



Tecnológico de Monterrey

Modelación de Sistemas Multiagentes con Gráficas Computacionales

Reto

Diego Abdo Velásquez A01783808

Tomás Molina Pérez Diez A01784116

29 de noviembre del 2024

Problema a Resolver

Hay una ciudad dada por un archivo txt, cada n pasos de la simulación entran 4 coches a la ciudad por las esquinas. Cuando todas las esquinas estén bloqueadas por otros coches, la simulación para. Cada coche tiene un destino al cual llegar, este tiene que seguir la reglas de circulación (obedecer semáforos, seguir el sentido de las calles y no chocar con otros coches). Estos se tienen que mover de forma eficiente para no generar tráfico y que no se bloqueen las esquinas.

Propuesta

Tenemos que programar el comportamiento de los coches para que sea eficiente y lleguen a sus destinos de la forma más eficiente posible. Al leer el mapa, se crea un grafo (con BFS) de las calles, incluyendo semáforos. El grafo lo guardamos como un diccionario de adyacencia, donde la llave es la posición y el valor es una lista con los posibles movimientos desde esa celda, incluyendo cambios de carriles y posibles vueltas a la derecha e izquierda.

Cuando se inicializa el coche, se le da un destino randomizado y se usa BFS sobre el grafo para buscar la manera óptima de llegar a su destino. Usamos memoización para no repetir cálculos y optimizar el programa. Esto le da al coche su ruta en forma de una lista de los movimientos a seguir.

El coche sigue sus movimientos, a menos de que esté sobre un semáforo en rojo o haya un coche en la celda a la que se quiere mover. En este caso el coche se espera. Si hay coche enfrente, checa si se puede cambiar de carril, checa las celdas en diagonal en la dirección que se estaba moviendo, si estas están libres, se cambia de carril y recalcula su ruta desde esa nueva celda.

Al llegar a su destino, el agente deja de existir.

Diseño del Agente

Objetivo

El objetivo de cada agente es llegar a su destino.

Capacidad efectora

El agente afecta el ambiente ocupando las celdas. Pues si él la está ocupando, otro agente no puede pasar por ahí, creando tráfico.

Percepción

El agente conoce el mapa y así puede calcular su ruta, pero el agente no conoce la posición de otros agentes dentro del mapa, destinos, posición de semáforos, etc. Solo conoce lo que está en su posición y lo que está en la celda a la que se quiere mover.

Proactividad

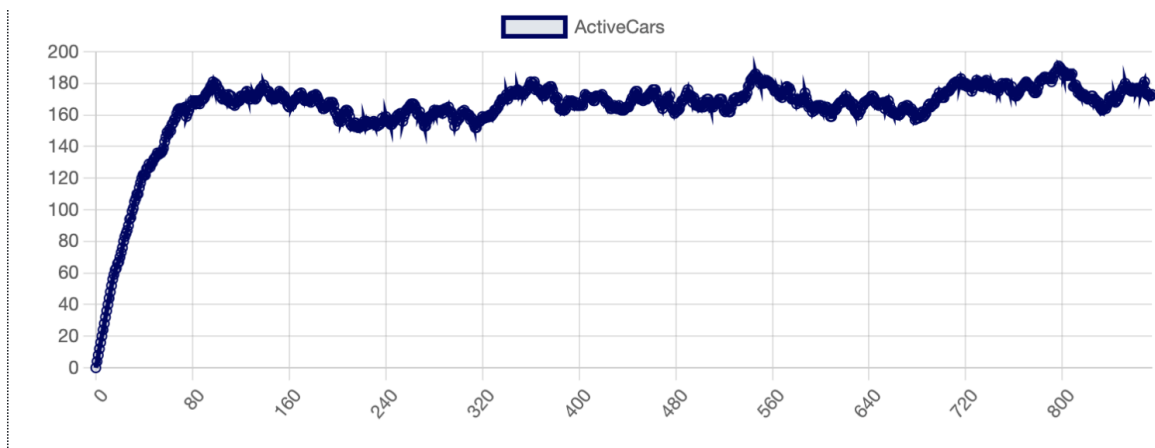
El agente calcula su ruta a su destino y por cada step se mueve (si es posible) hacia el.

