Búsqueda Heurística VII

Pedro Meseguer
IIIA-CSIC
Bellaterra, Spain
pedro@iiia.csic.es

Búsqueda en tiempo real

- Búsqueda clásica (*off-line search*):

Búsqueda de una solución completa	Ejecución de la solución

- No sirve para tareas que
  - exigen respuestas en tiempo limitado:
    - robot
    - puzzle de 35

    no podemos esperar
  - tienen información limitada:
    - robot explorador

    no podemos calcular

Búsqueda en tiempo real

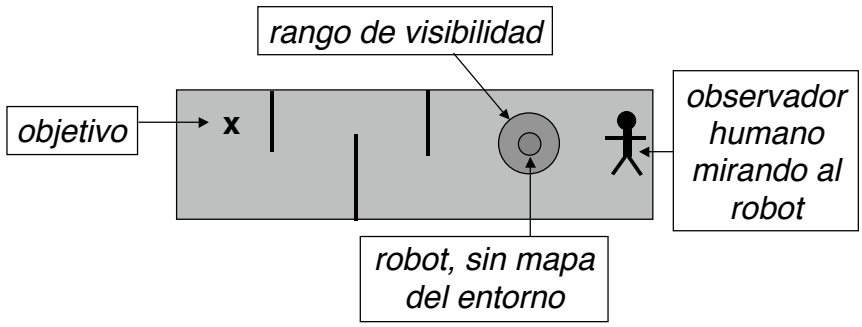
IDEA:

- calcular un trocito de la solución / ejecutarlo
- se alterna búsqueda y ejecución (*on-line search*)

- La solución óptima NO se encuentra la primera vez
- Se espera que la calidad de la solución mejore al repetir la resolución de ese problema concreto

Problemas para búsqueda en tiempo real

- Respuestas críticas en tiempo:



el robot ha de comenzar a moverse en un tiempo corto, si no el observador creerá que no funciona

Problemas para búsqueda en tiempo real

Puzzle de 35

1	2	3	4	5	6
7		8	9	10	11
12	13	14	15	16	17
18	19	20	21	22	23
24	25	26	27	28	29
30	31	32	33	34	35

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	

Espacio de estados: 36!
Demasiado grande para pretender calcular la solución óptima (gran cantidad de tiempo)
Alternativa: búsqueda en tiempo real

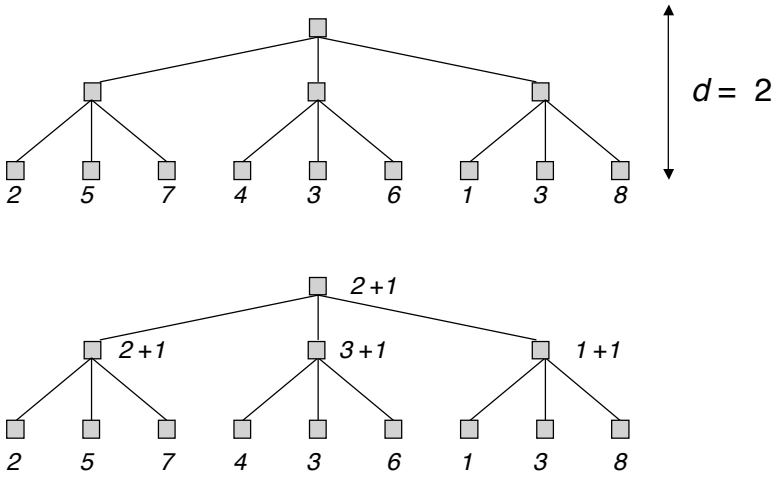
Movimiento

- Dado un estado x: ¿cuál es el siguiente estado y?
- IDEA: escoger el mejor de los sucesores

$$y \leftarrow \arg\min \{ c(x,z) + h(z) \}, \qquad z \in succ(x)$$

- IDEA: para calcular $h(z)$, podemos hacer anticipación
  - Se busca a profundidad limitada
  - Función de evaluación en la frontera — *minimin*
  - Los valores *mínimos* se propagan hacia arriba

