Proyecto Integrador: Tercer Avance Búsquedas Locales

Equipo 5

Sebastían Neri	A01750190
Aldo Sandoval	A01751137
Emiliano Padilla	A01658972

Descripción Medio Ambiente

El rover tiene la capacidad de recorrer el planeta calculando la presión atmosférica del punto en el que se encuentra para estimar la distancia a la atmósfera. Cuando encuentra una montaña con altura más alta o baja que la que consideraba como más alta o baja manda una señal con las coordenadas del lugar.

Su entorno es accesible porque todos sus sensores pueden detectar la información disponible del entorno. Es determinista porque es un agente único el que se encuentra en el planeta, pues el planeta no tiene actividad sísmica ni fenómenos metereológicos. Es no episódico (secuencial) porque las decisiones tomadas por el rover afectan las muestras que tome. Estático porque no hay otro agente con el rover y porque no tiene fenómenos naturales. Es discreto porque el entorno donde está el rover tiene un número finito de estados.

Actúa a través del sistema de movimiento (llantas, motores), dispositivo de comunicación. Tiene los siguientes sensores: Cámaras, micrófonos, sensores de viento, escáner de entornos, radar y dispositivo que calcula la presión atmosférica para obtener una aproximación de la distancia a la atmósfera.

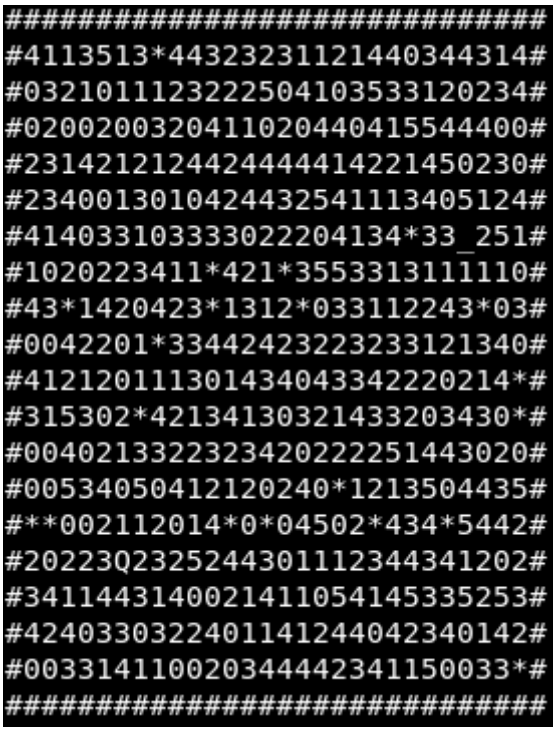
Descripción Formal del Problema

1. Configuraciones

El mapa es un string con 620 caracteres, entre estos se encuentran 20 obstáculos, 10 lagos, una “*Q*”, puntos representados con números escogidos aleatoriamente del 0 al 5, saltos de línea, y bordes del mapa.

Una configuración es un número entero que representa la posición en la que se visita el punto (elemento N del string).

Imagen 1. Ejemplo de configuración



2. Reordenamientos

1. Se avanza a un punto si la diferencia del nivel del estado actual y del siguiente es 1.
2. Se avanza a una posición si esta no es un obstáculo o un borde.
3. No se puede mover a la derecha o izquierda.

3. Función objetivo

$$E = mapa(p)$$

> p es la posición dentro del mapa asociada a un nivel

4. Calendarización de recocido

- En cada iteración se calcula el cambio de energía y se utiliza la función de probabilidad de Boltzmann.

$T_0 = 100$
$\lambda = 1.4$
$T_{i+1} = \lambda T_i$

Notas a considerar

En el programa, el valor a optimizar es el nivel en el mapa que puede encontrar el rover. Puede ser el nivel más bajo representado por cero o el más alto por el cinco. En las corridas posteriores se identificó al rover con el color verde y el nivel que consideró óptimo de color amarillo (dependiendo de que búsqueda realizó). Se definió el algoritmo de random restart con 500 iteraciones. En la calendarización de puntos extra se usó la función de probabilidad de Boltzmann.

