Rover Autónomo Marciano para Exploración en Entornos Desconocidos

Simulación y Planificación de Misiones Autónomas

Jennifer de la Caridad Sánchez Santana

3 de noviembre de 2025

Repositorio del Proyecto

1. Descripción del Problema

La exploración autónoma de entornos desconocidos constituye uno de los desafíos más significativos en la robótica y la inteligencia artificial. Este proyecto aborda dicho desafío mediante el desarrollo de una simulación de un rover autónomo, diseñado para ejecutar misiones científicas en un entorno discreto modelado como un grafo ponderado. En este modelo, cada nodo representa una celda del terreno con características específicas, y las aristas ponderadas representan los costos (energéticos y temporales) de desplazarse entre ellas.

El objetivo principal es desarrollar un sistema de inteligencia artificial capaz de planificar y ejecutar rutas óptimas hacia puntos de interés científico (POIs), operando bajo restricciones críticas de energía y comunicación. Para ello, se integra un conjunto de técnicas de IA que incluyen algoritmos de búsqueda, planificación con metaheurísticas y satisfacción de restricciones (CSP).

1.1. Características del Entorno

El entorno de simulación está diseñado para reflejar la complejidad de un terreno marciano:

- **Tipos de Terreno:** Plano, rocoso, arena fina y dunas. Cada tipo de terreno implica un costo energético y temporal de tránsito diferente.
- **Elementos Dinámicos y Estáticos:**
  - **Obstáculos:** Celdas no transitables que bloquean el paso del rover.
  - **Puntos de Interés Científico (POIs):** Ubicaciones clave que el rover debe visitar para recolectar muestras o datos.
  - **Punto de Recolección:** Zona designada donde el rover debe depositar las muestras recolectadas para su posterior análisis.
  - **Base:** El punto de partida y final de la misión. Aquí el rover puede recargar su batería hasta su capacidad máxima y transmitir los datos recopilados.

1.2. Objetivos de la Misión del Rover

El rover debe cumplir con los siguientes objetivos de manera autónoma:

1. **Exploración Científica:** Visitar todos los POIs definidos en el mapa.
2. **Gestión de Muestras:** Recolectar muestras en cada POI y depositarlas en el punto de recolección.
3. **Optimización de Recursos:** Minimizar el consumo de energía durante toda la misión.
4. **Comunicación y Retorno:** Regresar a la base antes de que se agote por completo su energía para transmitir la información recolectada y finalizar la misión con éxito.

2. Módulos de Inteligencia Artificial Implementados

2.1. Búsqueda: Navegación Punto a Punto

Para la navegación local entre objetivos (ej. de un POI a la base), se implementaron y compararon dos algoritmos de búsqueda de caminos óptimos en grafos: **Dijkstra** y **A***. El costo de cada movimiento no es uniforme, sino que depende del tipo de terreno de la celda de destino.

2.1.1. Fundamento Teórico:

- **Algoritmo de Dijkstra:** Es un algoritmo de búsqueda exhaustiva y no informada. Garantiza encontrar el camino de menor costo desde un nodo origen a todos los demás nodos en un grafo con pesos no negativos. Su funcionamiento se basa en expandir sistemáticamente el nodo con el costo acumulado más bajo hasta alcanzar el destino. Si bien es óptimo, puede ser computacionalmente costoso, ya que explora en todas las direcciones por igual.

- **Algoritmo A* (A-asterisco):** Es un algoritmo de búsqueda informada que mejora el rendimiento de Dijkstra utilizando una heurística para guiar la exploración hacia el objetivo. A* evalúa los nodos mediante la función $f(n) = g(n) + h(n)$, donde:

  - $g(n)$ es el costo real del camino desde el nodo inicial hasta el nodo n.
  - $h(n)$ es el costo estimado (heurística) desde el nodo n hasta el nodo objetivo.