Ejemplo

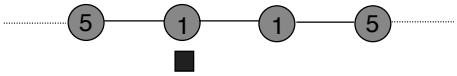


De un estado a la secuencia

- MiniMin: nos da la primera acción

  estado actual → un sucesor

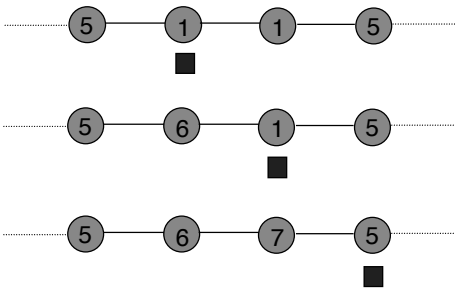
- ¿Cómo encontrar la secuencia solución?
  - Aplicar Minimin al nuevo estado:



  - No funciona, se entra en ciclo
  - Se ha de modificar la *h* del estado actual

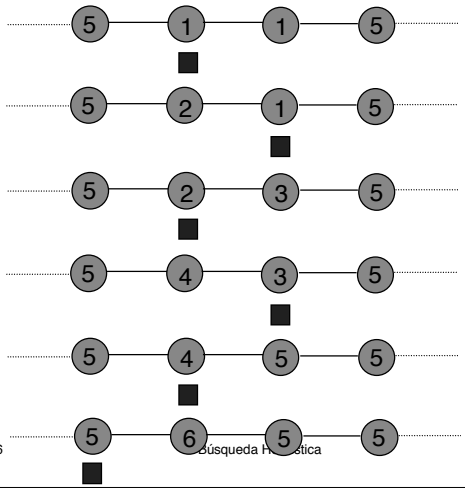
Dos formas de actualizar

- RTA*: $h(x) \leftarrow \max\{h(x), \text{2nd min } c(x,y)+h(y)\}$



Dos formas de actualizar

- LRTA*: $h(x) \leftarrow \max\{h(x), \text{min } c(x,y)+h(y)\}$



RTA*

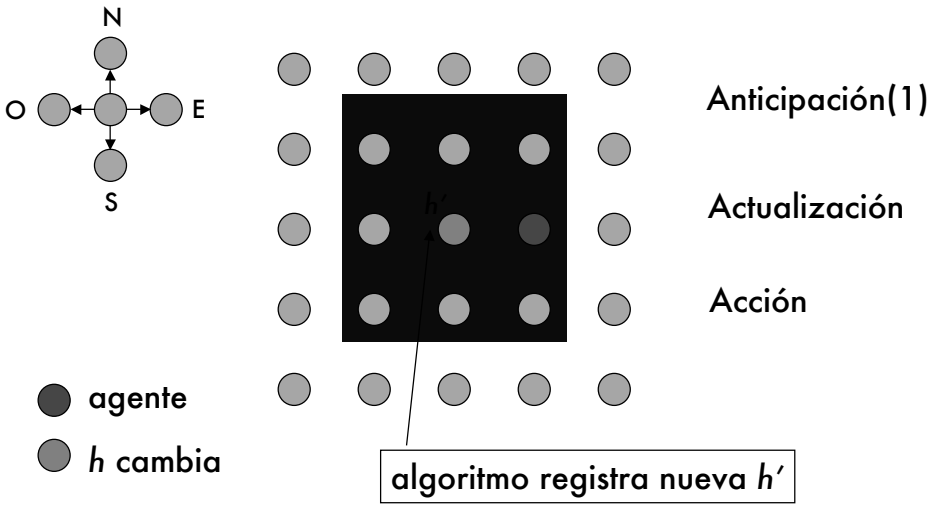
- Función heurística:
  - $f(n) = g(n) + h(n)$
  - $g(n)$: coste desde el nodo actual hasta n
- RTA*:
  - Algoritmo del tipo *primero el mejor*
  - Heuristica diferente del A*
  - Actualiza la h del nodo actual, para evitar ciclos

