

Modelado y Simulación de una Intersección de Calles en un Contexto Multi-Agente

Modelling and Simulation of a Street Intersection in a Multi-Agent Context

Joaquín Nacht ^{#1}, Mariana Falco ^{*2}, Gabriela Robiolo ^{#3}

[#] LIDTUA (CIC), Facultad de Ingeniería, Universidad Austral,
Pilar, Buenos Aires, Argentina

¹ joaquin.nacht@ing.austral.edu.ar

³ grobiolo@austral.edu.ar

^{*} CONICET - LIDTUA, Facultad de Ingeniería, Universidad Austral,
Pilar, Buenos Aires, Argentina

² mfalco@austral.edu.ar

Recibido: 05/11/18; Aceptado: 29/11/18

Abstract—Losses of time due to congestion and traffic jams are current events in cities due to a non-optimal configuration of the traffic lights. Various sciences and specialties have tried to understand the phenomenon and identify the causes in order to obtain an appropriate solution. The present work introduces the extension of a system oriented to agents whose purpose is to reduce the waiting time of the drivers in an intersection of streets; modelling now vehicles (cars and taxis), pedestrians and an intelligent traffic light that evaluates the weights of the flows in two directions (X and Y), turn left and right. For which, the validation was carried out in six possible scenarios, defined by means of the variation of the flow of vehicles and pedestrians, of a turning frequency in 20 and 80 and the application of different adjustment factors for intelligent traffic lights. The simulation environment was implemented in NetLogo, which allowed comparing the impact of the use of the intelligent traffic light versus a fixed-time traffic light. Finally, we will present the conclusions and future work

Keywords: multi-agent systems; agent-oriented programming; traffic; NetLogo.

Resumen—Las pérdidas de tiempo por congestión y embotellamientos de tráfico son sucesos corrientes en las ciudades por una configuración no óptima de los semáforos. Diversas ciencias y especialidades han intentado comprender el fenómeno e identificar las causas en pos de obtener una solución apropiada. El presente trabajo introduce la extensión de un sistema orientado a agentes cuyo fin es reducir el tiempo de espera de los conductores en una intersección de calles; modelando ahora vehículos (autos y taxis), peatones y un semáforo inteligente que evalúa los pesos de los flujos en dos direcciones (X e Y), giro a la izquierda y derecha. Para lo cual, la validación fue llevada a cabo en seis escenarios posibles, definidos por medio de la variación del flujo de vehículos y peatones, de una frecuencia de giro en 20 y 80 y la aplicación de diferentes factores de ajustes para los semáforos inteligentes. El entorno de simulación se implementó en NetLogo, lo que permitió comparar el impacto del uso del semáforo inteligente versus un semáforo de tiempos fijos. Finalmente, presentaremos las conclusiones y el trabajo futuro.

Palabras clave: sistemas multi-agentes; programación orientada a agentes; tráfico; NetLogo.

I. INTRODUCCION

Las rutas y calles hacen posible la circulación y el recorrido de las ciudades, pero desafortunadamente el tráfico y particularmente la congestión del mismo conlleva a que los conductores sufran pérdidas de tiempo para llegar de un lugar a otro, por el tiempo que tardan en transitar sobre los caminos, sumado a los tiempos de espera por semáforos. Diversas disciplinas han estudiado dicha problemática en pos de comprender el fenómeno [1], [2], [3], y lograr una solución viable [4], [5], [6]. Algunos autores han encontrado que los conductores atascados en condiciones de alta congestión presentan niveles elevados de estrés [7], lo que puede llevarlos a modificar su comportamiento mientras manejan su vehículo. En consecuencia, es importante cambiar la forma en que las personas conducen, y encontrar mejores soluciones para lidiar con el tráfico, como semáforos inteligentes o incluso automóviles que se comunican entre sí.

