



Tecnológico de Monterrey

Modelación de sistemas multiagentes con gráficas computacionales (Gpo 302)

Semestre Agosto - Diciembre 2023

Reto:

Reto Movilidad Urbana

Profesores:

Gilberto Echeverría Furió

Octavio Navarro Hinojosa

Equipo 8

Name	ID	Career	Campus
David Flores Becerril	A01368391	ITC	Santa Fe
Juan Pablo Ruiz de Chávez Diez de Urdanivia	A01783127	ITC	Santa Fe

Fecha de Entrega:

30 de noviembre del 2023

Problema que se está resolviendo, y la propuesta de solución.

La movilidad urbana, que implica la capacidad de desplazarse de un lugar a otro, es esencial para el desarrollo económico y social, así como para la calidad de vida en las ciudades. A lo largo del tiempo, la conexión entre movilidad y el uso del automóvil ha sido considerada como un indicador de progreso, pero esta asociación ya no es sostenible. El crecimiento desmedido del uso de automóviles, respaldado por políticas públicas mal enfocadas en movilidad sostenible, está generando impactos negativos significativos en los ámbitos económico, ambiental y social en México.

En las últimas décadas, se ha observado un preocupante aumento en el uso de automóviles en el país. Los Kilómetros-Auto Recorridos (VKT) se han triplicado, pasando de 106 millones en 1990 a 339 millones en 2010. Este aumento se ha traducido en efectos negativos como contaminación atmosférica, accidentes, enfermedades y congestión vehicular.

Para que México pueda alcanzar su potencial como una de las economías más grandes del mundo, es imperativo abordar el problema de la movilidad en las ciudades. Por lo que se desarrolló una solución que reduce la congestión vehicular mediante la simulación gráfica del tráfico, representando la interacción de un sistema de múltiples agentes.

Como parte de esta solución se implementa el uso de multiagentes, como se había comentado anteriormente, en un ambiente de desarrollo de programación en donde se simulan los comportamientos de distintos elementos dentro de un ambiente por medio de la librería Mesa de Python. Este modelo se podrá replicar en Unity a través del uso de un API de la librería de Flask. De esta manera tendremos un modelo en 3D en donde esperamos ver diversos comportamientos cercanos a la realidad en un ambiente arbitrario, para a través del mismo poder diseñar mejores vías urbanas para los ambientes que lo necesiten.

El diseño de los agentes (objetivo, capacidad efectora, percepción, proactividad, métricas de desempeño, etc.).

Agentes:

Agente Car (Carro)

El agente carro no tiene la capacidad de modificar el ambiente, sin embargo tiene un repertorio de acciones que serán utilizadas dependiendo de sus necesidades. Tiene la **capacidad efectora** de moverse a una nueva posición en función de su destino y la presencia de obstáculos, como otros autos y semáforos. Es **proactivo** en el sentido de que elige un destino aleatorio y trata de moverse hacia él. En este caso la prioridad es un atributo con fuerte importancia ya que dichos agentes presentan comportamientos de iniciativa en el trayecto a su destino. También considera la presencia de semáforos y otros autos para evitar colisiones. Es **reactivo** ya que reacciona a la presencia de otros autos y semáforos en su entorno. Se detiene si el semáforo está en rojo o si hay un auto frente a él. En cuanto a habilidad social, el agente no tiene comunicación directa con otros agentes de su tipo, sin embargo puede conocer sus posiciones por medio de la alteración de los pesos en los edges en los grafos y también la posición de sus celdas vecinas.

Mecanismos:

- Percepción de otros semáforos por medio de sensores apuntando a $x > \text{neighbor}[0]$, $x < \text{neighbor}[0]$, $y < \text{neighbor}[1]$, $y > \text{neighbor}[1]$, dependiendo de la dirección de movimiento con $\text{moore} = \text{False}$ (Vecinos en esquinas no se encuentran)
- Percepción de vehículos por medio del cálculo de $A^*(\text{network}x)$
- Alteración del peso de un nodo
- Percepción de vehículos por medio de función de cálculo de next step
- Cálculo de ruta y movimiento por medio de $A^*(\text{network}x)$