Algoritmo RTA* / LRTA*

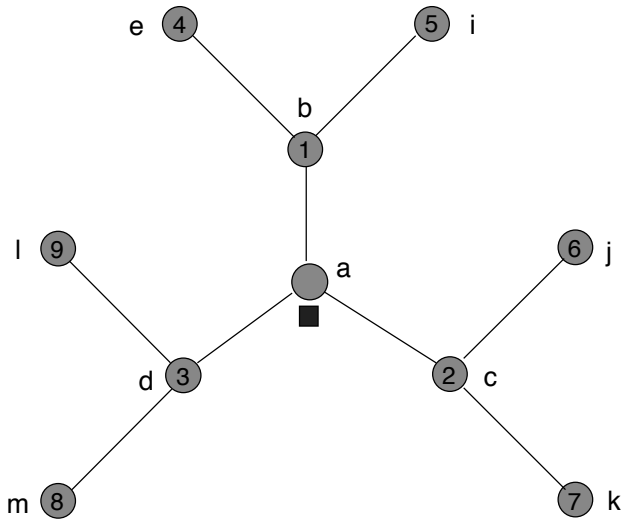
- Tabla T: h de nodos ya expandidos
- Bucle:
  - Generar sucesores del estado actual
  - Si un sucesor ya está en T, usar su h de T
  - Sino, calcular f de sucesores por Minimin
  - Almacenar estado actual en T con

    h = 2do min f / min f (LRTA*)
  - Acción: ir al mejor sucesor (min f)

