

Planificación de trayectorias con ejecución simultánea para múltiples robots móviles

F. J. Martínez-González, J. González-Ortega, J. E. Rangel-Rodríguez, C.L. Alcocer-Soriano, G. A. García-Chaire, J. P. Ramírez-Paredes¹

¹Departamento de Ingeniería Electrónica, Campus Irapuato-Salamanca
jpi.ramirez@ugto.mx¹

Resumen

La planificación de rutas para múltiples robots móviles, que se desplazan en un escenario común, es un problema que requiere considerar sus cambios de posiciones en el tiempo y que involucra la posibilidad de que ocurran colisiones entre ellos. Este problema surge en aplicaciones tales como el desplazamiento de robots plataforma en almacenes y el uso de múltiples robots en tareas de vigilancia, entre otras. Proponemos una modificación del conocido algoritmo A*, de manera que permita planificar rutas para múltiples robots que se muevan simultáneamente y así evitar la asignación de turnos y tiempos de espera. Mostramos resultados de simulaciones para mostrar que nuestra técnica reduce significativamente el tiempo de ejecución de las trayectorias de múltiples robots, comparado con ejecuciones secuenciales de rutas de costo mínimo.

Palabras clave: planificación de trayectorias, planificación de movimientos, robótica móvil

Introducción

Hoy en día los grandes avances tecnológicos están presentes en gran parte de nuestra vida, debido al gran aumento del comercio electrónico y la necesidad de optimizar la gestión de inventarios, estos avances están afectando los almacenes logísticos. A partir de esto surge la coordinación de múltiples robots de carga en almacenes, lo cual tiene como objetivo diseñar sistemas más eficientes donde los robots puedan manejar mercancías sin intervención humana directa (Tecnología para la Industria, 2024). Para ello, los robots están equipados con una gran cantidad de sensores que les permiten analizar su entorno. Con la información recolectada y procesada por diferentes sistemas, los robots pueden actuar de forma precisa y eficiente.

En el ámbito de la coordinación de robots de carga, estos dispositivos pueden comunicarse y coordinarse entre sí para optimizar el manejo de materiales mediante distintas tecnologías y algoritmos. Los robots pueden intercambiar información en tiempo real sobre distintos parámetros como su posición, velocidad o trayectoria planeada. Esto permite que, en caso de intersecciones o puntos de encuentro, los robots puedan negociar y acordar secuencias de paso y ajustar sus velocidades para mantener una distancia segura, mejorando así la eficiencia del flujo de trabajo del almacén (Tecnología para la Industria, 2024). Lo anterior es una aplicación práctica en la industria del desarrollo de este proyecto, la planificación de trayectorias para múltiples robots. En ambos casos se busca coordinar el movimiento de múltiples robots en un espacio compartido para lograr una operación eficiente y segura.

En el contexto de los vehículos autónomos, la planificación de rutas para múltiples robots implica que cada vehículo determine su trayectoria óptima a seguir, teniendo no solo en cuenta su destino y los posibles obstáculos que pueda haber en el trayecto, sino también las trayectorias de los demás vehículos. Varios retos aparecen al emprender estas tareas de planificación, por ejemplo, la coordinación para evitar colisiones entre robots, la evasión de obstáculos durante la navegación, y el uso eficiente de la energía durante la ejecución de las tareas. Estos modelos usan reglas simples y decisiones locales basadas en lo que sucede a su alrededor, permitiéndoles trabajar juntos de manera efectiva y resolver tareas complicadas sin necesidad de un operador humano (Gil, 2019). En este proyecto se busca lograr que múltiples robots móviles se desplacen por un escenario común, buscando una ejecución eficiente de sus movimientos con trayectorias diseñadas para este fin.