El rápido crecimiento de las ciudades, la localización de las instituciones educativas y la diversidad de puestos de trabajo generan un aumento en el flujo diario del tráfico, lo que conlleva a un incremento en el número de vehículos y medios de transporte en las calles en particular y en las ciudades en general. Si bien diversos dominios han abordado el problema de la

congestión de tráfico [8-12] consideramos que la implementación de un enfoque simple, que implemente técnicas de modelado y simulación basados en agentes, con un nivel de dificultad bajo-medio de implementación es capaz de ser efectivo y viable para su aplicación en la vida real. Existe un gran número de herramientas y plataformas como NetLogo [13] y Repast, y un elevado número de aplicaciones, donde los usos más comunes tienen lugar en simulaciones sociales y problemas de optimización como el comportamiento humano, la simulación urbana, la gestión del tráfico, entre otros [14].

En [15], los autores presentan una simulación de tránsito en NetLogo, en una intersección en pos de su optimización mediante un enfoque basado en agentes, que disminuye el tiempo de espera de los vehículos, evitando que estos esperen un tiempo indeterminado o excesivamente largo; con la meta final de reducir la congestión de tráfico en una intersección de dos calles - doble mano y doble carril, donde circulan vehículos. En este contexto, el objetivo que persigue el presente artículo es introducir una extensión de [15], agregando a la simulación, semáforos de giro a la izquierda, taxis y peatones; y adaptando el semáforo inteligente al nuevo escenario.

La simulación posee sus raíces y motivaciones en una intersección real (precisamente aquella existente entre Av. Mariano Acosta y Ruta n° 8 en Pilar, Buenos Aires) en la cual, en el momento de ingresos o egresos a la Universidad se generan largas colas de autos, desde la entrada a la Facultad de Ingeniería hasta la Ruta n° 8, convirtiéndose en más de dos cuadras de autos.

En dicha intersección se modelan, por tanto, tres agentes: el agente vehículo (incluyendo autos regulares y taxis), el agente peatón y el agente intersección. Los vehículos pueden avanzar y en la intersección girar a la derecha o a la izquierda, para lo que habrá un semáforo de giro. Como parte de esta solución, el agente vehículo podrá avanzar, frenar, cambiar de carril y doblar a la derecha. La extensión de la funcionalidad del agente vehículo será el giro a la izquierda, los taxis y los choques. El giro a la izquierda lo podrá hacer únicamente estando en el carril izquierdo y cuando el semáforo de giro a la izquierda este en verde. Los taxis, serán distinguidos por su color amarillo. Funcionaran como autos normales con

la diferenciación de que si algún peatón está esperando un taxi, el mismo podrá detenerse y levantarlo. Además habrá choques entre vehículos, cuando dos vehículos estén demasiado cerca uno va a quedarse quieto (chocado) obstruyendo el tránsito por un tiempo definido por el usuario.

Luego, el agente peatón representa una persona que podrá avanzar por la vereda, frenar y doblar en las esquinas a la izquierda y a la derecha. Por otro lado, el agente intersección tendrá dos estrategias de funcionamiento: tiempos fijos o tiempos variables, siendo la primera por tiempos determinados previamente, donde el semáforo estará en verde o rojo por un tiempo predeterminado. Los tiempos variables son dinámicos, por lo que la estrategia cambiará a partir de los datos sensados por el agente intersección. Así, tendrá en cuenta la cantidad de vehículos que están esperando, el tiempo que estuvieron esperando y el tiempo desde la última vez que estuvo en verde cada dirección.

Teniendo como base lo anterior, se han planteado diversos escenarios donde varía el flujo de agentes y el factor para el semáforo inteligente. También, se variará la frecuencia de giro de los automóviles. El trabajo se estructura de la siguiente forma: en la sección 2, se contextualiza un breve estado de arte de sistemas basados en agentes aplicados al tráfico. La sección 3 explica el entorno de simulación e implementación del algoritmo, describiendo la herramienta elegida NetLogo, los agentes y sus funcionalidades, el ambiente y la interfaz de usuario, y la lógica (el código del algoritmo se presenta en un anexo aparte, subido en la dirección de la nota¹). La sección 4, busca mostrar y analizar el comportamiento emergente del sistema a partir de la comparación de las estrategias en los seis escenarios. Finalmente, la sección 5 presenta las conclusiones y los lineamientos del trabajo futuro.