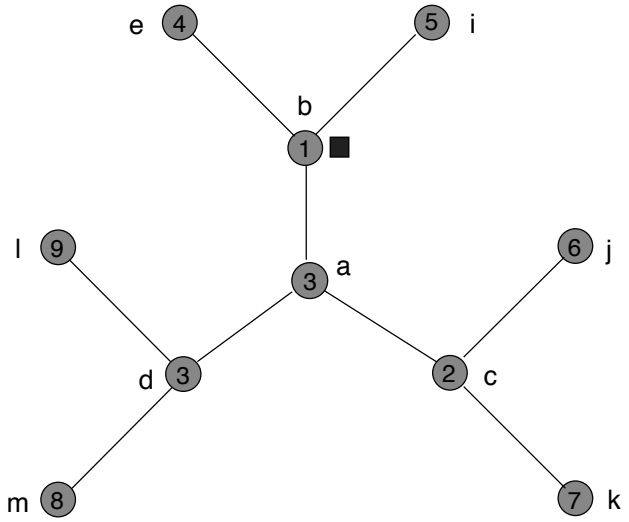
Ejecución



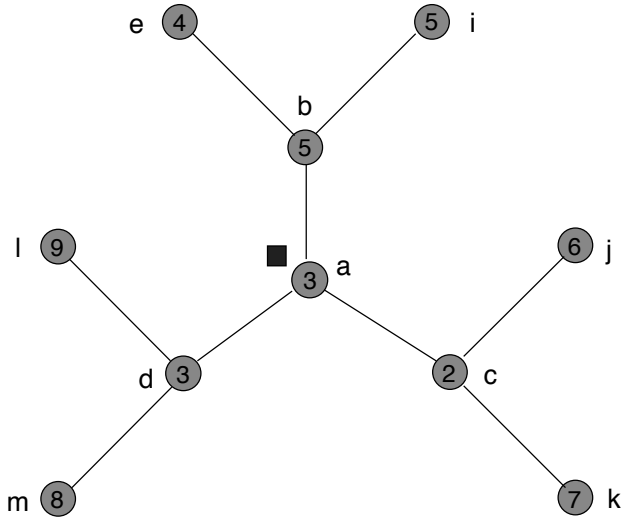
RTA*: ejemplo



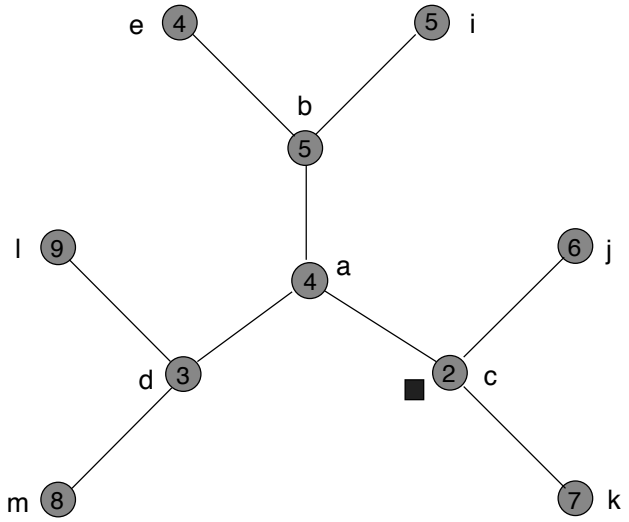
RTA*: ejemplo



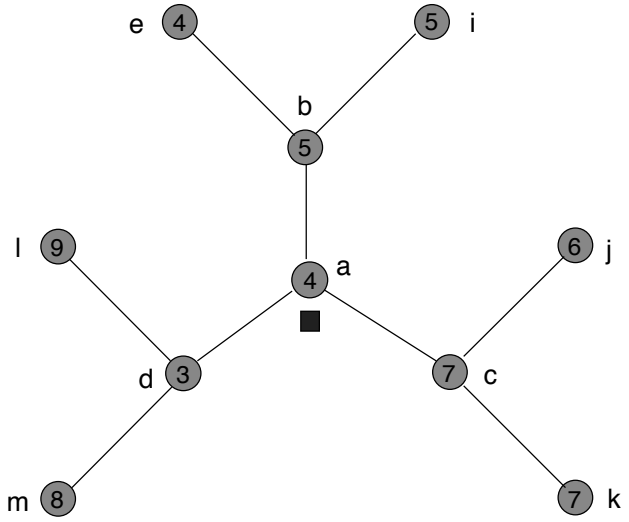
RTA*: ejemplo



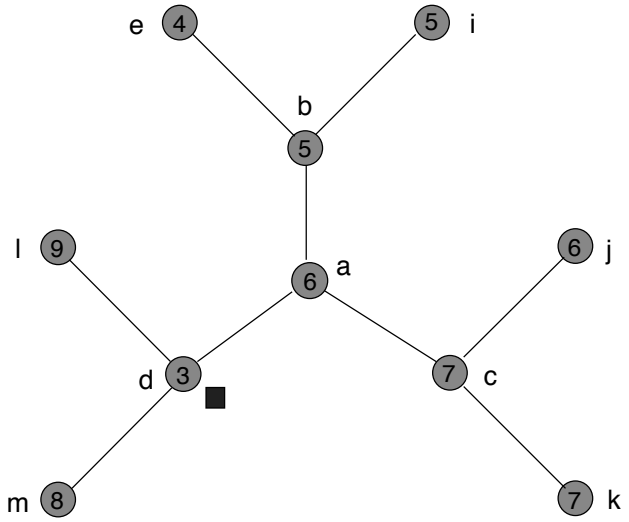
RTA*: ejemplo



RTA*: ejemplo



RTA*: ejemplo



Completitud

- Completitud RTA*/LTRA*: ambos son completos en
  - espacios de estados finitos
  - valores heurísticos finitos
  - solución alcanzable desde cada estado
- Primera ejecución:
  - RTA* es más rápido que LRTA*
- Ejecuciones repetidas sobre el mismo ejemplar:
  - LRTA* converge a caminos óptimos si *h* admisible
  - RTA* no converge a caminos óptimos:
    h se puede volver no admisible

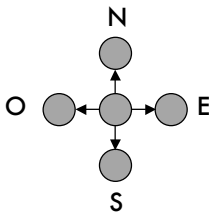
Convergencia

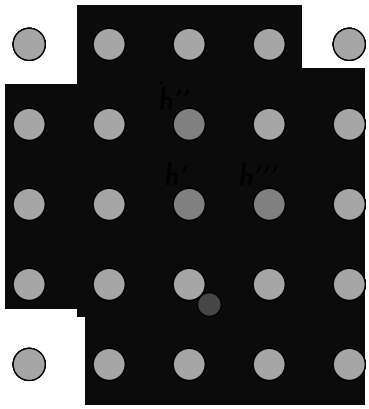
- Al resolver un problema: $H_0 \rightarrow H_1$,
  matriz de valores heurísticos de estados de la búsqueda
- Al resolver repetidamente el mismo ejemplar:
  - resolución *i*-ésima
  - toma como entrada H_{i-1}, los valores heurísticos finales de la resolución *i*-1
- LRTA* converge a caminos óptimos (*h* admisible):
  - a partir de un punto, sólo recorre caminos óptimos

LRTA*(*k*)

- LRTA(*k*) = LRTA* + propagación de cambios de la heurística hasta *k* estados
- Diferencias:
  - RTA* / LRTA*: *1 actualización por acción*
  - LRTA(*k*): *k actualizaciones por acción*
  - *Más aprendizaje*
  - *Mejor* calidad de la heurística pero *mayor* tiempo de búsqueda por nodo