Para resolver esta planificación de trayectorias para múltiples robots han sido usados múltiples métodos como el planteado en (Zhen Yang, 2022), donde se emplea un enfoque basado en un algoritmo de optimización

colaborativa. En este los robots comparten información sobre sus posiciones y trayectorias planeadas en tiempo real, permitiendo una coordinación eficiente y evitando colisiones. Además, utilizan una estrategia de optimización para ajustar sus rutas de manera dinámica, teniendo en cuenta tanto los obstáculos fijos como los móviles. Este enfoque no solo mejora la eficiencia del sistema en términos de tiempo y energía, sino que también asegura que los robots puedan operar de manera autónoma en un entorno compartido, adaptándose a cambios imprevistos en su entorno

Este tipo de coordinación se puede lograr mediante el uso de diversos algoritmos y modelos de autoorganización. Entre los distintos algoritmos que pueden ayudar en el cumplimiento de estas tareas, se encuentra el algoritmo A - star. El algoritmo A* (A - star) es un algoritmo de búsqueda inteligente o respaldado por información que busca el camino más corto desde un estado inicial al estado meta a través de un espacio de problema usando una heurística óptima. La información que utiliza es el costo estimado desde el nodo explorado hacia el nodo destino proporcionado por una función llamada heurística. Como el algoritmo ignora los pasos más cortos, en algunos casos se obtiene una solución subóptima (Lopez-Takeyas, 2005).

El algoritmo propuesto en este proyecto busca encontrar trayectorias óptimas entre varios puntos de salida y meta. Inicialmente, se calcula una estimación de la distancia en línea recta para cada ruta. Posteriormente, para cada par de puntos de salida y meta, se agrega el nodo inicial a una lista abierta y se itera hasta encontrar la meta. En cada iteración, se selecciona el nodo de menor costo y se mueve a una lista cerrada. Si este nodo es la meta, se registra la ruta y los valores de las distancias desde el nodo inicial al actual. Si la distancia recorrida es mayor o igual a la estimada de otras rutas, se dibujan círculos en el mapa para indicar la proximidad de otros robots. Finalmente se verifica y actualiza el camino a los nodos vecinos, asegurando que siempre se elija el camino con menor costo posible. Este enfoque utiliza una búsqueda informada para guiar la exploración y optimización de las trayectorias.

Materiales y Métodos

Como se ha mencionado anteriormente, la elección natural para la planificación de rutas óptimas en grafos conexos es el algoritmo de Dijkstra, o su modificación A* (Corke, 2011). Este algoritmo puede ser utilizado para crear rutas para robots móviles que se desplazan en un mapa de rejilla de ocupación, pero en su forma original no está concebido para manejar múltiples robots o agentes desplazándose simultáneamente (Candra, Budiman, & Hartanto, 2020). Una posibilidad, si la aplicación no limita el tiempo de ejecución, es trazar rutas óptimas con A* ó el algoritmo de Dijkstra para cada agente por separado, y después establecer un orden de ejecución de las rutas, por lo que los agentes seguirían en secuencia cada uno su ruta. Esto disminuye las posibilidades de colisión, aunque si los agentes se ignoran por completo entre sí pueden ocurrir conflictos. Por ejemplo, la posición de destino de un agente puede quedar en el camino de una ruta de costo mínimo para otro. Dicha situación se evita si los planes y su ejecución son secuenciales.

Como alternativa, desarrollamos una modificación multiagente del algoritmo de Dijkstra, que busca generar planes para que los movimientos de los robots sean simultáneos rumbo a su meta. Llamamos a este método PMSim. El costo mínimo incluye penalización por rutas que invaden las posiciones de otros agentes mientras se desplazan. Comenzamos por considerar una jerarquía de prioridad, para evitar situaciones ambiguas donde no se pueda establecer un orden de ocupación de determinadas celdas en el mapa. Por lo tanto, a cada robot se le asigna un ordinal $k = 1, \dots, N$ que lo identifica.

La rejilla de ocupación, representando el entorno en el que se mueven los robots, puede considerarse como un grafo $G = (V, E)$, donde V es el conjunto de los vértices y E el de las aristas. La conectividad de este grafo depende de las vecindades locales de celdas que se definan; los casos más comunes son las conectividades de 4 y 8 vecinos. En nuestra implementación, consideramos el caso de 8 vecinos. En nuestro caso no utilizamos costo heurístico como en A*, ya que nuestro enfoque combinado con dicho costo resultaba en rutas ineficientes.

