

Clase de Inteligencia Artificial: Proyecto de Búsqueda

Profesores:

Luis A. Pineda Cortés, IIMAS, UNAM
Arturo Rodríguez García, PCIC, UNAM

Octubre 22, 2015

Condiciones de Entrega

Formato: Código fuente dentro de una carpeta comprimida llamada *Proyecto2.zip*. Archivo *doc.pdf* con la documentación del proyecto.

Fecha y modo de entrega: Enviar el código y la documentación por correo electrónico a más tardar el 26 de noviembre a las 11:00 am. Enviar correo con asunto *Proyecto de Búsqueda* y archivos adjuntos a lpineda@unam.mx y arturo8602@yahoo.com.mx.

Entregar una copia impresa de la documentación justo al inicio de la clase del 26 de noviembre.

Descripción

Un robot mesero se encuentra en una fiesta para atender a los invitados, llevando los alimentos solicitados hasta la mesa de los comensales.

Sea $L = \{l_1, l_2, \dots, l_n\}$ el conjunto de locaciones de interés para el robot, que incluyen su posición inicial, la ubicación de los estantes que contienen comida, y la ubicación de las mesas en las que los invitados esperan los productos solicitados. Sea $O = \{o_1, o_2, \dots, o_m\}$ el conjunto de objetos que pueden ser solicitados por los comensales. El robot conoce en cuál estante se encuentra cada uno de los objetos. El robot posee dos brazos (izquierdo y derecho), y cada brazo puede cargar sólo un objeto a la vez.

La tarea inicia con el robot en la posición inicial, con sus dos brazos libres. Un humano llega y le da una instrucción en la que solicita realizar k entregas de objetos a distintas locaciones dentro de un tiempo límite t_{max} en segundos. Cada entrega es descrita de la forma “Lleva el objeto o_i a la locación l_j ”. Al terminar de recibir la orden, un cronómetro empieza a medir el tiempo desde cero.

El robot debe planear la secuencia de acciones a realizar para cumplir el mayor número de entregas dentro del tiempo límite. Las acciones que puede realizar el robot son:

- $mover(l_i, l_j)$.- moverse de una locación $l_i \in L$ a una locación $l_j \in L$.
- $buscar(o_i)$.- buscar un objeto $o_i \in O$ con el sistema visual en la locación actual. Esta acción sólo puede tener éxito si en la locación actual se encuentra el objeto buscado.

- *agarrar*(o_i) .- agarrar un objeto $o_i \in O$ en la locación actual usando un brazo libre. El objeto debió haber sido identificado en la acción anterior (mediante *buscar*(o_i))
- *colocar*(o_i) .- colocar un objeto $o_i \in O$ en la locación actual. El robot debe tener el objeto en uno de sus brazos, y la locación actual debe corresponder al lugar en el que se pidió entregar el objeto.

Cada acción a tiene asociada una recompensa $r(a)$, una probabilidad $p(a)$ de éxito y un costo $c(a)$ que corresponde al tiempo promedio de ejecución en segundos. En el caso de las recompensas, no se aplican si la misma acción con los mismos parámetros ya fue realizada anteriormente, o si su aplicación no tiene utilidad para resolver alguna de las entregas solicitadas. Elegir la mejor solución implica considerar:

- Que la suma de las recompensas sea lo más alta posible.
- Que la suma de los tiempos sea lo más baja posible, y además no exceda el tiempo límite.
- Que la probabilidad de ejecutar la serie de acciones elegida sea lo más alta posible.
- Que la secuencia de acciones tome en cuenta las restricciones mencionadas en la descripción del proyecto (por ejemplo, considerar que el robot tiene dos brazos, que cada brazo puede cargar un objeto a la vez, etc.).

El sistema debe tener una fase de planeación, en la que se considera que el robot se encuentra en el estado inicial, recibe una orden y determina la mejor secuencia de acciones para cumplir lo solicitado. También debe contar con una fase de ejecución, en la cual el robot ejecuta el plan acción por acción. Para realizar esta simulación, deberás incluir un generador de números aleatorios que determine si una acción se cumplió o no. En caso de que una acción falle, el robot debe intentar continuar con el plan original o replanear la serie de acciones a partir de donde se quedó. La simulación debe considerar el tiempo de procesamiento ocupado por la planeación. En tu documentación deberás incluir:

- Una explicación clara del algoritmo de búsqueda que utilizaste.
- Una explicación breve de cómo utilizar tu sistema.
- Evidencia empírica del funcionamiento de tu sistema. Incluye casos interesantes que demuestren que tu implementación funciona correctamente con distintos datos de prueba.

Observaciones:

- Tu sistema debe funcionar con distintos tipos de datos que incluyen variaciones en el número de objetos y locaciones, así como las funciones de recompensa, probabilidad y costo de las acciones. Toda esta información debe estar almacenada en la base de conocimiento creada en el Proyecto 1 de este curso.
- Como parte de la evaluación, en la clase del 26 de noviembre cada equipo tendrá que explicar brevemente su proyecto, y se le pedirá que realice algunas pruebas para demostrar su funcionamiento.
- Trabajar en equipos de dos personas.
- Utilizar SWI Prolog versión 7.1.20 o superior.