II. ESTADO DE ARTE

Uno de los primeros sistemas de control de tráfico fueron aquellos con planes de tiempos de intersección fijos y señalizados, donde las longitudes del ciclo variaban entre los 35 y los 120 segundos [16]. Tiempo después surgieron, los semáforos con control dinámico que utilizan sensores para detectar distintos tipos de vehículos y usuarios no motorizados como es el caso de los

¹ <http://ccl.northwestern.edu/netlogo/upload.shtml>

peatones; y el control coordinado lo que posibilita que los conductores encuentren una onda verde: una cadena de luces verdes en los semáforos, fenómeno conocido como progresión [17]. Finalmente, el control de tráfico adaptativo responde a una estrategia en la que el tiempo de la señal del semáforo cambia o se adapta según la demanda del tráfico en tiempo real.

Existen varias simulaciones desarrolladas con la plataforma NetLogo de sistemas dinámicos y adaptativos de control de tráfico basados en agentes, para una o varias intersecciones. Para el caso de una intersección, es viable mencionar a [18] y [19]. En el primero de ellos, Bui et al. [18] reformulan los modelos de competencia económica de Cournot y Stackelberg para el desarrollo del algoritmo inteligente del semáforo, para vehículos con y sin prioridad. En este contexto, muestran una mejora en el tiempo de espera para dos condiciones, el flujo de vehículo uniforme y no uniforme en las dos direcciones, comparando el algoritmo propuesto versus uno basado en el método Round-Robin (en scheduling de procesos, las franjas horarias o quantum de tiempo se asignan a cada proceso en partes iguales y en orden circular, manejando todos los procesos sin prioridad), con semáforos de tiempos fijos y el otro usando un sensor de red.

En el segundo ejemplo, los autores [19] desarrollan en el contexto de Internet de las Cosas, el modelado de la comunicación entre los agentes, para mejorar el flujo de tráfico en una intersección en tiempo real. Los vehículos se pueden identificar en función de su ID (por ejemplo, matrícula del vehículo), a partir de la cual también puede comunicarse con la infraestructura. Cada vehículo está equipado con el sistema de posicionamiento para determinar su ubicación.

En el caso de los peatones, al usar dispositivos de conexión como teléfonos inteligentes, pueden comunicarse con el sistema. La simulación mide el rendimiento de los algoritmos de control del tráfico al aumentar la densidad de los agentes que llegan a la intersección, considerando una distribución aleatoria para cada dirección de la intersección.

Los autores [19] realizan la comparación de la eficiencia del algoritmo propuesto versus un algoritmo de programación de Round-Robin para controlar el semáforo. Los resultados muestran que el tiempo de espera del vehículo obviamente

aumenta cuando la densidad de los vehículos aumenta. Sin embargo, usando el algoritmo propuesto la variación no es tan significativa, dado que la prioridad de los mensajes de solicitud del vehículo está dada por un modelo FIFO (First Come, First Out), lo que permite que el vehículo pase la intersección lo antes posible. Concluyen que el algoritmo propuesto reduce el tiempo de espera de los agentes.

Para el caso de múltiples intersecciones, en el contexto de complejas redes de tránsito desarrollaron en [20] un enfoque multi-agente utilizando clustering difusos para administrar el flujo del tráfico en condiciones determinadas, con el empleo de NetLogo para testear y evaluar el enfoque planteado. Los autores concluyen que para una grilla de intersecciones, el tiempo medio de retardo se puede reducir en un 42,76%, en comparación con la señal de tráfico de secuencia fija convencional.

Otros autores presentan en [21] un modelo adaptativo que modela una ciudad de 19x19 calles, comparando dos enfoques diferentes: a) en cada intersección se cuenta el número de automóviles que se aproximan a la luz roja, independientemente de su estado o velocidad; b) la duración del próximo ciclo verde de cada luz es directamente proporcional a la cantidad de automóviles que cruzaron la intersección en el último ciclo verde. Dichos autores encuentran mejoras del segundo enfoque con respecto al primero, para la velocidad promedio de los autos, el número promedio de los autos parados y el tiempo medio de espera de los autos, para bajas densidades de autos y una mejora menor para altas densidades de autos. Concluyen que el segundo es más fácil de implementar debido a que es posible usar bandas de goma o detectores de bucle enterrados para la detección de autos.

