



Tecnológico de Monterrey

Actividad: Roomba

Facundo Gabriel Esparza A01784521

Modelación de sistemas multiagentes con gráficas computacionales
(Gpo 302)

Octavio Navarro Hinojosa

11 de Noviembre del 2024

Índice

Índice	1
Introducción	2
Diseño de los Agentes	2
Objetivo	2
Percepción	2
Proactividad	3
Arquitectura de Subsunción	3
Diseño del Ambiente	3
Estadísticas Recolectadas en las Simulaciones	4
Análisis de Resultados	4
Conclusiones	4

Introducción

Este proyecto aborda el diseño y la simulación de agentes autónomos tipo Roomba en un ambiente de limpieza doméstica. El problema planteado es optimizar la limpieza de una habitación virtual que contiene obstáculos, celdas sucias, y estaciones de carga. La simulación permite estudiar el comportamiento de agentes con baterías limitadas y la capacidad de detectar y evitar obstáculos, recargarse, y limpiar celdas sucias de manera eficiente. La solución propuesta consiste en una arquitectura de agentes basada en principios de subsunción, que prioriza tareas de recarga y limpieza.

Diseño de los Agentes

Los agentes están diseñados como aspiradoras autónomas con una arquitectura de subsunción, en la que cada agente evalúa continuamente las siguientes tareas prioritarias:

1. Recargar batería (si el nivel es bajo).
2. Limpiar celdas sucias cercanas.
3. Explorar áreas del ambiente para maximizar la limpieza.

Objetivo

El objetivo principal de cada agente es maximizar el porcentaje de celdas limpias en el ambiente. Para lograrlo, el agente se mueve por el área buscando celdas sucias, limpiándolas y recargando su batería cuando sea necesario.

Capacidades Efectoras

- **Movimiento:** Los agentes pueden moverse una celda por paso en cualquier dirección cardinal (arriba, abajo, izquierda o derecha).
- **Limpieza:** Los agentes pueden cambiar el estado de una celda sucia a limpia al encontrarse sobre ella.
- **Recarga:** Los agentes pueden recargar su batería al posicionarse en una estación de carga, recuperando 5% de batería por paso.

Percepción

- **Detección de Celdas Sucias:** El agente puede detectar si la celda actual está sucia.
- **Identificación de Estaciones de Carga:** El agente identifica las estaciones de carga en el ambiente, las cuales inicialmente son conocidas o descubiertas durante la exploración.
- **Reconocimiento de Obstáculos:** Los agentes evitan celdas con obstáculos, las cuales son detectadas en celdas adyacentes.

Proactividad

Los agentes son proactivos en la búsqueda de celdas sucias, planificando sus movimientos con un algoritmo A* para encontrar la ruta óptima hacia el objetivo (celda sucia o estación de carga). También priorizan su recarga cuando su nivel de batería baja del umbral predefinido (30%).

Métricas de Desempeño

- **Número de Celdas Limpias:** El porcentaje de celdas limpias al finalizar la simulación.
- **Número de Movimientos:** Cuántos movimientos realiza cada agente durante el tiempo de simulación.
- **Tiempo de Simulación:** El número de pasos requeridos para limpiar completamente o hasta alcanzar el tiempo máximo de ejecución.

Arquitectura de Subsunción

La arquitectura de los agentes sigue un esquema de subsunción que organiza las prioridades de cada agente:

- **Subsistema de Recarga de Batería:** Al llegar al umbral de batería baja, el agente interrumpe la búsqueda de celdas sucias y busca la estación de carga más cercana.
- **Subsistema de Limpieza:** Cuando la batería está por encima del umbral, el agente se centra en encontrar y limpiar celdas sucias. Este subsistema toma prioridad sobre el de exploración.
- **Subsistema de Exploración:** Si no hay celdas sucias visibles, el agente activa el subsistema de exploración, moviéndose por el ambiente para descubrir nuevas áreas.

Diseño del Ambiente

El ambiente simulado es una habitación de tamaño configurable en celdas MxN, que se configura al inicio de cada simulación. Este ambiente es caracterizado por los siguientes elementos:

- **Celdas Sucias:** Distribuidas aleatoriamente según un porcentaje definido en la configuración, representan áreas que el agente debe limpiar.
- **Estaciones de Carga:** Inicialmente ubicadas en posiciones conocidas para el agente, corresponden a lugares donde el agente puede recargar su batería.
- **Obstáculos:** Distribuidos aleatoriamente en el ambiente, representan elementos que el agente debe evitar.
- **Tiempo de Simulación:** Configurado por el usuario mediante un deslizador, limita el número máximo de pasos que puede ejecutar cada simulación.

Estadísticas Recolectadas en las Simulaciones

En cada simulación, se recopilan las siguientes estadísticas relevantes:

- **Porcentaje de Celdas Limpias:** La proporción de celdas que el agente logra limpiar durante el tiempo de simulación.
- **Porcentaje de Celdas Sucias:** La proporción de celdas que permanecen sucias al final de la simulación.
- **Número de Movimientos:** La cantidad total de movimientos ejecutados por cada agente, representando el nivel de actividad y el consumo de batería.
- **Tiempo Total de Ejecución (Pasos):** El número de pasos completados hasta que se limpian todas las celdas o se alcanza el límite máximo de pasos.

Análisis de Resultados

Al analizar los datos de la simulación, se observa que:

- Los agentes son efectivos al limpiar celdas cercanas en ambientes con densidad de obstáculos baja, pero su efectividad disminuye a medida que la cantidad de obstáculos aumenta, lo que reduce su capacidad de navegación.
- Cuando la batería se reduce al umbral crítico, los agentes interrumpen eficazmente la limpieza y se dirigen a la estación de carga, lo que permite maximizar su tiempo activo en el ambiente. Si las estaciones de carga se encuentran lejos de las celdas “sucias” se disminuye la eficiencia de la limpieza para priorizar la carga.
- En simulaciones de múltiples agentes, los agentes colaboran indirectamente cubriendo diferentes áreas del ambiente, aunque a veces pueden intentar ir a limpiar la misma celda o cargarse sobre la misma estación de carga.
- El número de pasos varía en función de la distribución y cantidad de celdas sucias, la cantidad de agentes y la cantidad de obstáculos.

Conclusiones

La simulación muestra que los agentes autónomos tipo Roomba con percepción limitada son efectivos para limpiar ambientes controlados, con un comportamiento adaptativo que prioriza la recarga y limpieza. Sin embargo, factores como la densidad de obstáculos y la distribución de celdas sucias afectan su desempeño. Es importante tomar en cuenta diversos factores al desarrollar este tipo de proyectos, como los algoritmos para encontrar las mejores rutas, el factor de que haya más de 1 agente y como hacer uso eficiente de su posible colaboración.

Conclusión personal: Este proyecto me obligó a profundizar en el conocimiento de algoritmos de navegación y planificación aplicados a agentes autónomos, además de presentarme escenarios bastante desafiantes, como el uso de la biblioteca Mesa, que era completamente nueva para mí. Me

gustó especialmente la oportunidad de agregar un toque personal mediante la personalización del modelo y la creatividad en la forma de resolver algunos problemas. Este proceso no solo me ayudó a entender los conceptos básicos de la simulación multiagente, sino también a desarrollar soluciones específicas para optimizar el rendimiento y adaptabilidad de los agentes en un ambiente complejo.