Para que la planificación de rutas pueda resultar en su ejecución simultánea, modificamos A* de manera que se ejecuten N subrutinas A* secuenciales, donde N es el número de robots o agentes. Por lo tanto, se tienen N grafos, que podemos denotar como G_k , donde $k = 1, \dots, N$.

Durante la planificación, las rutas se van expandiendo un paso a la vez. El agente k usa un G_k donde las posiciones de los $N - k$ agentes previos han sido marcadas como posiciones prohibidas (se removieron los vértices).

Como alternativa, para comparar PMSim, se puede considerar una planificación multiagente secuencial, donde se generen rutas óptimas con A* para cada agente sin considerar más que las posiciones iniciales y finales de los demás. Como es probable que la ejecución de estas rutas involucre colisiones si se intenta de manera simultánea, la solución simple aunque ineficiente es que cada agente ejecute su ruta de manera secuencial, según le corresponda un turno marcado por una lista de prioridad. Llamaremos a esta forma de planificar las rutas PMSec.

Resultados

Para ilustrar de manera comparativa cómo se realiza la planificación multiagente, mostramos una serie de rutas para un escenario de 100x100 celdas, sin obstáculos. En la Figura 1 se aprecia la diferencia entre las rutas diseñadas para su ejecución secuencial PMSec, y las rutas que permiten ejecución simultánea con nuestro algoritmo PMSim. Los puntos rojos indican la posición inicial de cada agente, mientras que los puntos verdes indican su destino. En ambas imágenes se tienen los mismos orígenes y destinos, pero las rutas de la imagen derecha han sido modificadas para que los robots puedan desplazarse simultáneamente sin colisionar entre sí.

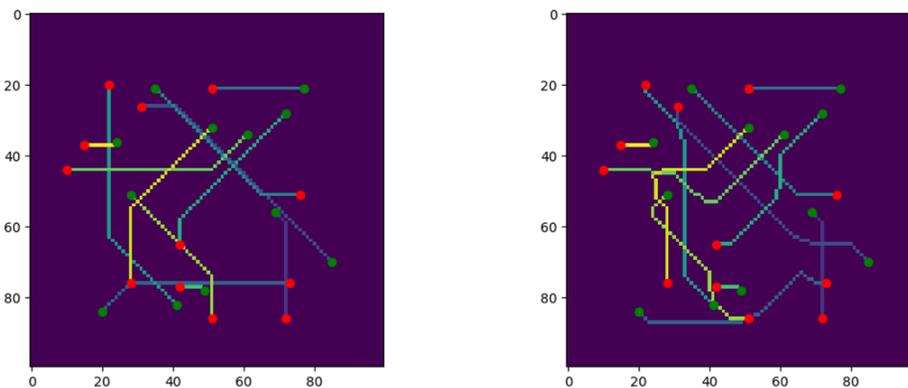


Figura 1. Comparación entre rutas secuenciales y simultáneas, para múltiples robots navegando en un entorno 2D. A la izquierda: rutas de costo mínimo para ejecución secuencial (PMSec). A la derecha: rutas modificadas para permitir la ejecución simultánea (PMSim).

Para verificar el comportamiento del algoritmo desarrollado, y compararlo con otro método, se dispuso un protocolo de pruebas con múltiples robots en un escenario vacío de forma cuadrada. Se emplearon 7 o 9 robots simulados en un entorno de 10x10 metros cuadrados. Dos métodos fueron aplicados para la planificación de trayectorias: el primero es el algoritmo desarrollado en este trabajo (PMSim); el segundo es la ejecución secuencial de rutas de costo mínimo por parte de cada agente (PMSec), lo que garantiza movimientos libres de colisiones, pero requiere que los agentes esperen su turno para desplazarse.