Por último, otro artículo [22] presenta una simulación de la coordinación dinámica del sistema de semáforos urbanos, utilizando los datos de tráfico disponibles localmente y el estado del tráfico de las intersecciones vecinas, con el fin de mejorar el tránsito urbano en la ciudad de Colima, México. Muestra la simulación de tres intersecciones (500 metros de distancia entre cada intersección) en una avenida. Se define y utiliza un algoritmo inteligente para el control adaptativo con capacidades de aprendizaje que permite un tráfico más fluido, reduciendo el tiempo de espera

promedio y el tamaño medio de cola de espera por intersección. Los autores comparan la situación real con la propuesta, en escenarios de baja y alta afluencia de autos. Los mismos concluyen que el tiempo de espera para la dirección norte-sur se reduce un 16%, para el sur-norte el 59%, para el este-oeste el 57% y para el oeste-este hasta el 33% con respecto a la situación actual, para tráfico normal y un 29% para horas picos.

Consecuentemente, los ejemplos citados demuestran que es posible reducir el tiempo de espera en una o varias intersecciones, aspecto que motiva presentar nuestros resultados, considerando que aún quedan tantas intersecciones o arterias en las ciudades en general y particularmente en Buenos Aires, donde es posible agilizar el tráfico.

III. ENTORNO DE MODELADO, SIMULACIÓN E IMPLEMENTACIÓN DEL SISTEMA

La simulación y modelado computacional permite abordar el análisis del comportamiento de los sistemas, brindando estrategias para su funcionamiento. En este contexto, un modelo es regularmente considerado como una representación abstracta y simplificada de la realidad; mientras que una simulación es una manifestación del modelo, encapsulando escenarios reales [23].

Ahora bien, un agente representa un nivel alto de abstracción que define una unidad compleja de software de una forma eficiente y conveniente. Puede definirse un agente de software como un sistema de computadora que es capaz de realizar acciones autónomas dentro de un ambiente, exhibiendo un comportamiento flexible: es decir, abordando la reactividad, proactividad, y la habilidad social [24].

El paradigma basado en agentes puede ser considerado como una extensión intrínseca de la orientación a objetos, donde un agente representa un objeto que tiene control sobre su ejecución [25]. Existen diversas diferencias de los agentes con los objetos, como que los agentes son autónomos, que se comunican con otros agentes mediante mensajes, que no invocan métodos (como los objetos), que un sistema multi-agente es multi-threaded (a diferencia de un sistema orientado a objetos que tiene un solo thread de ejecución), entre otros.

Conceptualmente, el paradigma de agentes encaja perfectamente en la necesidad de modelar sistemas complejos. El Modelado y Simulación basado en Agentes (en inglés, *Agent-based Modelling and Simulation* – ABMS) hace referencia a modelos computacionales que involucran acciones dinámicas y reacciones entre agentes que interactúan en un ambiente, con el fin de evaluar su diseño y performance; y así derivar en *insights* sobre su comportamiento [23].

En el modelado basado en agentes, un sistema es modelado como una colección de agentes, donde cada agente evalúa individualmente su situación y toma decisiones sobre la base de un conjunto de reglas. Los agentes pueden ejecutar diversos comportamientos apropiados para el sistema que representan, por ejemplo, producir, consumir o vender. En el nivel más simple, un modelo basado en agentes consiste en un sistema de agentes y las relaciones entre ellos. Pero un modelo simple basado en agentes puede mostrar patrones de comportamiento complejos [26] y proporcionar información valiosa sobre la dinámica del sistema del mundo real que emula.

Además, los agentes pueden ser capaces de evolucionar, permitiendo que surjan comportamientos imprevistos. El modelado basado en agentes sofisticado puede incorporar redes neuronales, algoritmos evolutivos u otras técnicas de aprendizaje para permitir el aprendizaje y la adaptación realistas.

