



## I. Visión general del proyecto

Los sistemas de planeación de rutas para múltiples agentes suelen depender de un planificador centralizado que escala mal cuando aumenta el número de vehículos y la complejidad del mapa. Esto provoca solapamientos de trayectorias, conflictos entre agentes y altos tiempos de cómputo, especialmente en entornos urbanos dinámicos. En esta simulación, una flota de camiones recolectores de residuos opera de forma autónoma sobre un mapa discretizado. Cada camión decide:

- qué contenedor atender,
- cómo planear la ruta hacia él,
- cuándo regresar al depósito a descargar o recargar energía.

El proyecto integra:

- Partición geométrica del espacio** mediante diagramas de Voronoi;
- Planeación de rutas** con A\* y D\* Lite;
- Aprendizaje por refuerzo (RL)** para coordinación adaptativa.

## II. Objetivos

### Objetivo general

- Diseñar e implementar un planificador multi-agente que use partición espacial, A\*/D\* Lite y aprendizaje por refuerzo para generar rutas eficientes y con baja tasa de conflictos en un entorno de vehículos autónomos recolectores.

### Objetivos específicos

- Reducir el tiempo de planificación en comparación con un planificador sin partición ni RL.
- Disminuir el solapamiento de rutas y los conflictos entre camiones.
- Incorporar re-planeación dinámica ante cambios en el entorno usando D\* Lite.
- Evaluar la escalabilidad al incrementar número de agentes y tamaño del mapa.

## III. Planteamiento del problema

Los métodos tradicionales de planeación de rutas escalan mal cuando aumenta el número de agentes y la complejidad del entorno, generando conflictos y altos tiempos de cálculo. Se necesita una solución que divida el espacio, planifique rutas de forma local y aprenda a coordinar agentes sin depender de una búsqueda centralizada.

**Formalmente**, dado:

- un mapa modelado como grafo dirigido y ponderado,
- un conjunto de camiones con posiciones iniciales, depósitos y contenedores,
- y restricciones de energía y capacidad,

se busca una política que asigne contenedores y rutas a cada agente de modo que:

- se minimice el coste total (distancia/tiempo recorrido),
- se reduzca el número de conflictos y solapamientos,
- y se mantenga la operación ante cambios en el entorno.

## IV. Diseño del algoritmo

El sistema sigue una arquitectura por etapas que combina grafos, geometría computacional y búsqueda informada.

### 1) Modelado del entorno

- El mapa se representa como un **grafo dirigido y ponderado** donde los nodos son celdas accesibles y las aristas movimientos válidos con un coste asociado.
- Depósitos, camiones y contenedores se ubican sobre este grafo.

### 2) Partición geométrica (Voronoi)

- Se construyen regiones de Voronoi a partir de las posiciones base de los camiones o depósitos.
- Cada agente atiende prioritariamente los contenedores dentro de su región para distribuir la carga de trabajo y reducir competencia.

### 3) Planeación de rutas con A\* y D\* Lite

- A\***: rutas iniciales hacia contenedores o depósito.
- D\* Lite**: re-planeación incremental cuando cambian los costes o aparecen obstáculos.

### 4) Máquina de estados del camión

- idle*  $\rightarrow$  *to\_bin*  $\rightarrow$  *collecting*  $\rightarrow$  *to\_depot*  $\rightarrow$  *idle*.
- Las transiciones dependen de contenedor vacío, camión lleno o baja energía.

### 5) Protocolo de reserva de contenedores

- Cada contenedor tiene *reserved\_by* y *reserved\_until*.
- Durante el *cooldown* otros camiones evitan ese objetivo, reduciendo viajes duplicados.

### 6) Coordinación con aprendizaje por refuerzo

- Decide qué contenedor atender, cuándo volver al depósito y cómo balancear energía vs. distancia.
- La recompensa combina contenedores atendidos, distancia/tiempo y penalizaciones por estados ineficientes.

## V. Análisis de complejidad

### A\*

- Tiempo:  $O(E \log V)$  usando cola de prioridad, donde  $V$  es el número de nodos y  $E$  el número de aristas.
- Espacio:  $O(V)$  para costes, heurística y estructura de frontera.

### D\* Lite

- Basado en A\*, pero con actualizaciones incrementales.
- En mapas con cambios locales, el coste es menor que recalcular la ruta completa, ya que solo se actualizan nodos afectados.

### Voronoi

- Construcción típica en  $O(n \log n)$  para  $n$  sitios generadores.
- La partición reduce el problema global a subproblemas locales, haciendo que cada búsqueda A\* recorra un subgrafo más pequeño.

### Aprendizaje por refuerzo

- El coste depende del número de episodios y del tamaño del espacio de estados y acciones.
- Al apoyarse en rutas ya razonables dadas por A\*/D\* Lite, el RL explora sobre decisiones de alto nivel en lugar de aprender navegación desde cero.

