

Navegación de un Robot en un Terreno Dinámico

1. Planteamiento del Problema

El objetivo de esta práctica es implementar el algoritmo A* modificado con un parámetro ϵ (A-Epsilon) para guiar un robot en la búsqueda de una ruta en un mapa, evitando obstáculos. El robot debe moverse desde una posición de inicio hasta un objetivo, en un entorno de 5x5 donde algunas celdas están bloqueadas (representadas por 1) y otras son transitables (representadas por 0).

El algoritmo A-Epsilon es una versión modificada del algoritmo A*, en la que se introduce un parámetro ϵ que ajusta cómo se pondera la heurística en comparación con el costo acumulado. Este valor permite equilibrar la exploración (búsqueda de soluciones más rápidas) y la explotación (búsqueda de soluciones más óptimas), lo que mejora la eficiencia de la búsqueda.

Descripción del Mapa:

- **Dimensiones del mapa:** 5x5 celdas.
- **Celdas bloqueadas:** Representadas por 1.
- **Celdas transitables:** Representadas por 0.
- **Posición de inicio:** Esquina superior izquierda (0,0).
- **Posición de objetivo:** Esquina inferior derecha (4,4).

2. Implementación del Algoritmo A-Epsilon

2.1 Representación del Estado del Problema

El estado del problema se representa mediante un mapa bidimensional (matriz 5x5). Cada celda puede ser libre (0) o bloqueada (1). El robot comienza en la celda (0,0) y debe llegar al objetivo en la celda (4,4), evitando los obstáculos.

2.2. Definición del Algoritmo A-Epsilon

El algoritmo A-Epsilon se basa en la evaluación de nodos mediante la función $f(n)$, que es la suma de dos componentes:

- **g(n):** El costo acumulado desde el nodo inicial hasta el nodo nnn.
- **h(n):** La heurística, que en este caso es la distancia Manhattan entre el estado actual del robot y el objetivo.

La principal diferencia en A-Epsilon es que la heurística $h(n)$ es ponderada por un parámetro ϵ , modificando la función de evaluación a:

$$f(n)=g(n)+\epsilon \cdot h(n)$$

2.3. Heurística de la Distancia Manhattan

La heurística utilizada es la **distancia Manhattan**, que calcula la distancia entre dos puntos en un plano de acuerdo con las diferencias absolutas de las coordenadas x e y. Para cada celda, se calcula la distancia entre la celda actual y el objetivo (4,4).

2.4. Funciones Principales del Algoritmo

1. **heuristic(a, b):** Calcula la distancia Manhattan entre dos puntos a y b.
2. **get_neighbors(node, rows, cols):** Devuelve los vecinos de una celda, ignorando los obstáculos.
3. **a_epsilon(start, goal, epsilon):** Implementa el algoritmo A-Epsilon para encontrar la ruta desde el inicio hasta el objetivo.
4. **reconstruct_path(came_from, current):** Reconstruye la ruta desde el nodo final hasta el inicio.
5. **plot_path(terrain, path, start, goal):** Visualiza la ruta encontrada en el mapa, mostrando el recorrido del robot.

2.5. Visualización del Proceso

El código incluye una función `plot_path` que usa la biblioteca `matplotlib` para dibujar el mapa en cada paso del algoritmo. Esto permite observar cómo el robot se mueve hacia el objetivo mientras evita los obstáculos. En el gráfico, las celdas bloqueadas son de color negro, las celdas libres son blancas, el inicio es marcado con un círculo verde, el objetivo con un círculo rojo, y el camino encontrado por el algoritmo es mostrado con puntos azules.

3. Resultados Obtenidos

Durante la ejecución del algoritmo A-Epsilon, se encontró una ruta que lleva al robot desde el punto de inicio (0,0) hasta el objetivo (4,4), evitando las celdas bloqueadas en el mapa. El algoritmo utilizó la distancia Manhattan como heurística, ajustada con un valor $\epsilon=2.0$ para balancear la exploración y explotación.

Estado inicial:

0	0	0	1	0
0	1	0	1	0
0	1	0	0	0
0	0	0	1	0
1	0	0	0	0

Estado objetivo:

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

Durante la ejecución, el robot avanzó paso a paso hacia el objetivo, evitando obstáculos en su camino. La visualización gráfica mostró cómo el robot siguió el camino correcto.

4. Análisis de Resultados

El algoritmo A-Epsilon funcionó correctamente para encontrar una ruta eficiente en el mapa dado. La elección del valor de ϵ afectó directamente el rendimiento del algoritmo:

- **Valores bajos de ϵ** dieron mayor peso al costo acumulado, haciendo que el robot explorara más profundamente, pero sin encontrar necesariamente la solución más eficiente.

- **Valores altos de ϵ** orientaron al algoritmo a concentrarse más en la heurística (distancia Manhattan), lo que redujo el tiempo de búsqueda.

El rendimiento de A-Epsilon fue bueno, encontrando rutas válidas en un tiempo razonable.

5. Recomendaciones para Futuras Implementaciones

- **Ajuste de ϵ :** Se recomienda experimentar con diferentes valores de ϵ para analizar su impacto en la eficiencia y el tiempo de ejecución.
- **Mejoras en la heurística:** Se pueden probar otras heurísticas, como la distancia Euclidiana o una heurística basada en la densidad de obstáculos, para observar si mejoran la velocidad del algoritmo.
- **Pruebas con mapas más grandes:** Se puede probar con mapas de mayores dimensiones para evaluar la escalabilidad del algoritmo.