Consecuentemente, es viable afirmar que el modelado basado en agentes es una técnica de simulación poderosa, que abordan grandes áreas de aplicación y para la cual se han desarrollado herramientas software para lograr el modelado de los sistemas.

A. Modelado basado en Agentes mediante NetLogo

El modelado basado en agentes cuenta con un gran número de plataformas, que pueden dividirse en dos categorías. La primera define los modelos utilizando lenguajes de programación como Java o C++, donde las más conocidas son Swam y Repast; mientras que la segunda categoría utiliza lenguajes dedicados y las plataformas de código abierto más conocidas son NetLogo y GAMA [27].

NetLogo [13] creado en 1999 por U. Wilensky, es un entorno de modelado programable que permite simular fenómenos naturales y sociales, y es viable dar instrucciones a cientos o miles de

agentes que operan independientemente lo que posibilita la exploración de la relación el comportamiento a nivel micro (individuos e interacciones entre ellos) y los patrones a nivel macro (representado en los patrones de organización o comportamiento emergentes de los individuos como un todo).

Es lo suficientemente simple para los estudiantes y profesores, pero lo suficientemente avanzado como para servir como una poderosa herramienta para los investigadores en muchos campos. En NetLogo todo es un agente, desde cada cuadrado que conforma el plano donde se visualiza la simulación, hasta quien se encarga de pintar las calles, los semáforos y los autos. Para crear un nuevo tipo de agente, es necesario utilizar la palabra reservada <breed>, para agregar propiedades a cada agente debe emplearse la palabra reservada <own>.

Con el fin de poder operar sobre cierto tipo de agente debe usarse la palabra reservada <ask>. Además, ofrece una función <go> que ejecuta todo lo que esté dentro de ella a cada tick, considerando que un tick es una medida de tiempo que puede ser modificada. Se ejecuta en la JVM, por lo que funciona en todas las plataformas principales como Mac, Windows y Linux. Se ejecuta como una aplicación de escritorio y la operación de línea de comando también es compatible [28].

B. Descripción de los Agentes Modelados

Los agentes implementados son los que se observan en las Tablas I, II y III, donde se describen los agentes por sus atributos, por la información que sensan del ambiente y por su comportamiento.

TABLA I
DESCRIPCIÓN DEL AGENTE AUTO

Descripción	Agente Auto
Atributos	Velocidad: velocidad actual del auto, Paciencia: el nivel de tolerancia restante que tiene el conductor de tanto esperar, Dirección: hacia dónde va el auto, Carril actual, Coordenadas actuales, Girar a la derecha(Sí o no): indica si el auto va a girar en la intersección, Girar a la izquierda(Sí o no): indica si el auto va a girar en la inter-sección, Tiempo de espera en el semáforo, Aceleración, Desaceleración, Taxi (Si o no): Indica si es un taxi, Está libre (Aplica sólo para taxis): Indica si tiene pasajeros,

	Chocado.
Información sensada del ambiente	Autos cercanos: utilizando un ángulo de visión de 15 grados y un radio que varía dependiendo de la velocidad del auto, Velocidad máxima de la calle, Color de la luz del semáforo, Autos en carril adyacente, Posición dentro del ambiente, Peatones esperando taxi
Comportamiento	Acelerar: lo hace si no tiene ningún auto en su espectro de visión yendo a una velocidad menor a la suya, si no supera la velocidad y si no se encuentra en el límite previo al semáforo, mientras este está en rojo, Frenar: lo hace si tiene algún auto en su espectro de visión yendo a una velocidad menor a la suya y si se encuentra en el límite previo al semáforo, mientras este está en rojo, Doblar a la derecha: lo hace sí se encuentra en el carril derecho al momento de llegar al área crítica de la intersección con una probabilidad definida por una variable y el semáforo está en verde, Doblar a la izquierda: lo hace sí se encuentra en el carril izquierdo al momento de llegar al área crítica de la intersección con una probabilidad definida por una variable y el semáforo de giro está en verde, Cambiar de carril: lo hace sí la paciencia del conductor llega a 0 (está va disminuyendo a medida que el tiempo de espera aumenta) y si no hay ningún auto en su posible futura posición, Levantar un pasajero: Lo hace si es un taxi y está libre y pasa por al lado de un peatón que está esperando un taxi, Chocar: Al superponerse con otro vehículo se considera chocado y se queda quieto en el lugar hasta ser removido después de un límite de tiempo