  Si la heurística es **admisible** (nunca sobreestima el costo real) y **consistente**, A* es óptimo y, por lo general, mucho más rápido que Dijkstra, ya que evita explorar caminos que poco prometen.

- **Heurísticas Implementadas:**

  1. **Distancia Manhattan:** Calculada como $|x_1 - x_2| + |y_1 - y_2|$. Es ideal para movimientos restringidos a una cuadrícula y es computacionalmente muy eficiente.
  2. **Distancia Euclidiana:** Calculada como la raíz cuadrada de la suma de los cuadrados de las diferencias de coordenadas. Es más precisa si el movimiento pudiera ocurrir en cualquier dirección.

2.1.2. Resultados Comparativos

Para evaluar el rendimiento, se realizaron 448 simulaciones de navegación entre puntos aleatorios.

Cuadro 1: Comparativa de rendimiento de algoritmos de búsqueda.

Algoritmo	Heurística	Promedio Pasos	Promedio Costo	Promedio Tiempo (ms)
a_star	manhattan	15.67	26.91	0.228
a_star	euclidiana	15.69	26.91	0.258
dijkstra	—	15.67	26.91	0.322

2.1.3. Análisis y Gráficos

Los resultados muestran que los tres algoritmos encuentran caminos con una longitud y costo prácticamente idénticos, lo que confirma la optimalidad de A* con ambas heurísticas. La diferencia clave reside en el **tiempo de ejecución**. A* es significativamente más rápido que Dijkstra porque la heurística reduce drásticamente el espacio de búsqueda. Entre las heurísticas, **Manhattan** se muestra ligeramente superior en tiempo de ejecución, probablemente debido a su menor costo computacional.

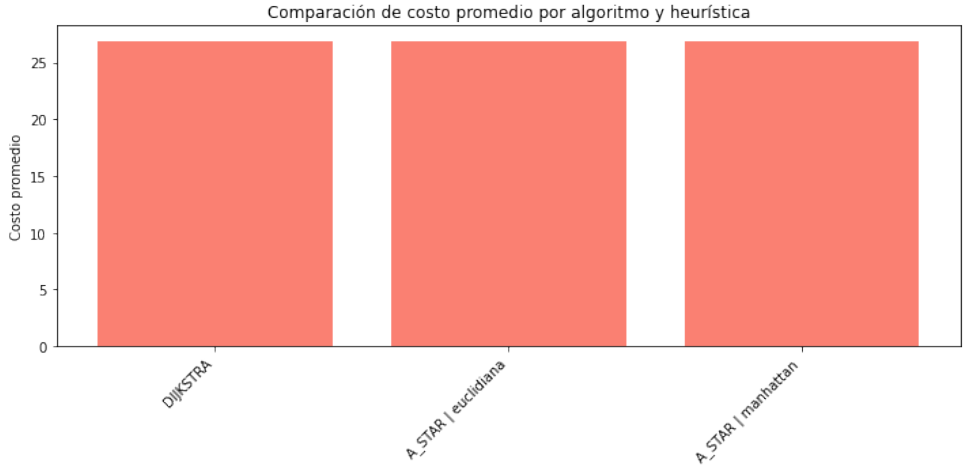

Figura 1: Comparación de Coste Promedio por Algoritmo. Muestra que el promedio de coste para Dijkstra, A* (Manhattan) y A* (Euclidiana) es casi idéntico.

