

Problema 1

Ejercicio 1: Modela el agente usando la formalización REAS. (0.5p)

Rendimiento: Tiempo de llegada al destino.

Entorno:

- Tablero de 8x8 casillas con casillas de diferente tipo. Las podemos codificar con un numero entero de forma que las casillas con valor 0 son las casillas normales, las 1 las negras, las 2 las verdes y las 3 las rosas
- A la vez tenemos un vector de floats que nos dice el coste de movernos por cada una de ellas.
 - o 0 = 1
 - o 1 = infinito
 - o 2 = 2
 - o 3 = 0.5

Actuadores:

- Moverse arriba
- Moverse abajo
- Moverse izquierda
- Moverse derecha
- Enviar alerta

Sensores:

- Sensor de casillas negras
- Sensor de casillas verdes
- Sensor de casillas rosas
- Sensor de salida

Ejercicio 2: ¿Qué tipo de agente es? Razona la respuesta

Agente deliberativo basado en objetivos. La razón es que no solo necesitamos memoria (lo cual excluye el reactivo simple) si no que además, necesitamos predecir el comportamiento futuro. No tenemos claro la forma de encontrar la solución al problema y el enunciado no nos dice como hacerlo, porque no hay una receta para hacerlo. Hay que buscar una solución y generar un plan para llevarlo a cabo.

Ejercicio 3: Modela el agente con el formalismo que consideres más adecuado. Este debe garantizar que siempre encuentre el mejor camino

La propia pregunta te limita deja pocas opciones, solo hemos visto en clase dos algoritmos que son óptimos: A* y sus derivados y búsqueda en anchura. Elegir el segundo solo es viable si no tenemos información heurística, pero aquí sabemos cosas del entorno, así que usaremos A*

Para formalizar un problema como problema de búsqueda, necesitamos:

Estado inicial: agente en la posición 0,1 y salida en la posición 6,5

Estado final: agente en la posición 6,5

Operadores:

- moverIzquierda(x,y) = x-1,y => comprobar que $x \geq 0$
- moverDerecha(x,y) = x+1,y => comprobar que $x < 8$
- moverArriba(x,y) = x,y-1 => comprobar que $y \geq 0$
- moverAbajo(x,y) = x,y+1 => comprobar que $y < 8$

Dado un estado, no se generarán sucesores si este lleva a una casilla de tipo 1.

Heurística:

La distancia de manhattan pero, asumiendo que todas las casillas cuestan 0.5, para asegurarnos que la heurística es admisible. Si elegimos el valor de 1, podría pasar que un camino fuese mas corto de lo esperado. Estamos relajando dos restricciones, la de los costes de las casillas más caras, las blancas y las verdes y la imposibilidad de transitar por las negras.

Distancia euclídea también valdría, pero es menos informada.

Problema 2

Ejercicio 1: ¿Qué tipo de agente es? Razona la respuesta. (0.5p)

Es un agente deliberativo basado en objetivos. No solo necesitamos memoria, necesitamos planificar y buscar un resultado lo más optimo posible. Nótese que no se exige optimalidad, por lo que podemos aplicar cualquier algoritmo sub-optimo como búsqueda voraz, aprendizaje por refuerzo, etc.

Ejercicio 2: Modela el agente para que resuelva el problema con el algoritmo que consideres más adecuado (1,5p).

Se podría resolver de múltiples formas, pero en un problema de optimización como es este caso, con variables discretas, la forma mas sencilla de las que hemos visto en clase es genéticos.

Hay 5 profesores en el departamento y 3 asignaturas de 3 créditos y otras 3 de 6 créditos. Cada profesor podría tener eventualmente cualquier carga docente, pero vamos a asumir que siempre serán los mismos profesores y las mismas asignaturas.

Para resolver el problema de forma general, lo que planteamos es una codificación del genético de la siguiente forma:

Cada gen representa a una de las asignaturas. Por lo tanto, hay 6 genes. Por decisión de diseño (es decir podría ser de otra forma) vamos a poner las 3 primeras asignaturas las de 6 créditos y las 3 últimas las de 3 créditos:

[A][B][C][D][E][F]

A – C = 6 créditos y de D - F = 3 créditos.

El valor de cada uno de los genes implica el profesor que se le ha asignado la asignatura.

Estos valores pueden estar repetidos, pero si se quedase sin créditos crearía un individuo invalido. Numeramos los profesores en el orden que están desde 1 a 5

Este individuo válido

[2][3][5][5][3][4]

El fitness sería el número de créditos sobrantes que hay en la asignación de la docencia. En este caso, sobrarían los créditos del profesor 1 = **3**, que no tendría docencia, el del profesor 2 estaría cubierto, el 3 estaría cubierto, el 4 le sobraría **1** y el 5 le sobrarían **3** total 7

Podemos ver que la asignación optima en este caso produciría siempre el mismo resultado es decir, para calcular la solución óptima, el resultado del fitness del genético sería siempre de 34 disponibles y 27 asignables, luego siempre deberían sobrar 7. En el caso se ve que son 7. Cual es el problema de esta codificación, que no hay evolución porque es puro prueba y error. Si lo hago mal el fitness será infinito y si lo hago bien el fitness será de 7.

Vamos a plantear el fitness de otra forma, para guiar mejor la solución del problema.

Imaginemos que tenemos una serie de profesores bien asignados pero que nos faltan otros profesores que están mal asignados (es decir, que dichos profesores están sobrepasados de horas). Nos interesa para el cruce elegir aquellos individuos que ya tengan una parte del gen correcto para que luego en el cruce o la mutación sea más fácil conseguir un individuo válido.

Así pues, vamos a puntuar con el fitness aquellos individuos que sean parcialmente correctos. En el caso del ejemplo anterior el fitness será máximo y valdrá 6 ya que los 6 están bien colocados y ninguno excede sus horas lectivas.

En este individuo

[1][1][1][1][1][1] el fitness es 0 porque no hay ninguno que esté bien colocado (se ha colocado un profesor con exceso de horas en todos los genes)

[2][1][1][1][1][1] este otro, tendrá un fitness de 1 porque sólo el primero está bien colocado

[2][1][1][2][1][1] este valdrá 2 porque hay 2 asignaturas con profesores asignados sin exceso de horas.

[2][3][1][2][4][1] sería de 4 porque hay 4 bien colocados.

Podríamos contar como inválidos solo los excesos es decir en este ultimo caso en vez de 4, 5 porque uno de los 1's que tiene exceso de horas en verdad si está correcto (el último)

Esta parte final era para nota.

Ejercicio 3: Sin describir como hacerlo, explica que otros algoritmos podrían resolver el problema y justifica por qué no los has usado. (0.5p)

Podría resolverse con A*. con algoritmo voraz, con aprendizaje por refuerzo, etc. Con A* plantear el problema en términos de encontrar la mejor solución óptima habría sido más complicado. ¿Cuál sería la heurística? es posible pero el problema se complicaría. En cualquier caso, se podría haber hecho así, porque en este caso la complejidad del problema es abordable por A*, así que el que haya elegido hacerlo por A* y lo haya planteado bien también valdría. El mismo problema tiene el algoritmo voraz. Aprendizaje por refuerzo podría usarse, pero aquí el problema principal es que refuerzo le darías al agente. Aquí no hay un agente que vaya resolviendo un problema poco a poco, se podría forzar, pero la representación es mucho más forzada que con genéticos.