Reactividad

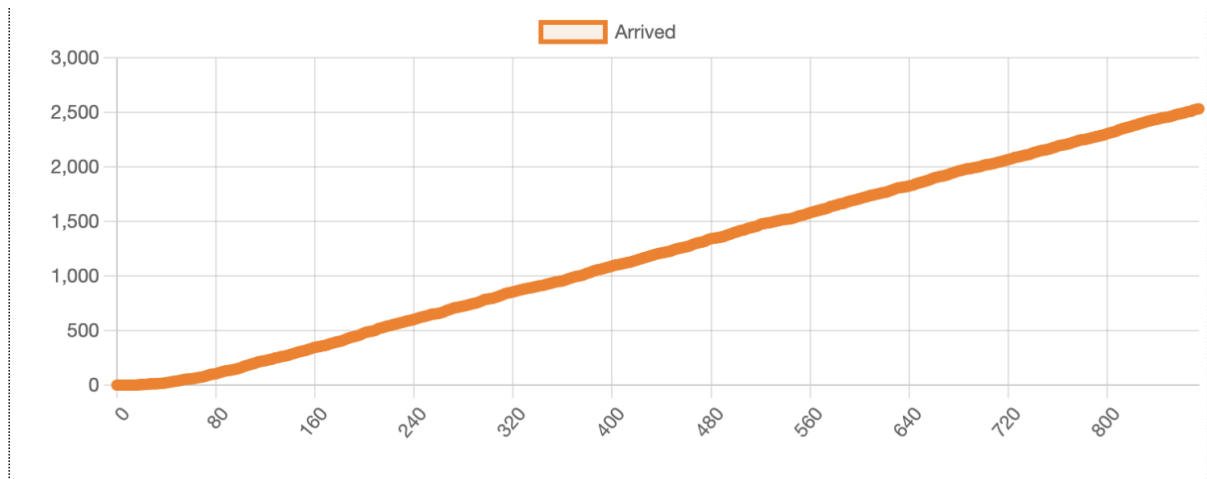
El agente reacciona si está sobre un semáforo en rojo, en el caso que se queda parado y espera que se ponga en verde. También reacciona en el caso de que tenga un coche enfrente. En este caso intenta cambiar de carril, si es posible se cambia de carril, sino se queda parado y espera que se mueva el coche enfrente.

Métricas de desempeño

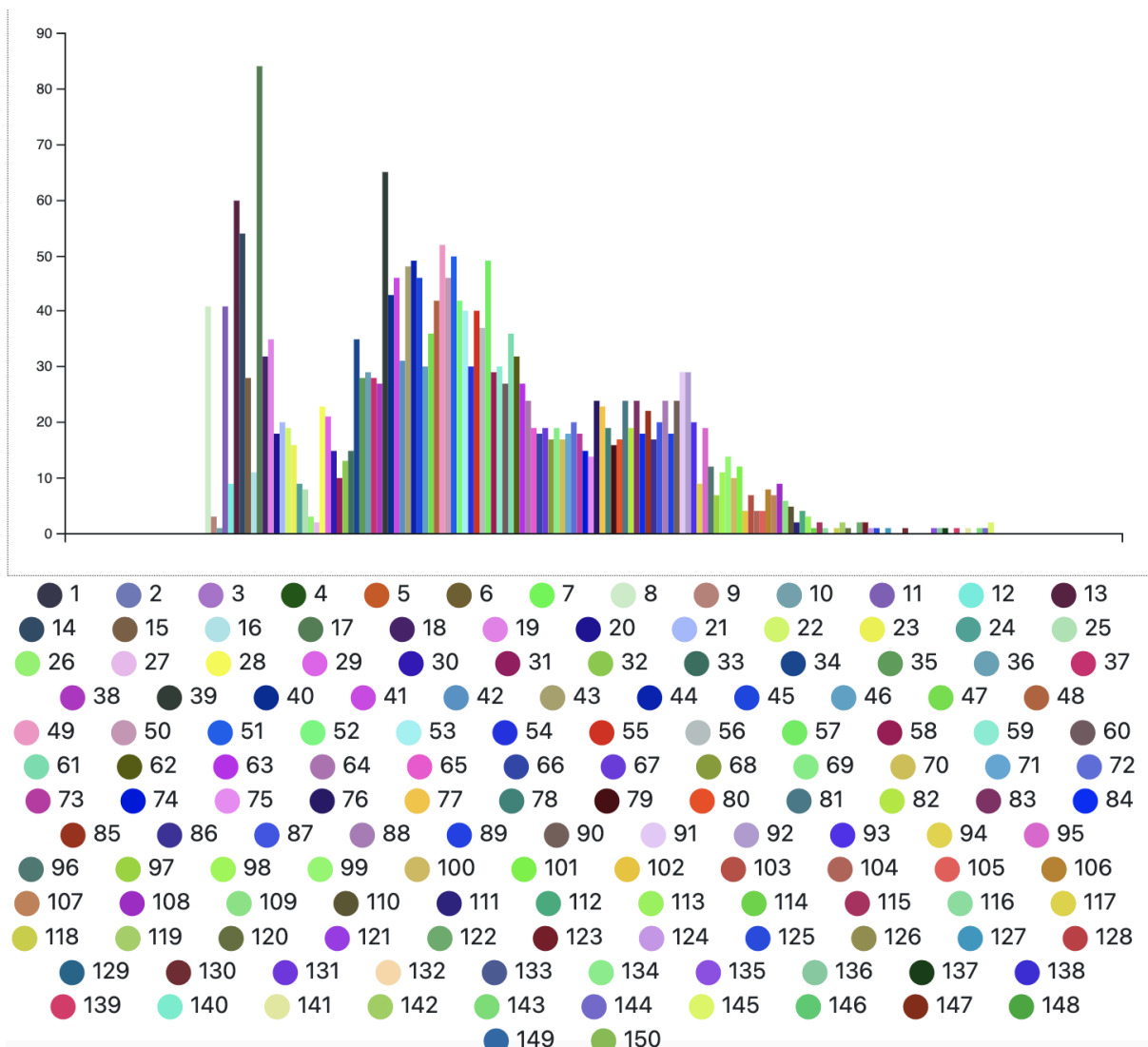
La métrica de desempeño es cuántos pasos le tomó llegar a su destino, mientras más tráfico haya se va a tardar más. Igual se mide si uso memoización para calcular sus rutas, viendo lo eficiente que está siendo el modelo. Al igual cuántos coches hay en cada momento de la simulación y los pasos antes de que se active la condición de parada de la simulación.