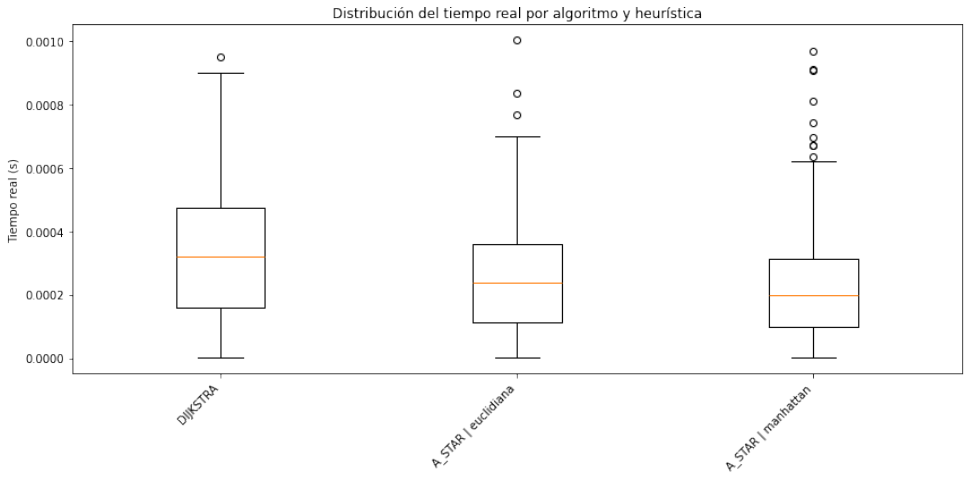

Figura 2: Comparación de Tiempo de Ejecución Promedio (ms) por Algoritmo. Destaca la diferencia de rendimiento. Dijkstra muestra el mayor tiempo de ejecución, mientras que A* con heurística Manhattan es el más rápido, demostrando su eficiencia.

2.2. Planificación con Metaheurísticas: Estrategia de Misión

Para abordar el problema de la planificación a alto nivel —decidir el orden en que visitar los POIs, cuándo recargar y cuándo depositar muestras—, se implementaron dos

metaheurísticas clásicas: un **Algoritmo Genético (AG)** y un **Recocido Simulado (Simulated Annealing, SA)**.

Las acciones del rover se modelaron con un enfoque tipo STRIPS:

Acción	Precondiciones	Efectos
`ir_a_poi`	Suficiente Energía, no estar en el POI	Cambia posición, consume energí
`ir_a_base`	Suficiente Energía , no estar en la base	Cambia posición, consume energí
`recargar`	Estar en la base, energía menor que la máxima	Energía se establece al máximo
`dejar_muestras`	Tener muestras, estar en punto de recolección	Muestras transferidas, inventario

- **Algoritmo Genético (AG):** Optimiza el orden de visita de los POIs. El AG evoluciona una población de "planes"(secuencias de acciones) a través de selección, cruce y mutación. Una función clave, `reparar_plan`, asegura la viabilidad de cada individuo.
- **Recocido Simulado (SA):** Comienza con un plan aleatorio y lo mejora iterativamente. Genera planes "vecinos"mediante pequeñas modificaciones. Acepta planes peores al principio (con alta "temperatura") para escapar de óptimos locales, y se vuelve más selectivo a medida que la temperatura disminuye.

2.3. CSP – Satisfacción de Restricciones

Este módulo actúa como un validador de la viabilidad de los planes generados por las metaheurísticas. Un plan es una solución a un Problema de Satisfacción de Restricciones (CSP) definido por:

- **Variables:** Secuencia de acciones del plan, estado de energía y posición del rover en cada paso.
- **Dominios:** Conjunto de acciones posibles (`ir_a_poi`, `recargar`, etc.) y ubicaciones válidas.
- **Restricciones:**
  1. La energía del rover debe ser siempre mayor que cero.
  2. No se puede visitar el mismo POI más de una vez.
  3. La energía debe ser suficiente para completar cada tramo del viaje planificado.
  4. La misión debe finalizar en la base con las muestras depositadas.

Se implementó un solucionador basado en propagación de restricciones. Si un plan propuesto por el AG o el SA viola alguna restricción, es descartado o marcado como inviable.

3. Resultados Experimentales

Se realizaron 30 simulaciones completas de misiones para evaluar la efectividad de las estrategias de planificación. El costo total de la misión y el tiempo de cómputo para generar el plan fueron las métricas principales. Se utilizó el costo de la ruta encontrada

por una búsqueda exhaustiva (BFS) como referencia de optimalidad(en un espacio de 5 dimensiones).