Agente TrafficLight (Semáforo)

El agente semáforo, tiene la capacidad de modificar el ambiente, en el sentido que maneja la cantidad de tráfico que se genera, pues de cierta forma afecta a la ruta de que los carros toman de una manera no directa, debido a que el peso de un nodo se define dependiendo de si sobre la misma se encuentra un agente de tipo carro.. Puede cambiar su estado (verde/rojo) después de cierto número de pasos. En cuanto reactividad y proactividad realmente no se presentan dichas características ya que el semáforo no cuenta con percepción de su entorno, sino que simplemente cumple con su función alternando entre un estado de alto o siga a través del tiempo.

Mecanismos:

- Cambio de de estado en periodo de tiempo determinado (Rojo, verde)
- Estado perceptivo a agentes carro (por medio de los sensores del carro)

Agente Road (Calle)

El agente calle se utiliza como una base para crear el grafo. Dentro de los atributos del agente calle se encuentra la dirección de la misma. Esto sirve para hacer el trazado del grafo en función de dicha dirección definida en su diccionario. La ruta de las calles se calcula por medio de los nodos del grafo ya creados, por lo que realmente este tipo de agente no cuenta con propiedades de reactividad o proactividad. La única capacidad perceptiva con la que cuenta el agente es el hecho de saber si un agente de tipo carro se encuentra sobre él, para de esta manera actualizar los pesos del nodo generado en dicha coordenada.

Mecanismos:

- Detección de agentes carro para actualización del peso del nodo

Agente Obstacle (Obstáculo)

Este agente, al igual que Road y TrafficLight no es considerado como un agente inteligente, puesto que no cuenta con las características de proactividad, reactividad o actividad en sociedad. Este tipo de agente únicamente se incluye en la elaboración del mapa para definir zonas inaccesibles para el coche. No cuenta con mecanismos.

Métricas de desempeño:

Debido a que los agentes calle, semáforo y obstáculo no son agentes inteligentes, no cuentan con métricas de desempeño, por lo que estas solo son establecidas para el agente carro.

Métricas agente carro:

- El agente llega a su destino, se marca como arribó y se elimina del modelo
- El agente crea la ruta menos pesada hasta el destino
- No se genera tráfico por medio de la proactividad de cada agente
- La entrada y salida de vehículos en el modelo alcanza un punto de equilibrio

El agente posee la **habilidad social** de interactuar con el agente semáforo (traffic light) según su estado para seguir avanzado por la ruta (estado = True = Verde), o pararse (estado = False = Rojo). Recordemos que un semáforo sirve para regular el flujo de vehículos y peatones en las vías, de esta manera se evitan accidentes de tráfico, colisiones y congestionamientos, manteniendo la seguridad vial.

El agente es **racional** porque sabe cuál es su destino dentro del mapa, y siempre tomará la ruta más óptima a este utilizando un algoritmo avanzado de búsqueda: A*. Este algoritmo busca el camino más corto desde un estado inicial al estado meta a través de un espacio de problema. Sin embargo, al no considerar las rutas más “largas” en algunos casos rinde una solución subóptima, debido a que no podría encontrar una ruta diferente. Este último problema se soluciona actualizando el peso de los nodos sobre donde se encuentran los agentes carro, para de esta manera incentivar al coche a tomar vías alternas con menor flujo de tráfico.

Características del ambiente.

Ambiente:

El agente principal (carro) desempeña su función en un ambiente **inaccesible** debido a que no conoce la totalidad del entorno en el que se encuentra. El agente sólo tiene acceso a sus coordenadas destino y la ruta más corta hacia éste; proporcionada mediante un grafo dirigido que contiene la dirección en la que se puede mover sobre las calles del mapa y también. Al agente se le agrega información progresivamente mientras recorre el mapa. La información correspondiente será:

- Estado de las celdas vecinas (Para moverse a esa posición)
- Posición de agentes carro vecinos
- Estado de los semáforos (Para seguir moviéndose o no)
- Nuevos pesos de nodos dependiendo del tráfico