Hill Climbing

Imagen 2. Nivel más bajo

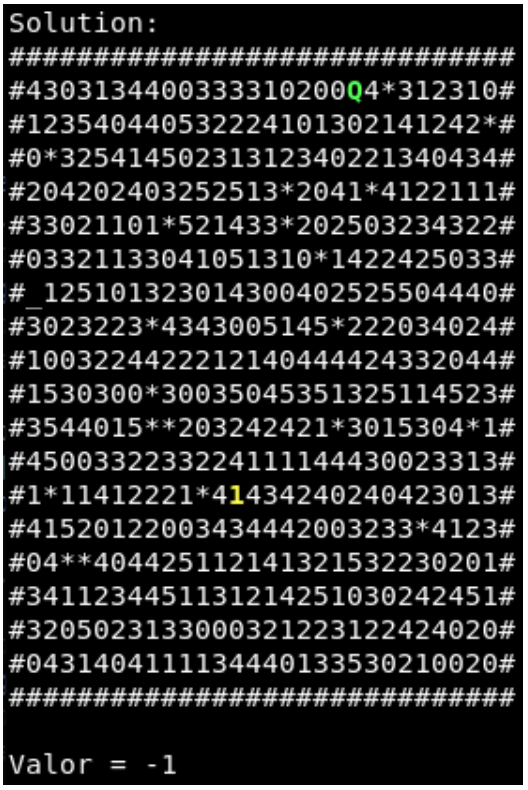
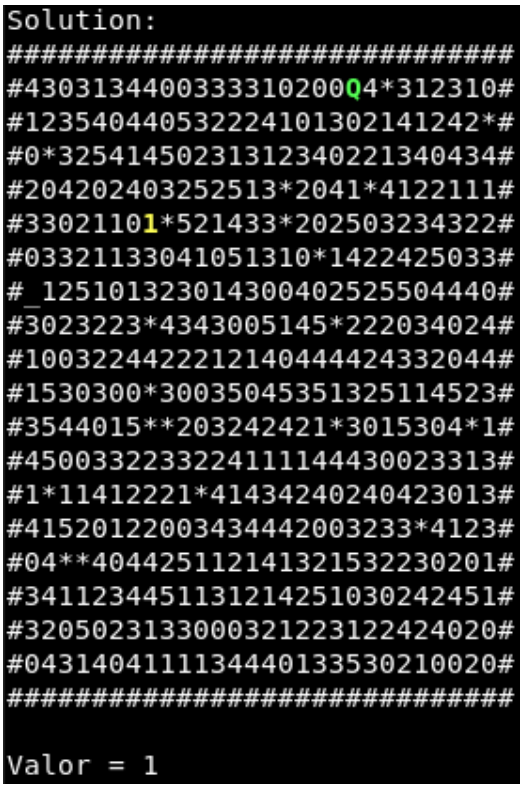


Imagen 3. Nivel más alto



Random-restart Hill Climbing

Imagen 4. Nivel más bajo

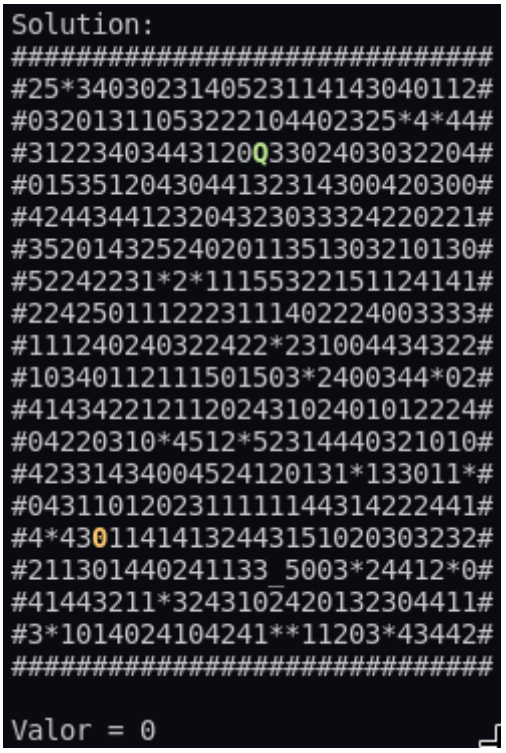
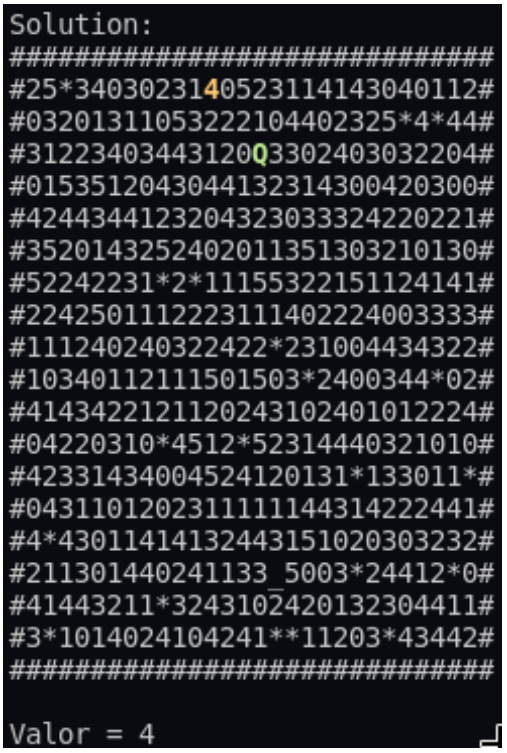