Cuadro 2: Comparación de estrategias de planificación de misión.

Algoritmo	Count	Mean (Costo)	Std (Costo)	Mean (Tiempo ms)
BFS	28.0	50.57	11.01	2.55
SA	16.0	51.87	17.09	3.83
GA	16.0	59.25	15.53	7.93

3.1. Análisis

El **Recocido Simulado (SA)** demostró ser la estrategia más equilibrada y eficaz. Obtuvo un costo promedio de misión (51.87) muy cercano al óptimo de referencia proporcionado por BFS (50.57), pero con un tiempo de planificación (3.83 ms) significativamente menor.

El **Algoritmo Genético (GA)**, aunque funcional, mostró un costo promedio mayor (59.25) y fue el más lento en generar un plan (7.93 ms). Esto sugiere que, para la configuración de este problema, el mecanismo de búsqueda del SA fue más efectivo.

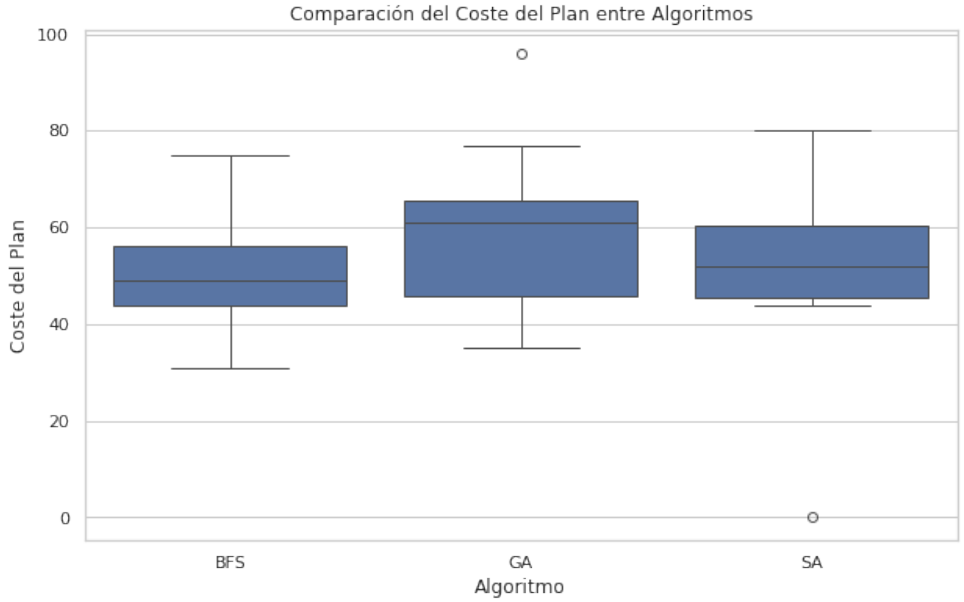

Figura 3: Comparación de Coste Promedio por Algoritmo.

4. Discusión

El desarrollo del proyecto presentó varios desafíos técnicos y de diseño:

1. **Balance entre Complejidad y Rendimiento:** Aumentar la resolución del mapa o el número de POIs eleva exponencialmente el espacio de búsqueda. El principal

