

Descenso al fondo de un cráter en Marte

Código de Python

```
C: > Users > cmate > OneDrive > Documentos > Tec > 4to Semestre > agentesCrater CD.py > ...
1  import numpy as np
2  mapa_marte = np.load('crater_map.npy')
3  nr, nc = mapa_marte.shape
4  from simpleai.search import SearchProblem, astar, depth_first, breadth_first, greedy
5  from random import randrange
6  class CraterMarte(SearchProblem):
7
8      def actions(self, state):
9          x = state[0]
10         y = state[1]
11         step = state[2]
12         height = state[3]
13
14         lst = [(-1, -1), (-1, 0), (-1, 1), (0, -1), (0, 1), (1, -1), (1, 0), (1, 1)]
15         actions = []
16
17         for i in range(len(lst)):
18             a, b = lst[i]
19             if abs(mapa_marte[x+a,y+b] - height) <= step and mapa_marte[x+a][y+b] != -1:
20                 actions.append((x+a, y+b))
21
22         return actions
23
24     def result(self, state, action):
25         newx = action[0]
26         newy = action[1]
27         return newx, newy, state[2], mapa_marte[newx, newy], state[4]
28
29     def is_goal(self, state):
30         height = state[3]
31         targetheight = state[4]
32
33         return height <= targetheight
34
```

```
C:\Users\cmate> OneDrive > Documentos > Tec > 4to Semestre > agentesCrater CD.py > ...
34
35     def heuristic(self, state):
36         currx = state[0]
37         curry = state[1]
38         goalx = nr - int(5000/10.045)
39         goaly = int(3000/10.045)
40         height = state[3]
41
42         return 100*height + np.sqrt((currx-goalx)**2+(curry-goaly)**2)
43
44     def cost(self, state, action, state2):
45         return 10.045
46
47     x0 = nr - int(5800/10.045)
48     y0 = int(3350/10.045)
49
50     print("Se parte desde:",x0,y0)
51     result = astar(CraterMarte(initial_state=(x0, y0, 2, mapa_marte[x0][y0], 5)), graph_search=True)
52     print(len(result.path()), result.state)
53
54
55     for i in range(5):
56         randomx = randrange(mapa_marte.shape[0])
57         randomy = randrange(mapa_marte.shape[1])
58         print("Se parte desde:", randomx, randomy)
59         result = astar(CraterMarte(initial_state=(randomx, randomy, 2, mapa_marte[randomx][randomy], 5)), graph_search=True)
60         if result.state:
61             print(len(result.path()), result.state)
62         else:
63             print("No hay un camino posible a seguir")
```

Búsqueda codiciosa

- ¿Hasta dónde es capaz el explorador de llegar?

Puede llegar hasta el fondo del cráter, es muy parecido a astar, solo que puede no escoger el camino óptimo, sino el más rápido que llegue al destino.

- ¿Qué tan bueno es el algoritmo para llegar al fondo del cráter en los casos probados?

Es bueno, pero depende mucho de la heurística que se utiliza, ya que de ser una heurística mala, puede pasar que se atore en un mínimo local y que no llegue realmente al fondo del cráter.

Búsqueda con recocido simulado

- ¿Qué algoritmo logra llegar más profundo en el cráter?

En teoría, el algoritmo que debería llegar más rápido al fondo del cráter y con menos probabilidades de equivocarse es el de greedy, ya que este busca la solución más rápida, sin embargo puede que no sea la óptima y que en algunos casos se quede atorado en un mínimo local, por lo que ahí sería de mayor utilidad el algoritmo de recocido simulado.

- ¿Recomendarían a los ingenieros del robot utilizar alguno de estos algoritmos?

Depende, si lo que buscan es encontrar una solución lo más rápido posible, sí, pero si están buscando la solución óptima, seguramente existen algoritmos de búsqueda heurística más avanzados capaces de resolver de una forma más rápida y eficiente los problemas. En todo caso si se tuviera que emitir una recomendación, creo que diríamos que el de recocido simulado sería mejor, ya que no se estancaría en mínimos locales, aunque puede que tarde mucho antes de que el factor de la suerte sea el correcto.

Conclusión personal Daniel:

