para intercambiar azulejos.

3. Una heurística que toma en cuenta esta dificultad, es la que utiliza un *factor de cambio*, cuando multiplica por un número pequeño; *en este caso 2*, cada regreso directo de un azulejo. Esto es, cuando *dos azulejos adyacentes deben intercambiarse* (en cualquier sentido) para estar en el orden del objetivo.

Dificultad para crear buenas heurísticas

- El objetivo es usar la información limitada disponible para hacer selecciones inteligentes.
- El diseño de buenas heurísticas es un problema empírico, que se apoya en *buen juicio e intuición*, pero la medida final de una heurística debe ser el *desempeño* en el *problema en cuestión para el que fue creado*.
- Y debe comprender los dos datos siguientes, en la *función de evaluación*:
 - $f(n) = g(n) + h(n)$
 - donde:
 - $g(n)$: representa la longitud del estado n al estado inicial y
 - $h(n)$: representa la estimación heurística de la distancia de n al objetivo

Puntos a considerar en el diseño de funciones heurísticas

- En una función heurística no solo existe el deseo de encontrar una solución sino también encontrar el camino más corto a una meta.
 - Esto es importante cuando en una aplicación las etapas extras tienen un *costo excesivo*, tal como la planificación de un camino para un robot autónomo a través de *ambientes peligrosos*.
- **Admisible**
 - Se denomina a la función heurística que encuentra el camino más corto a una meta siempre que exista. Por ejemplo:
 - El método de búsqueda a lo ancho es una estrategia *admisible*, desgraciadamente, en general, es ineficiente para usarse en la práctica.

Al utilizar funciones heurísticas

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

- Se pueden caracterizar una clase de estrategias de *búsqueda admisible*.
- Determinando las propiedades de heurísticas admisibles se define la función de evaluación $f^*(n) = g^*(n) + h^*(n)$
- Donde $g^*(n)$ es el camino más corto del nodo inicial *al* nodo n y h^* representa el costo del camino más corto *desde* n *a* la *meta*.
- Una función heurística h_1 es *más informada* que otra h_2 si ambas son admisibles y además.
 - $h_1(n) \leq h_2(n) \quad \forall n \mid n, \text{ donde } n, \text{ no es un nodo meta.}$

Monotonía

- Se dice que una función heurística es monótona ssi:

1. Para todos los estados n_i y n_j , se cumple que:

■ $h(n_i) - h(n_j) \leq \text{costo}(n_i, n_j)$

2. $H(\text{meta}) = 0$

- n_j es *descendiente* de n_i
- *Costo* (n_i, n_j) es el costo actual, en número de movimientos, para ir del estado n_i al estado n_j

- Una manera de describir la propiedad de monotonía es que la heurística es admisible en todas partes.

FIN