TABLA II
DESCRIPCIÓN DEL AGENTE INTERSECCIÓN

Descripción	Agente Intersección
Atributos	Luces de los semáforos: estado actual de los semáforos, Inteligencia (Sí o no): define el algoritmo a utilizar para el manejo de luces de los semáforos, Tiempo mínimo de verde, Tiempo desde que se realizó el último cambio de luz, Dirección, Es de giro, Es de auto: Si aplica los autos o a los peatones.
Información sensada del ambiente	Cantidad de autos detrás de los límites de la intersección en todas las direcciones, Tiempos de espera de los autos
Comportamiento	Cambiar luz de semáforos: Sin inteligencia: lo realiza por tiempos definidos, el semáforo está en verde por un tiempo determinado y después cambia a rojo, Con inteligencia: lo realiza a partir de los datos censados del ambiente. A partir de un algoritmo que cuenta cuántos autos están esperando del otro lado y de cuánto tiempo llevan esperando decide si cambiar o seguir en el estado actual.

TABLA III
DESCRIPCIÓN DEL AGENTE PEATÓN

Descripción	Agente Peatón
Atributos	Velocidad: velocidad actual del peatón, Dirección: hacia dónde va el peatón, Coordenadas actuales, Girar a la derecha(Sí o no), Girar a la izquierda(Sí o no), Girar antes: Indica si va a girar antes de cruzar, Girar después: Indica si va a girar después de cruzar, Tiempo de espera en el semáforo,
Información sensada del ambiente	Peatones cercanos, Color de la luz del semáforo, Posición dentro del ambiente, Taxi esperando a que el peatón suba
Comportamiento	Acelerar: lo hace si no tiene ningún peatón exactamente delante suyo, Frenar: lo hace si tiene algún peatón delante suyo yendo a una velocidad menor a la suya y si se encuentra en el límite previo al semáforo, mientras este está en rojo, Doblar a la derecha: lo hace sí se encuentra en una esquina con la probabilidad definida para doblar, Esperar un taxi: Lo hace si cumple con la probabilidad definida para esperar el taxi y está en una zona en la que se puede parar uno, Subir a un taxi: Lo hace si está esperando un taxi y uno se detiene para levantarlo.

C. Algoritmo del semáforo inteligente

El algoritmo del semáforo inteligente desarrollado coordina dos semáforos (X e Y) uno para cada dirección de la intersección de las calles, con sus respectivos semáforos de giro a la izquierda y dos de los peatones. Además, considera 4 semáforos de peatones, que al mismo tiempo están todos en verde o rojo, por lo que el algoritmo lo considera como una unidad. Se distinguen 5 estados diferentes, descritos en la Tabla IV.

TABLA IV
ESTADOS DE LOS SEMÁFOROS

Semáforos	E 1	E 2	E3	E4	E5
Eje X	v	r	r	r	r
Semáforo giro X	r	v	r	r	r
Eje Y	r	r	v	r	r
Semáforo giro Y	r	r	r	v	r
Peaton	r	r	r	r	v

También considera la cantidad de vehículos (autos y taxis) que están esperando en ambas direcciones o ejes, siendo el eje la dirección del flujo de vehículos que tiene el semáforo en luz verde y la dirección del flujo de autos que tienen

luz roja. Evalúa los pesos de los cuatro semáforos y los peatones.