Imagen 5. Nivel más alto



Simulated Annealing (exp_schedule)

Imagen 6. Nivel más bajo

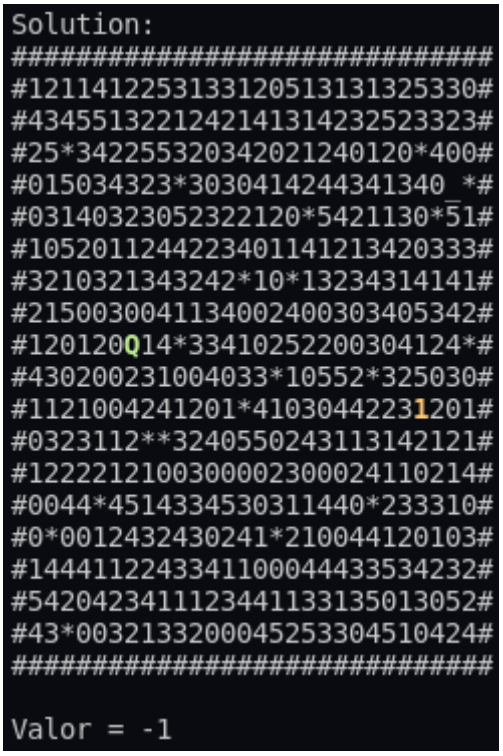
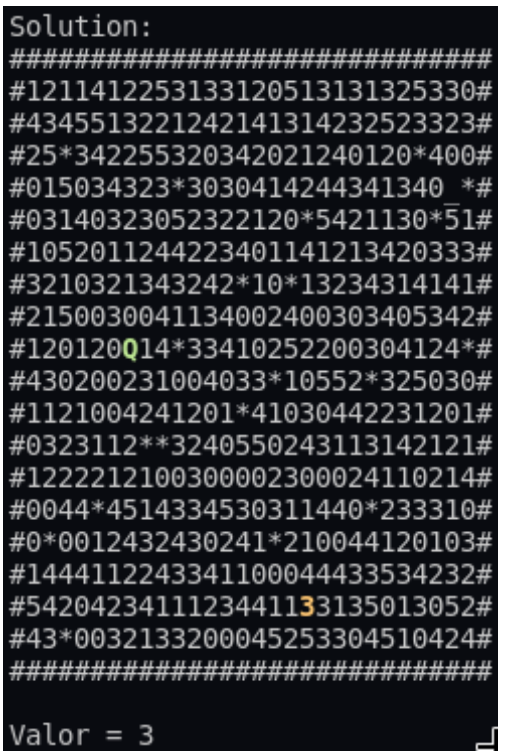


Imagen 7. Nivel más alto



Opcional - Simulated Annealing (calendarización de temperatura)