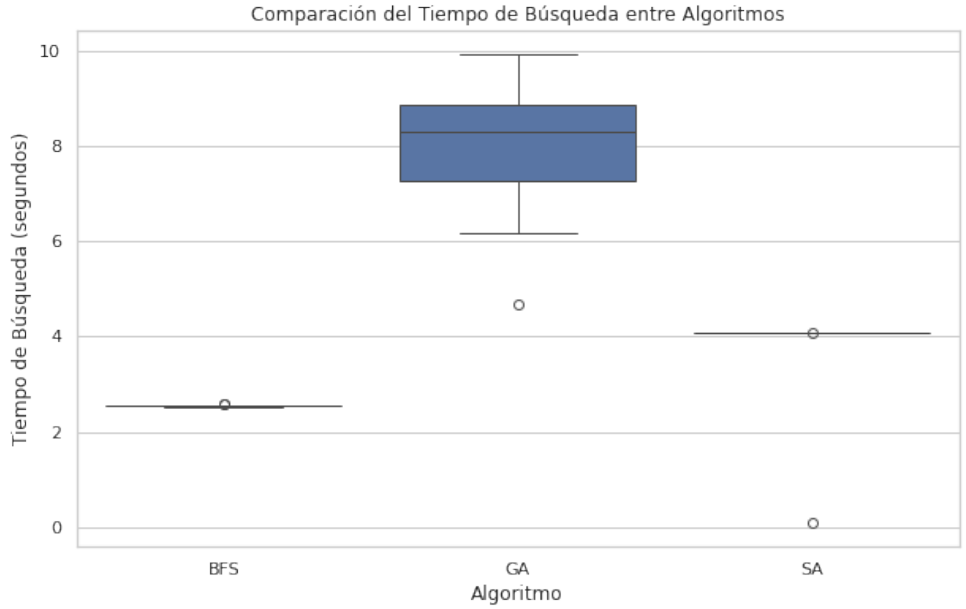

Figura 4: Comparación de Tiempo de Ejecución por Algoritmo. Destaca la diferencia de rendimiento. GA muestra el mayor tiempo de ejecución, mientras que AS es el más rápido, demostrando su eficiencia.

reto fue ajustar los parámetros de los algoritmos para mantener los tiempos de cómputo en niveles prácticos sin sacrificar la calidad de la solución.

2. **Integración de Módulos:** La arquitectura modular permitió una clara separación de responsabilidades.

La solución final integra exitosamente los módulos: el **CSP** garantiza que el terreno generado sea solucionable; las **metaheurísticas** (principalmente SA) definen una estrategia de misión de alto nivel; y los algoritmos de **búsqueda** (A*) ejecutan la navegación óptima entre los hitos de dicha estrategia.

5. Conclusiones y Mejoras Futuras

5.1. Conclusiones

El proyecto demuestra con éxito la viabilidad de una arquitectura de IA híbrida para la gestión de misiones autónomas complejas. El rover simulado es capaz de navegar, planificar y optimizar su comportamiento en entornos dinámicos con recursos limitados. Los resultados experimentales validan que:

- **A*** con la heurística de **Distancia Manhattan** es la opción más eficiente para la navegación punto a punto en este tipo de entornos de cuadrícula.
- El **Recocido Simulado (SA)** es una metaheurística superior al Algoritmo Genético para este problema específico de planificación, ofreciendo un excelente equilibrio entre calidad de la solución y tiempo de cómputo.

- La integración de técnicas de búsqueda, planificación y CSP permite descomponer un problema complejo en subproblemas manejables, mejorando la robustez y mantenibilidad del sistema.

5.2. Mejoras Futuras

1. **Aprendizaje por Refuerzo:** Implementar un agente que aprenda políticas de navegación óptimas a través de la experiencia. El rover podría aprender a evitar ciertos terrenos o a desarrollar estrategias de recarga más eficientes a lo largo de múltiples misiones.

2. **Re-planificación Dinámica:** Desarrollar un sistema que pueda adaptar el plan en tiempo real ante eventos imprevistos, como la aparición súbita de un obstáculo o el fallo de un componente.

3. **Comunicación Multi-agente:** Simular escenarios con múltiples rovers colaborando en una misión, lo que introduciría nuevos desafíos de coordinación, asignación de tareas y comunicación entre agentes.

6. Referencias

Referencias

[1] Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

[2] Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2), 100-107.

[3] Nilsson, N. J. (1998). *Artificial Intelligence: A New Synthesis*. Morgan Kaufmann Publishers.