## VI. Resultados experimentales

Las gráficas resumen el comportamiento de la flota en un episodio representativo y en el conjunto de simulaciones:

- Eficiencia por camión (botes/distancia)**: las eficiencias se concentran entre 0.08 y 0.17 botes por unidad de distancia. El camión 0 y el camión 2 muestran la mejor relación botes/distancia, mientras que el camión 3 contribuye menos pero sigue siendo útil para cubrir zonas específicas.
- Desempeño global**: en el episodio ilustrado se recolectan 36 botes con una distancia total recorrida de 300 unidades, lo que se alinea con una eficiencia promedio global de  $\approx 0.120$  botes por unidad de distancia.
- Mapa de posiciones**: el mapa de calor muestra una cobertura distribuida del grid; las regiones más visitadas coinciden con las áreas asignadas por la partición de Voronoi, indicando que cada camión opera principalmente dentro de su zona.
- Progreso del episodio**: la curva de botes recolectados crece rápidamente al inicio y se estabiliza alrededor de 36 botes, mientras que la distancia aumenta hasta alcanzar las 300 unidades, reflejando un uso eficiente de los pasos de simulación.
- Desempeño individual**: el camión 1 recolecta 19 botes y recorre 179 unidades, el camión 0 recolecta 8 botes en 46 unidades, el camión 2 recolecta 6 botes y termina con carga residual en el depósito, y el camión 3 recolecta 3 botes con 37 unidades de recorrido. Esto sugiere una ligera especialización entre camiones, pero sin desbalance extremo.
- Estadística agregada**: a lo largo de 1001 episodios registrados, la simulación mantiene una eficiencia media de 0.120 botes/distancia con una distancia total de 300 unidades en el episodio mostrado, lo que evidencia un comportamiento estable del planificador bajo la configuración actual.

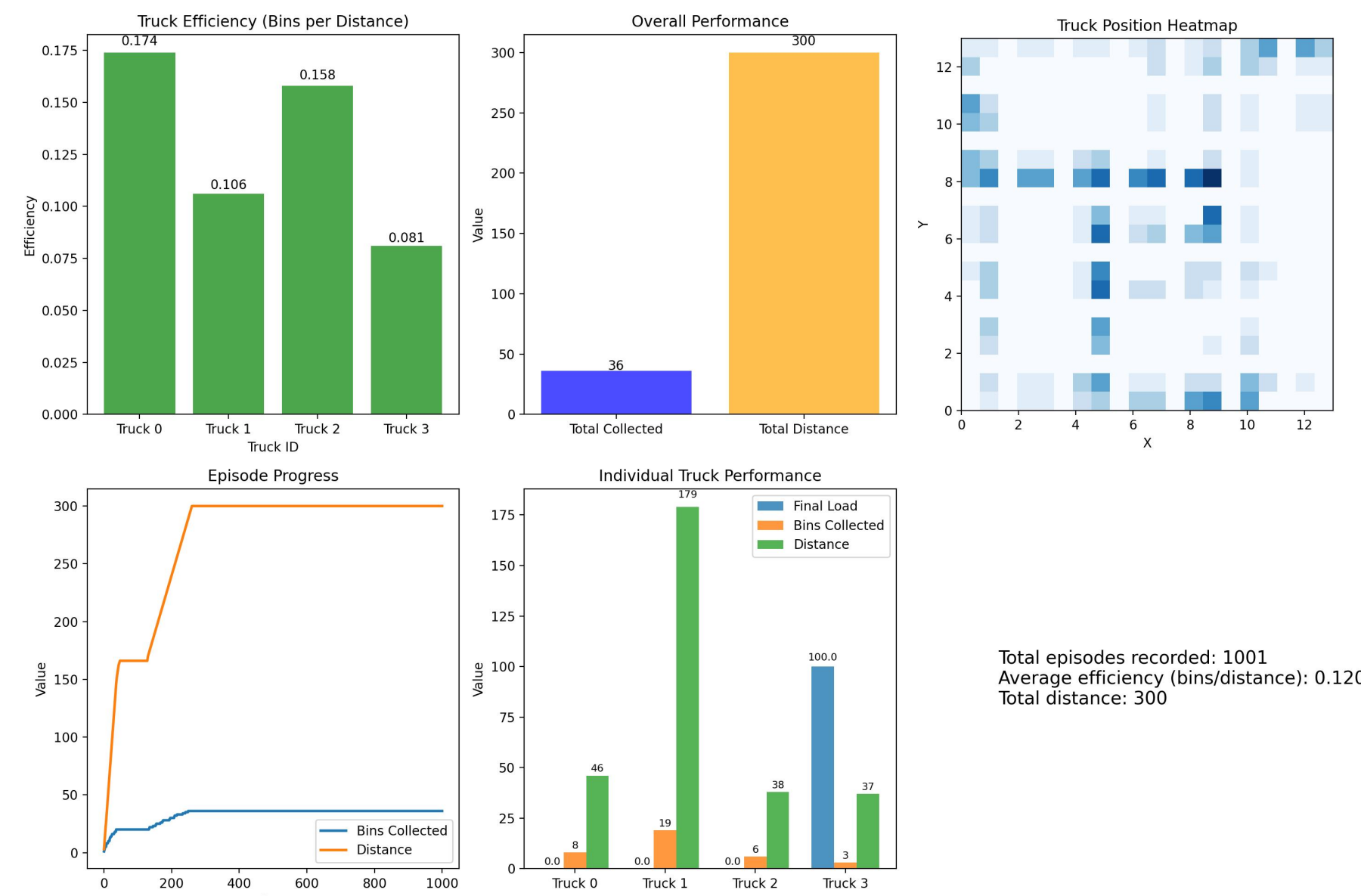


Figura: métricas de eficiencia, desempeño global, mapa de posiciones, progreso del episodio y desempeño individual por camión.

## Trabajo futuro

- Integración de un módulo de coordinación explícita entre agentes.
- Políticas de RL jerárquico o compartido entre camiones.
- Evaluación en mapas urbanos reales o datos sintéticos más complejos.
- Paralelización en GPU de los cálculos de A\* y D\* Lite.

## VIII. Referencias

- Koenig, S., & Likhachev, M. (2002). *D\* Lite*. Proceedings of AAAI.
- Okabe, A., Boots, B., Sugihara, K., & Chiu, S. N. (2000). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction*.