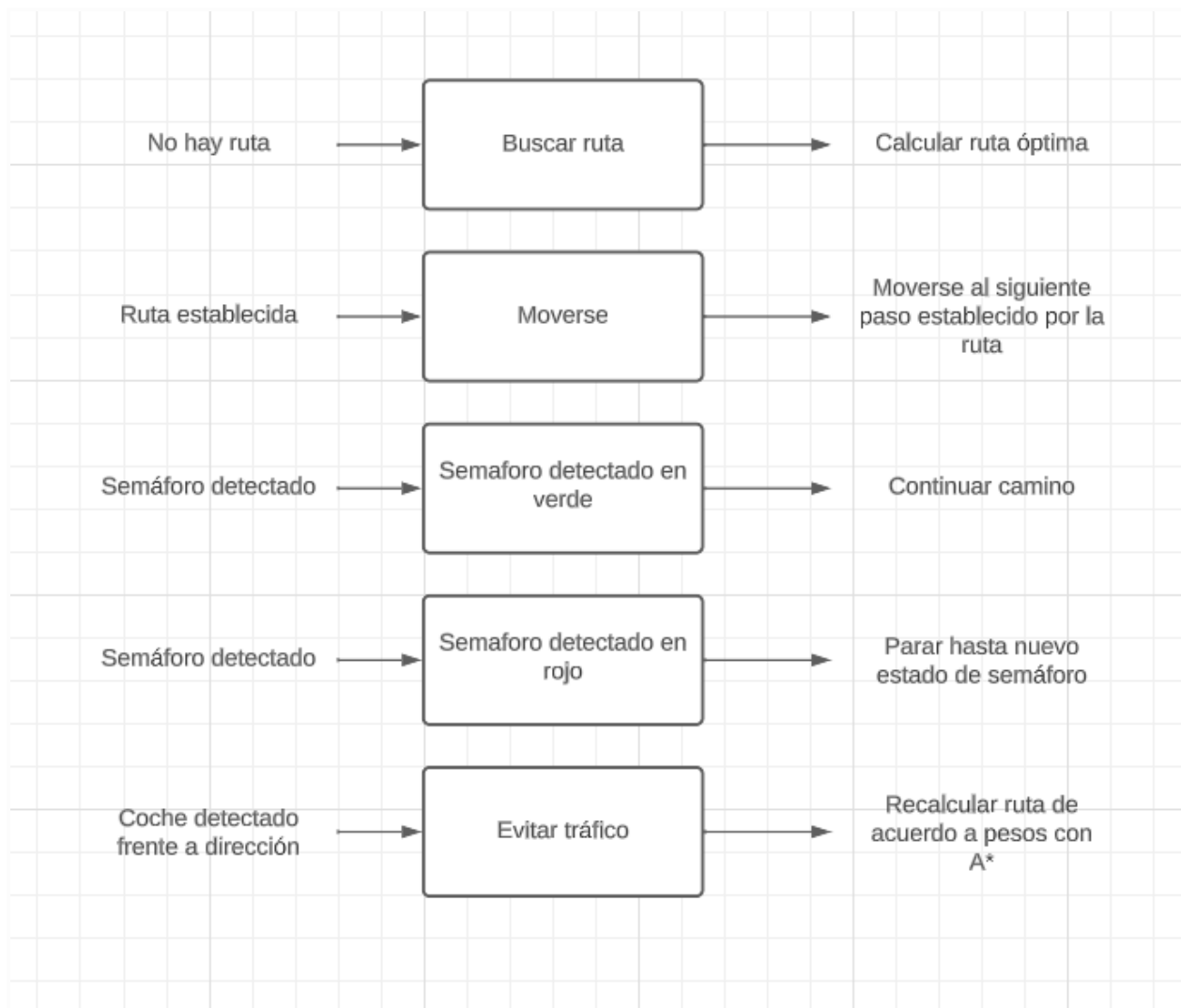
El ambiente es **determinista** porque las reglas que deberá seguir el automóvil para llegar a su edificio destino no cambiarán a lo largo de la simulación. No obstante el ambiente es **dinámico** porque aunque los coches tengan un punto de salida y llegada predefinidos, estos valores se tomarán de una manera aleatoria de un repertorio de llegadas y salidas. De igual manera, la ruta que tomará cada coche para cada destino será diferente para cada iteración de la simulación, debido al cambio de estado de cada semáforo y la cantidad de tráfico acumulada por estos mismos.