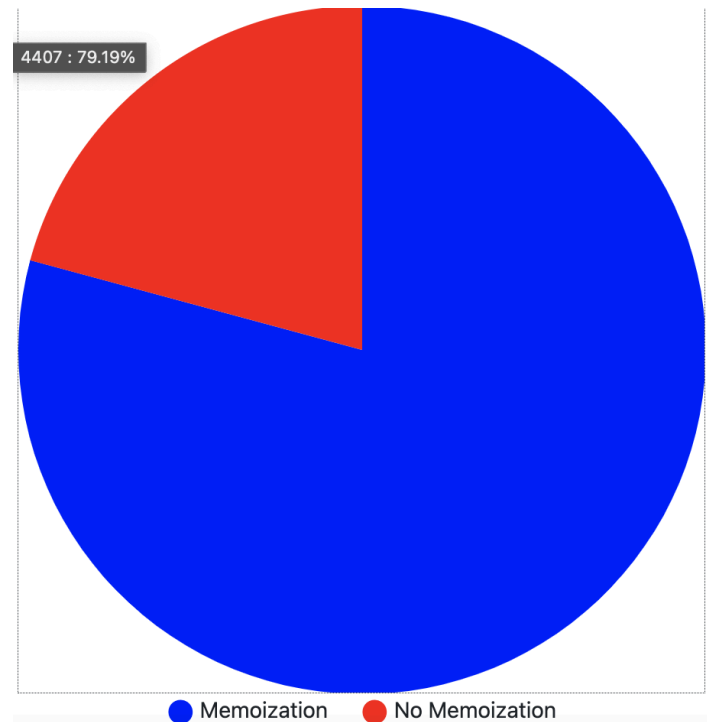
Gráfica mostrando cuántos coches activos hay, en este ejemplo se agregaron 4 coches cada step. Se puede observar que se mantiene en equilibrio entre 180 y 200 coches a la vez.



Gráfica mostrando los coches que ya llegaron a su destino. Parece completamente lineal, con un inicio lento, lo cual tiene sentido.



Distribución de cuantos steps se tardan en llegar a su destino los coches, pensamos que seguiría una distribución normal pero no se aprecia esa tendencia. Se puede observar que el modelo está trabajando eficientemente pues no se están tardando tanto los coches en llegar a sus destinos.



Gráfica que muestra el porcentaje de rutas calculadas con memorización y sin, después de alrededor de 900 steps. Mientras más steps pasan más se usa la memorización haciendo al modelo más eficiente con el tiempo.

Habilidad Social

Los agentes no tienen habilidad social per se.

Racionalidad

El agente no tiene racionalidad solo sigue su ruta, si está disponible y no hay semáforos en rojo. Si no se intenta cambiar de carril y sigue su ruta.

Arquitectura de subsunción de los agentes

1. Moverse

La acción con menos prioridad es la de moverse en dirección hacia el destino porque todas las demás toman prioridad antes de esta.