Los resultados, mostrados en la Tabla 1, indican una marcada ventaja en el tiempo de ejecución de las trayectorias por parte del conjunto de agentes al usar PMSim. Para calcular estos tiempos, se especificó una velocidad constante de 0.3 m/s, consistente con las capacidades del robot Pioneer P3DX que se consideró

para la ejecución de las trayectorias. Podemos destacar que el tiempo promedio de ejecución de PMSim en este escenario es desde 75% hasta 79% menor al de PMSec. También se observa que la desviación estándar y el rango son menores para PMSim. El introducir más robots incrementa significativamente el tiempo promedio de ejecución en la estrategia PMSec, al elevarse más de 30% al incluir otros 2 robots, mientras que para PMSim alcanza solo el 10% de incremento.

Tabla 1. Comparación de tiempos de navegación entre métodos, en segundos.

Métrica	Planificación multiagente simultánea (7 robots)	Planificación multiagente secuencial (7 robots)	Planificación multiagente simultánea (9 robots)	Planificación multiagente secuencial (9 robots)
Tiempo promedio de ejecución	24.652	102.616	27.334	134.536
Desviación estándar del tiempo de ejecución	4.474	15.604	4.237	17.870
Tiempo mínimo de ejecución	15.899	63.726	20.233	96.703
Tiempo máximo de ejecución	33.123	129.34	38.765	171.076

Conclusiones

En este proyecto, se ha propuesto y evaluado una modificación del algoritmo de Dijkstra para la planificación de trayectorias de múltiples robots móviles en un entorno compartido. Los resultados que se obtuvieron a partir de múltiples simulaciones demuestran que nuestro método permite una ejecución simultánea de las trayectorias, evitando colisiones y reduciendo de manera significativa el tiempo de ejecución en comparación a una planificación secuencial de las trayectorias.

Las pruebas realizadas en un entorno de simulación sin obstáculos muestran que la planificación simultánea de trayectorias logra reducciones superiores al 75% en el tiempo promedio de ejecución, comparado con la planificación secuencial de trayectorias, además de presentar una desviación estándar y un rango menor en los tiempos de ejecución. Estos resultados evidencian la eficiencia de la planificación simultánea en la coordinación y ejecución de movimientos de múltiples robots. Además, el algoritmo no solo optimiza el tiempo, si no que también introduce una solución para gestionar los conflictos potenciales al priorizar las rutas y marcar posiciones prohibidas. Esta capacidad de manejo eficiente libre de conflictos es crucial para aplicaciones en tiempo real, donde la coordinación precisa y la minimización de tiempos muertos son esenciales.

Como trabajo futuro queda pendiente la adaptación de un costo heurístico al estilo del algoritmo A* para acelerar la convergencia del método, así como el análisis de complejidad en tiempo del algoritmo de planificación para trayectorias simultáneas. Además, se puede explorar la implementación de este enfoque en entornos con obstáculos dinámicos y la incorporación de más factores contextuales que puedan influir en la eficiencia y seguridad de las trayectorias planificadas.

Bibliografía/Referencias

- Candra, A., Budiman, M. A., & Hartanto, K. (2020). Dijkstra's and A-Star in Finding the Shortest Path: a Tutorial. *2020 International Conference on Data Science, Artificial Intelligence, and Business Analytics (DATABIA)* (pp. 28-32). Medan, Indonesia: IEEE.
- Corke, P. (2011). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. New York: Springer.
- Gil, A. R. (2019). Coordinación emergente en sistemas multirobots. *UIS Ingenierías*, 25.
- Lopez-Takeyas, B. (2005, Octubre 17). Algoritmo A* (A-star). Nuevo Laredo, Tamaulipas.
- Tecnología para la Industria. (2024, Enero 25). *Cómo los Robots Colaborativos para Almacenes están Revolucionando la Logística*. Retrieved from Tecnología para la Industria: <https://tecnologiaparalaindustria.com/como-los-robots-colaborativos-para-almacenes-estan-revolucionando-la-logistica/>
- Zhen Yang, J. L. (2022). Path planning and collision avoidance methods for distributed multi-robot systems in complex dynamic environments. *Mathematical Biosciences and Engineering*, 34.