Ejemplo

Imaginemos que el robot se encuentra en una fiesta en la que hay un estante de comida y un estante de bebidas. En el estante de comida se ofrecen hamburguesas y sandwiches. En el estante de bebidas se ofrecen latas de refresco, botellas de agua y vasos de café. Además, hay tres mesas donde están los invitados. La Figura 1 muestra la distribución de las locaciones en el mapa del evento. El robot debe tener en su base de conocimiento los siguientes datos:

- $L = \{inicio, estante1, estante2, mesa1, mesa2, mesa3\}$
- $O = \{hamburguesa, sandwich, refresco, agua, cafe\}$

La Tabla 1 contiene la ubicación de los objetos. La Tabla 2 muestra la función de probabilidad de la acción moverse. La Tabla 3 incluye las funciones de probabilidad de las acciones buscar, agarrar y entregar. La Tabla 4 corresponde al costo en segundos de la acción moverse. La Tabla 5 presenta el costo en segundo de las acciones buscar, agarrar y entregar, mientras que la Tabla 6 contiene sus recompensas. La recompensa para la acción mover es cero en este ejemplo.

Si el comando que recibe el robot es “Lleva un refresco a la mesa 2, un sandwich a la mesa 3 y agua a la mesa 1 en menos de 60 segundos” una posible secuencia de acciones es la siguiente:

- Moverse a estante 2.
- Buscar refresco.
- Agarrar refresco.
- Buscar agua.
- Agarrar agua.
- Moverse a mesa 1.
- Entregar agua.
- Moverse a mesa 2.
- Entregar refresco.

Esta secuencia sólo logra resolver dos de las entregas solicitadas. El tiempo total para ejecutar esta secuencia de acciones es: $8 + 2 + 2 + 15 + 10 + 6 + 10 + 2 + 3 = 58$ segundos. La probabilidad de ejecutar la secuencia de acciones es: $(0.98)(0.96)(0.96)(0.70)(0.95)(0.98)(0.99)(0.99)(0.99) = 0.5711$. La recompensa es: $0 + 50 + 80 + 50 + 80 + 0 + 300 + 0 + 300 = 860$ puntos.

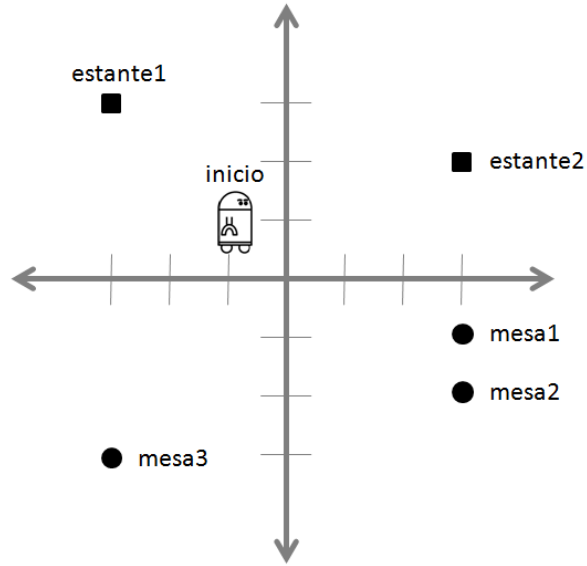


Figura 1: Mapa del evento

Objeto	Locación
hamburguesa	estante1
sandwich	estante1
refresco	estante2
agua	estante2
cafe	estante2

Tabla 1: Ubicación de los objetos

	<i>inicio</i>	<i>estante1</i>	<i>estante2</i>	<i>mesa1</i>	<i>mesa2</i>	<i>mesa3</i>
inicio	1.00	0.98	0.98	0.98	0.98	0.98
estante1	0.98	1.00	0.97	0.97	0.96	0.97
estante2	0.98	0.97	1.00	0.98	0.98	0.96
mesa1	0.98	0.97	0.98	1.00	0.99	0.97
mesa2	0.98	0.96	0.98	0.99	1.00	0.97
mesa3	0.98	0.97	0.96	0.97	0.97	1.00

Tabla 2: Función de probabilidad de la acción moverse

x	$buscar(x)$	$agarrar(x)$	$entregar(x)$
hamburguesa	0.95	0.90	0.99
sandwich	0.80	0.85	0.99
refresco	0.96	0.95	0.99
agua	0.70	0.95	0.99
cafe	0.85	0.70	0.85

Tabla 3: Funciones de probabilidad de las acciones buscar, agarrar y entregar

	<i>inicio</i>	<i>estante1</i>	<i>estante2</i>	<i>mesa1</i>	<i>mesa2</i>	<i>mesa3</i>
inicio	0	6	8	9	10	9
estante1	6	0	12	15	16	12
estante2	8	12	0	6	8	16
mesa1	9	15	6	0	2	13
mesa2	10	16	8	2	0	12
mesa3	9	12	16	13	12	0

Tabla 4: Función de costo de la acción moverse

x	$buscar(x)$	$agarrar(x)$	$entregar(x)$
hamburguesa	5	8	5
sandwich	5	9	5
refresco	2	2	3
agua	15	10	10
cafe	10	20	20

Tabla 5: Funciones de costo de las acciones buscar, agarrar y entregar

x	$buscar(x)$	$agarrar(x)$	$entregar(x)$
hamburguesa	50	100	300
sandwich	50	100	300
refresco	50	80	300
agua	50	80	300
cafe	50	150	400

Tabla 6: Funciones de recompensa de las acciones buscar, agarrar y entregar