

Proyecto de Curso

Simulación Basada en Agentes de Tráfico Urbano con Programación Paralela

Curso de Computación Paralela y Simulación

31 de enero de 2026

1. Introducción

La simulación basada en agentes (Agent-Based Modeling, ABM) es una metodología ampliamente utilizada para estudiar sistemas complejos en los cuales el comportamiento global emerge a partir de interacciones locales entre entidades autónomas.

El tráfico urbano constituye un ejemplo clásico de este tipo de sistemas, donde vehículos individuales interactúan con otros vehículos y con la infraestructura vial, generando fenómenos como congestión, cuellos de botella y variaciones dinámicas del flujo.

A medida que aumenta el número de agentes y la resolución espacial, este tipo de simulaciones presenta una alta demanda computacional, convirtiéndose en un escenario ideal para la aplicación de técnicas de programación paralela.

En este proyecto se desarrollará una simulación basada en agentes de tráfico urbano y se implementará una versión paralela del modelo con el fin de analizar tanto el comportamiento del sistema como su desempeño computacional.

2. Objetivo general

Diseñar, implementar y analizar una simulación basada en agentes de tráfico urbano, incorporando técnicas de programación paralela para mejorar el rendimiento computacional.

3. Objetivos específicos

- Construir un modelo secuencial de tráfico urbano basado en agentes.

- Definir reglas locales de movimiento e interacción entre vehículos y semáforos.
- Implementar una versión paralela del modelo.
- Analizar métricas del sistema de tráfico.
- Evaluar el desempeño computacional de las versiones secuencial y paralela.

4. Modelo del sistema

4.1. Representación espacial

El entorno de simulación se modelará como una rejilla bidimensional de celdas.

Cada celda puede representar:

- Un segmento de calle transitable
- Una intersección con semáforo
- Un espacio no transitable (bloque urbano)

Las calles se representan como secuencias contiguas de celdas transitables, pudiendo tener múltiples celdas de longitud entre intersecciones consecutivas.

Asuma que todas las calles son bidireccionales, así que la capacidad de cada celda son dos vehículos, uno en cada dirección.

Las intersecciones aparecen únicamente donde se cruzan calles perpendiculares y son los únicos puntos donde existen semáforos.

4.2. Agentes

Cada agente representa un vehículo e incluye como mínimo:

- Posición en la rejilla
- Dirección de movimiento
- Estado de avance o detención
- El número de agentes es un parámetro
- Los agentes deben ser ubicados en lugares aleatorios

4.3. Semáforos

Cada intersección posee un semáforo que alterna el paso de vehículos en direcciones perpendiculares.

El cambio de estado puede ser periódico o adaptativo según la congestión local.

5. Reglas del sistema

En cada paso de tiempo de la simulación:

1. Los semáforos actualizan su estado.
2. Cada vehículo:
 - Avanza a la siguiente celda si está libre y el semáforo lo permite.
 - Se detiene si la celda siguiente está ocupada o el semáforo está en rojo.
 - En intersecciones puede cambiar de dirección con cierta probabilidad.

6. Especificación externa de la rejilla

El programa deberá leer la configuración espacial desde un archivo de texto externo.

No se permite definir la rejilla directamente en el código fuente.

6.1. Símbolos utilizados

Símbolo	Significado
.	Segmento de calle transitable
+	Intersección con semáforo
#	Espacio no transitable

6.2. Reglas de consistencia

- Las calles se representan únicamente con el símbolo .
- Las intersecciones se representan con +
- Una intersección debe coincidir con el cruce de dos calles
- Los vehículos solo pueden moverse por celdas . y +
- Las celdas # son intransitables

6.3. Ejemplo de archivo de rejilla

```
#####
#####+.....+#####
#####.#####.#####.#####
#+.....+....+....+....+#
#.#####.####.####.####.#####
#.#####.####+....+####.####.#####
#.#####.####.####.####.#####
#+.....+....+....+....+#
#####.#####.#####.#####.#####
```

6.4. Interpretación

- Los símbolos . forman calles largas entre intersecciones
 - Los símbolos + representan semáforos en cruces
 - Los bloques # representan zonas urbanas no transitables

7. Implementación paralela

Los estudiantes deberán implementar al menos una de las siguientes estrategias:

7.1. Paralelización espacial

La rejilla se divide en subregiones, donde cada proceso o hilo gestiona una porción del entorno.

Se deberá intercambiar información en las fronteras para garantizar coherencia en el movimiento de vehículos.

7.2. Paralelización por agentes

Los agentes se distribuyen entre hilos o procesos para actualizar su estado concurrentemente.

Se deberá manejar adecuadamente la sincronización para evitar condiciones de carrera.

8. Métricas

8.1. Métricas del sistema

- Flujo promedio de vehículos
- Número de vehículos detenidos

8.2. Métricas computacionales

- Tiempo de ejecución secuencial
- Tiempo de ejecución paralelo
- Speedup
- Eficiencia

9. Requisitos técnicos

- El modelo secuencial debe ser completamente funcional antes de parallelizarse.
- La versión paralela debe mostrar mejora de rendimiento para simulaciones de tamaño significativo.
- Se deben incluir mediciones de tiempo de ejecución.

El lenguaje y las librerías quedan a elección del estudiante.

10. Entregables

1. Código fuente del modelo secuencial
2. Código fuente del modelo paralelo
3. Informe técnico de entre 5 y 8 páginas
4. Gráficas de comportamiento del sistema y desempeño computacional

11. Criterios de evaluación

Componente	Peso
Modelo ABM funcional	25 %
Implementación paralela	30 %
Análisis de desempeño	20 %
Informe técnico	15 %
Calidad del código	10 %

12. Extensiones opcionales

- Semáforos adaptativos según congestión
- Generación dinámica de vehículos
- Accidentes o bloqueos temporales
- Diferentes tipos de vías o velocidades