2. No chocar

Antes de moverse el coche siempre tiene que checar primero si hay coche a donde se quiere mover, si hay coche tiene que ver si se puede cambiar de carril para seguir avanzando y si no tiene que esperar.

3. Semáforo

Si el agente está en un semáforo la prioridad es que el agente cheque si ese semáforo está en rojo, si está en rojo el agente tiene que frenar hasta que el semáforo se ponga verde.

4. Checar Destino

Si el coche está en su destino entonces la prioridad número 1 siempre será remover ese coche de la simulación porque ya terminó su recorrido.

Características del Ambiente

Inaccesible

El ambiente es accesible en cuanto a los agentes (coches) porque tienen información sobre su entorno como la posición del destino al que tienen que llegar, pueden ver los semáforos cuando llegan a ellos, pueden detectar el sentido de las calles, etc. Aunque también tienen algunos elementos inaccesibles porque los coches no conocen el estado completo del mapa.

No Determinista

El ambiente es no determinista porque el resultado de las acciones de los agentes depende de varios factores como el estado de los semáforos, otros coches, congestiones, etc. Esta incertidumbre implica que las mismas condiciones no siempre dan el mismo resultado.

No Episódico

El ambiente es no episódico porque los agentes solo pueden actuar en base a su estado actual, los agentes no pueden hacer acciones en base a lo que sucederá varios steps en el futuro, es decir, los agentes no pueden ver el futuro resultado de sus acciones.

Dinámico

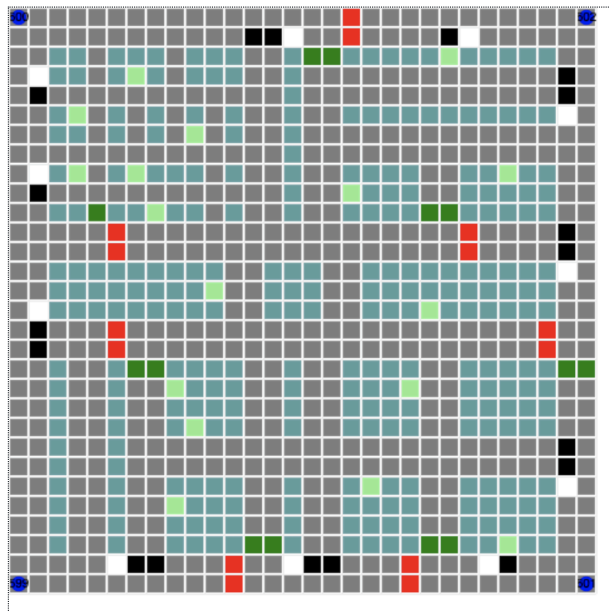
El ambiente es dinámico porque constantemente se agregan y se quitan coches, además de que los semáforos constantemente están cambiando de estado.

Discreto

El ambiente es discreto porque los agentes tienen un número finito de acciones y percepciones, sólo pueden hacer un cierto número de cosas dentro del ambiente.

Scheduler

Creamos nuestro propio scheduler para optimizar nuestra simulación, haciendo que se congestionen menos las calles. La idea es encontrar los “danger squares” los cuales son las celdas donde se pueden dar vuelta a la izquierda en las calles exteriores. Y los “pre danger squares” son las celdas atrás de esto, dependiendo de la dirección de la calle.



Los cuadros negros son los danger squares y los blancos son los pre danger squares. Nuestro scheduler, sigue el siguiente orden al activar los agentes: Semáforos, coches en danger squares, el resto de coches (en orden que se agregaron al scheduler) y al final coches en pre danger square.. De esta forma garantizamos que los coches que están en el carril exterior y quieren dar la vuelta a la izquierda lo logran siempre.

Al final cambiamos a solo considerar los danger zones del lado derecho, porque solo es necesario ahí.

Conclusiones

La simulación de tráfico que implementamos modela de manera eficiente el flujo de tráfico de coches en una ciudad, resolviendo el problema de la congestión y cumplimiento de las reglas de tráfico. Esto lo hacemos agentes que calculan rutas óptimas utilizando BFS y memorización. Los coches respetan semáforos, evitan colisiones y recalculan rutas en tiempo real, lo que garantiza un comportamiento realista y eficiente en un ambiente dinámico y no determinista.

Aunque los agentes no tienen habilidades sociales o comunicación, lo cual podría limitar su desempeño en situaciones más complejas, el modelo es suficientemente robusto para simular ciudades grandes y evaluar formas de manejar el tráfico.