Para atacar este problema, comenzamos con realizar lo mismo que en el entregable pasado, que primero fue realizar la descarga de la imagen, otra diferente en la que se puede ver la forma de un cráter con una profundidad significativa, en esta, el objetivo era bajar al fondo del cráter de la forma más rápida, tanto usando greedy como simulated annealing. En cuanto al código, ocupamos parte del código de la entrega pasada, en la que realizamos una matriz que contenía las coordenadas y su respectiva altura, solo que a diferencia del pasado, esta vez el rover podría bajar cerca de dos metros, lo que facilitaba la tarea de bajar hasta el fondo. En esta entrega fue donde tuvimos más problemas, ya que para el de recocido simulado teníamos que crear una función value, la cual según nosotros trataba de hacer lo que hace la heurística pero al revés, siendo el valor de value más positivo cuando se acerca más al objetivo, pero por alguna razón no funcionaron los diferentes métodos que intentamos y tuvimos que saltarnos esa parte. Esta decisión la tomamos porque al revisar la librería de SimpleAI, no encontramos ningún ejemplo de cómo utilizarla, ni después de revisar lugares de comunidades tecnológicas como reddit o stackoverflow. En este problema igual reforcé los conceptos que vimos en clase sobre las propiedades del ambiente en el que estaríamos trabajando, que es marte, en este caso, el ambiente sería observable totalmente, ya que se cuenta con información detallada del lugar, si fuera una exploración de reconocimiento de la superficie donde no se tuviera un mapa, ahí si sería parcialmente observable, porque solo se vería lo de alrededor, por otra parte, el ambiente es de forma estocástica, ya que no se conoce cuál será la composición del suelo o si el clima cambiará repentinamente, existen variables aleatorias dentro del ambiente de un planeta como Marte. Igualmente sabemos que es secuencial porque se deben considerar los movimientos pasados para realizar los siguientes, y los movimientos siguientes dependen de los movimientos pasados para encontrar un mínimo local (cráter). Otra de las características del ambiente es que

es dinámico porque existen variables que no se pueden controlar (como la temperatura, viento, si caen nuevos meteoritos, etc.) básicamente no se puede detener el tiempo, por lo que el ambiente puede cambiar en lo que el robot realiza una acción. Por último, este es un problema continuo porque hay infinitas posibilidades de movimientos, desde moverse un centímetro hasta moverse un metro, no será preciso el movimiento, por lo que no pueden contarse todos los posibles movimientos. En esta entrega fue muy importante el trabajo en equipo porque se necesitaba escuchar de todas las ideas de ambos miembros del equipo e incluso del profesor por medio de las asesorías, ya que sin esto, hubiera sido mucho más difícil y hubiéramos avanzado mucho menos al intentar resolver el problema con el método de recocido simulado, que a pesar de que hicimos todo lo posible por lograrlo utilizar, no lo logramos. Con esto quiero concluir que me llevo una buena experiencia con esta primera interacción con la IA, y que me ha llamado mucho la atención poder utilizar la heurística y funciones para resolver problemas complejos que tendríamos que haber hecho desde cero para resolver sin librerías como la que utilizamos en clase.

Conclusión personal Carlos:

Durante la realización del problema, se presentaron múltiples dificultades que frenaron el desarrollo del proyecto, desde el planteamiento, ya que al ser un problema extenso se requirió leerlo en repetidas ocasiones, además de que se realizó investigación por aparte, en otras fuentes como Internet, leyendo múltiples artículos y videos relacionados con el tema. A la hora de leer el problema, en equipo decidimos que cada integrante intentaría obtener ideas para resolverlo, personalmente, me resultó complicado entender las simulaciones y las gamas de colores, ya que al ser de pequeña escala y distintas texturas resultaba complicado analizarlo, esto se resolvió investigando problemas similares y tratando de ponerlo en contexto con el problema planteado. Me resultó muy interesante la situación planteada, me parece muy buena idea que se trate de buscar agua o algún otro elemento en los cráteres de Marte, para saber si hubo, hay o podría haber vida en ese planeta, en especial la del robot y sus restricciones, que incluían que solo se podía desplazar ocho píxeles a su alrededor y que para evitar daños en el robot, no se permite que el explorador se mueva a píxeles con una diferencia de altura mayor a 0.5 metros de la posición en donde se encuentre, las restricciones representaron una dificultad para mí, y al cuestionarme sobre cómo se podía resolver el problema, decidí realizar un análisis de las propiedades del ambiente, de las cuáles concluí que las del robot son no observable, ya que debido a su restricción sólo se puede desplazar ocho píxeles alrededor de dónde se encuentre, es estocástico, ya que al ser parcialmente observable, el cambio constante del ambiente, el cambio constante de las acciones del agente, además de otros factores, indican que será cuestión de suerte que el agente pueda encontrar datos precisos y se pueda desplazar por el cráter, además de depender de no caerse y que el robot pueda sufrir daños, dinámico, el ambiente va cambiando mientras el agente piensa que hacer o qué acción realizar, no se sabe con que se puede encontrar en la superficie, al estar en el espacio, el ambiente va cambiando continuamente, puede ocurrir algo en el cráter que dañe al agente, continuó, las acciones que se realizan no pueden ser enumeradas, se va analizando sobre un cráter, tiene que caminar, buscar, analizar, al realizar este análisis se facilitó resolver el problema en papel para entrar en contexto con el agente, pensar como él y generar ideas de cómo se podría comportar. Al tener una idea de esto, se procedió a realizar el código, a la hora de plantearlo, se presentaron un par de adversidades, las librerías utilizadas no corrían de una manera óptima, lo cuál frenó el desarrollo del proyecto, además de que si en el código, se llega en algún punto al mínimo local, ya no sale de ahí, es como si el robot

Daniel I. Núñez López

A01654137

Carlos Mateos Pérez

A01654085

se atorara, queda pendiente de que en futuros trabajos se realice una investigación más extensa, buscando y aprendiendo nuevas técnicas para ayudar a resolver estos problemas y se pueda obtener un resultado más óptimo y específico.