Propagación de cambios





agente

h cambia

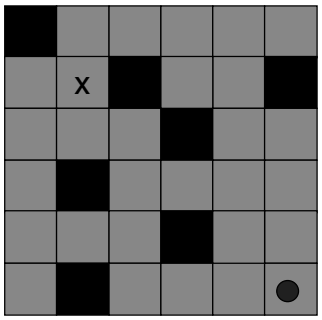
Anticipación(1)
Actualización
Anticipación(1)
Actualización
Anticipación(1)
Actualización
Anticipación(1)
Actualización
Anticipación(1)
Actualización
.....
Acción

En la práctica

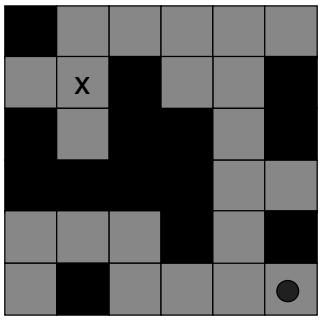
- LRTA*(k):
  - Pasos de búsqueda más largos
  - Mejora importante del rendimiento:
    - Primera ejecución
    - Convergencia
    - Estabilidad de la solución
  - Beneficios dependen del parámetro *k*: cuanto mayor es *k,* más beneficios, con el coste extra de tener pasos de búsqueda más largos

Problemas de prueba

Grid



Laberinto



Movimiento: una casilla N, S, E. O

Solución: secuencia de movimientos que llevan a x

Problemas de prueba

- **Grid35**. Retículo de tamaño 301x301 con un 35% de obstáculos colocados al azar.
- **Grid70**. Retículo de tamaño 301x301 con un 70% de obstáculos colocados al azar.
- **Maze**. Laberíntos acíclicos de tamaño 181x181 cuya estructura de salida se generó por búsqueda en profundidad.

Primera ejecución

	Grid 35		Grid 70		Maze	
	Coste	Tiempo/ Paso	Coste	Tiempo/ Paso	Coste	Tiempo/ Paso
LRTA*	100%	100%	100%	100%	100%	100%
LRTA* (k=15)	17%	237%	18%	222%	16%	231%
RTA*	45%	91%	29%	97%	6%	96%

Convergencia

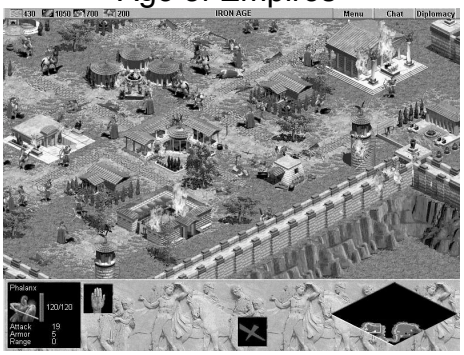
	Grid 35			Grid 70			Maze		
	coste	repeti ciones	tiemp/ paso	coste	repeti ciones	tiemp/ paso	coste	repeti ciones	tiemp/ paso
LRTA	100%	100%	100%	100%	100%	100%	100%	100%	100%
LRTA k=15	36%	48%	189%	27%	37%	179%	23%	24%	232%

Resumen

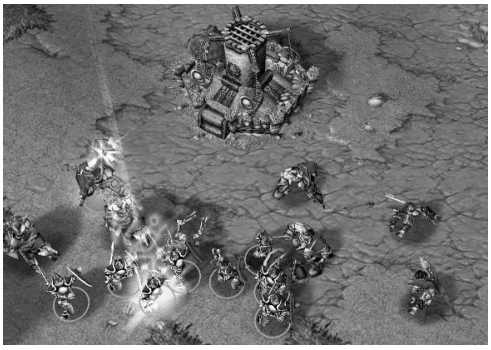
- Búsqueda clásica: (*off-line search*) no es adecuada para ciertos tipos de problemas
- Búsqueda en tiempo real: (*on-line search*)
  - Buscar la siguiente acción
  - Ejecutarla
- Siguiente estado: Minimin
- Secuencia solución: incluir $g(n)$, actualizar $h(actual)$
  - RTA*: completo pero no converge a solución óptima
  - LRTA*: completo, si h admisible converge sol óptima
  - LRTA*(k): como LRTA*, propaga cambios heurísticos

Aplicaciones de búsqueda en tiempo real

Age of Empires



Warcraft III



Agentes:

tiempo e información limitada

han de mejorar el rendimiento con la experiencia