Se trata de un ambiente **discreto**, ya que los cálculos realizados para moverse entre casillas dan como resultado números enteros. El agente desempeña su labor en un espacio finito con número de acciones finitas y no tiene que depender de acciones intermedias para llevar a cabo muchas otras.

El ambiente mostrado es **episódico**, porque el agente se mueve de acuerdo a las acciones realizadas anteriormente haciendo un step para cada movimiento en el grid.

La arquitectura de subsunción de los agentes.

La única clase de agentes inteligentes es el carro, puesto que el semáforo únicamente tiene un cambio de estado definido por tiempo sin ningún tipo de jerarquía ya que este es su único comportamiento. Mismo caso que se presenta en el agente calle, ya que cuenta con una única capa que comprueba si hay un agente carro sobre él para actualizar el peso del nodo. Sin embargo, el agente carro al ser un agente inteligente presenta la siguiente arquitectura:



Conclusiones:

A través de los sistemas computacionales tenemos la capacidad de simular problemáticas reales con una gran precisión, para de esta manera analizar comportamientos y áreas de oportunidad dentro de una posible solución. Los sistemas multiagentes nos permiten adentrarnos en diferentes escenarios en los cuales podemos definir diversos comportamientos de los agentes en cuestión dentro de un ambiente determinado. De esta manera podemos tener un acercamiento a situaciones específicas en donde el pensamiento científico entra en juego para buscar una solución óptima para el escenario en cuestión.

Como parte de este proyecto nos dimos cuenta de que dentro de la simulación tanto el diseño de las vialidades como los comportamientos de los agentes forman un papel muy importante en el manejo de tráfico en la ciudad. El hecho de tener agentes poco proactivos genera congestiones de manera continua por lo que se vuelve poco eficiente el modelo. Por otro lado, el tener calles mal conectadas igualmente puede causar embotellamientos.

Por medio de diversos intentos notamos que una alta proactividad de los agentes igualmente puede resultar contraproducente a la hora de la simulación, ya que en el momento en el que todos los vehículos quieren pasar existan aglomeraciones en puntos específicos, que efectivamente se desatascan en un punto, pero podría suceder antes o incluso no congestionarse de ninguna manera. Una de las mejores maneras de poder hacer un código eficiente es pensar en las situaciones tal y como las tomaría un humano real. Muchas veces las decisiones o comportamientos que tomarán los agentes resultan en una sorpresa, por lo que el pensamiento y posteriormente la prueba y error crean un modelo de trabajo sumamente efectivo.

Referencias

1. Handy, Susan. (2002). Accessibility- Vs. Mobility-Enhancing Strategies for Addressing Automobile Dependence in the U.S. European Conference of Ministers of Transport. Retrieved on February 21, 2012, from http://www.des.ucdavis.edu/faculty/handy/ECMT_report.pdf Links to an external site.
2. Medina Ramírez, Salvador. (2012). Transforming Urban Mobility in Mexico: Towards Accessible Cities Less Reliant on Cars. Institute for Transportation and Development Policy (ITDP Mexico). Retrieved on August 7, 2019, from <http://mexico.itdp.org/wp-content/uploads/Transforming-Urban-Mobility-in-Mexico.pdf> Links to an external site.
3. Eafit, U. (n.d.). *Semáforos*. www.eafit.com.
<https://www.eafit.edu.co/ninos/reddelaspreguntas/Paginas/como-funcionan-los-semaforos.aspx>

4. Instituto Tecnológico de Nuevo Laredo. (n.d.). ALGORTIMO A*. In *Inteligencia Artificial* (no).

Retrieved November 28, 2023, from

[https://nlaredo.tecnm.mx/takeyas/Apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/A-Star/A-Star\(2005-II-A\).pdf](https://nlaredo.tecnm.mx/takeyas/Apuntes/Inteligencia%20Artificial/Apuntes/tareas_alumnos/A-Star/A-Star(2005-II-A).pdf)