Imagen 8. Nivel más bajo

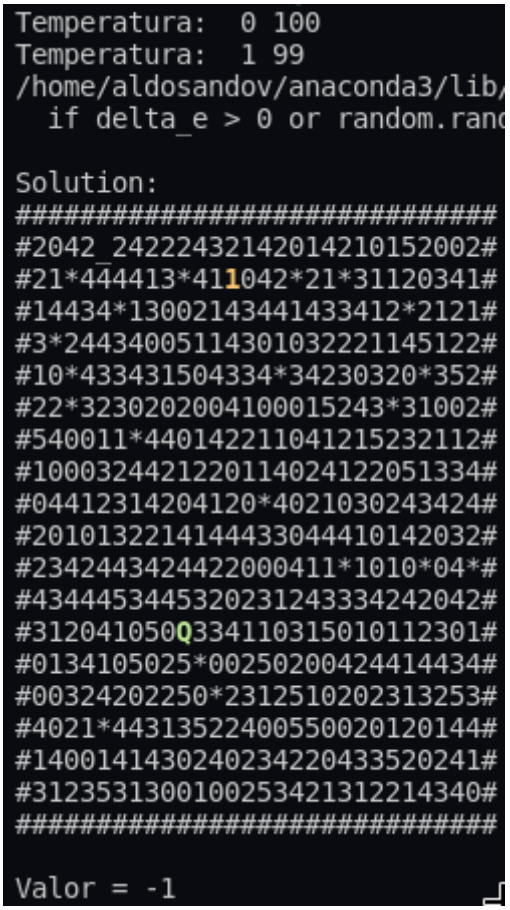
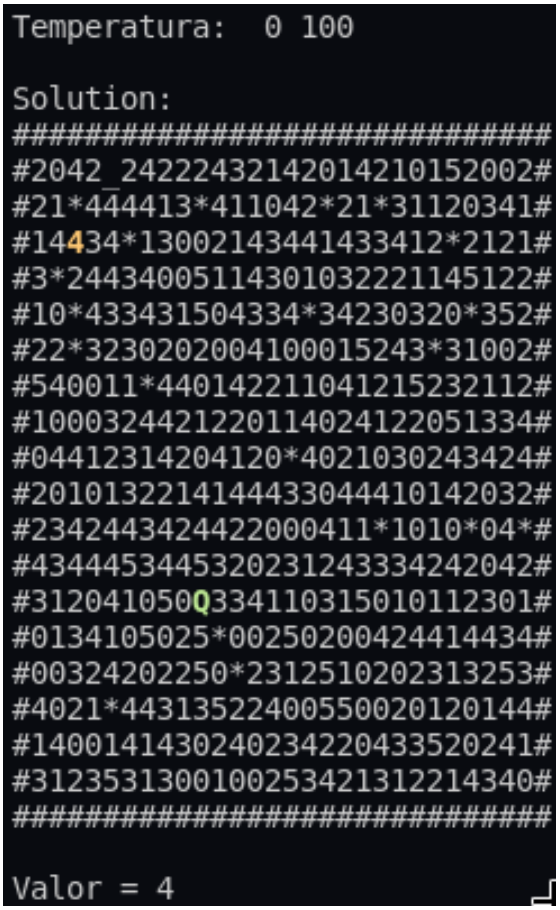


Imagen 9. Nivel más alto



Análisis de las búsquedas

En las corridas se pueden observar resultados diferentes dependiendo del algoritmo utilizado. Como se mencionó en clase el algoritmo de hill-climbing tiene un error en su implementación por lo que los resultados de esta búsqueda no aportan demasiado valor al contraste. En cambio, el de random restart nos ofrece un resultado muy cercano al valor deseado. Se realizan 500 iteraciones y se elige el valor óptimo para esa búsqueda. En el de simulated annealing depende de la calendarización utilizada. Se puede concluir que los mejores resultados se obtienen utilizando los algoritmos de random restart y recocido simulado usando nuestra función de temperatura. Se podría mejorar la búsqueda cambiando la función objetivo, así como los algoritmos utilizados. Por último, nuestro proyecto es funcional, sirve para ejemplificar los términos fundamentales de la inteligencia artificial, pero ya a una mayor escala la librería de simpleai deja mucho que desear.

Código en Github: https://github.com/sebastianneri/searchAlgorithms.git