Los agentes esperando en el eje son todos aquellos que se encuentran antes de cruzar la intersección para cada uno de los semáforos previamente descritos. El tiempo de semáforo en rojo es el tiempo desde que el semáforo se puso en rojo hasta el momento actual. Y, al valor del tiempo, se lo multiplica por un factor de ajuste, a este valor se lo puede hacer variar para conseguir mejores resultados dependiendo del flujo de los autos que hay en el momento. Para el eje que está en verde, el tiempo rojo será 0 por lo que sólo contarán los autos que quieren pasar por la intersección. Disminuyendo o aumentando el valor del factor, logramos variar la incidencia del tiempo en el cálculo del peso.

El algoritmo hará que la luz del semáforo con mayor peso se ponga en verde, salvo que aún no se haya completado el mínimo tiempo de verde fijado a través de los controles de usuario. Vale destacar que la luz verde corresponde a la dirección con mayor peso y los pesos i , se calculan según (1):

$$P_i = A_i + T_{sri} * F_a \tag{1}$$

donde:

- i , son el peso para la dirección X, Y, Giro izquierda eje X, Giro izquierda eje Y y Peatones
- A_i , los agentes (autos o peatones) llegando a la intersección en el semáforo (frenados y en dirección a)
- T_{sri} , el tiempo de semáforo en rojo
- F_a , es el factor de ajuste

Vale destacar que T_{sri} es multiplicado por un factor de ajuste cuyo valor fue obtenido por medio de pruebas experimentales, por lo que para el eje que está en verde, T_{sri} va a ser igual a 0. Por ejemplo, para un tiempo mínimo de luz verde de 200 ticks (unidad de tiempo en NetLogo) y un F_a de 0.001, el semáforo correspondiente al eje X (calle horizontal) acaba de cambiar su luz a verde, y el eje y (calle vertical) tiene el semáforo en rojo, consecuentemente la luz del semáforo permanecerá en verde el tiempo definido en un principio (200 ticks). Pasado el tiempo mínimo comienza a evaluarse la fórmula del peso en ambos ejes:

$$P_x = A_{ux} + T_{srx} * 0.001 \quad (2)$$

$$P_y = A_{uy} + T_{sry} * 0.001 \quad (3)$$

Luego, el peso para los ejes X e Y con giro y para los peatones son los siguientes:

$$P_{xg} = A_{uxg} + T_{srxcg} * 0.001 \quad (4)$$

$$P_{yg} = A_{uyg} + T_{sryg} * 0.001 \quad (5)$$

$$P_p = P + T_{srp} * 0.001 \quad (6)$$

siendo:

- P_{xg} , peso para el eje X con giro a la derecha
- P_{yg} , peso para el eje Y con giro a la derecha
- A_{uxg} , autos previos a la intersección en el eje X que van a girar a la izquierda
- A_{uyg} , autos previos a la intersección en el eje Y que van a girar a la izquierda
- P_p , peso para peatones
- P , peatones previos a la intersección

Para poner valores numéricos a un ejemplo particular, diremos que al pasar 1000 ticks del cambio de luz (el último cambio había sido del eje Y al eje X) en el eje X hay 10 autos previos a la intersección, en el eje Y hay 5 autos previos a la intersección, en el giro del eje X hay 2 y no está en verde hace 1200 ticks, en el giro del eje Y no hay nadie y está en rojo hace 1600 ticks y en el semáforo de los peatones hay 8 y está en rojo hace 1400 ticks. Por lo tanto, hemos realizado una breve comparación de los pesos:

$$P_x = 10$$

$$P_y = 5 + 1000 * 0.001 = 6$$

$$P_{xg} = 2 + 1200 * 0.001 = 3.2$$

$$P_{yg} = 0 + 1600 * 0.001 = 1.6$$

$$P_p = 8 + 1400 * 0.001 = 9.4$$

Consecuentemente, el P_x continua siendo el más fuerte. Este resultado nos indica que el P_x es el mayor por lo tanto el semáforo debe continuar en verde para el eje X. Al pasar unos 100 ticks más, el agente intersección observa que en el eje X hay 7 autos previos a la intersección y en el eje Y hay 6 autos previos a la intersección, el resto se mantiene igual. Por lo tanto realiza la comparación de los pesos:

$$P_x = 7$$

$$P_y = 6 + 1100 * 0.001 = 7.1$$

$$P_{xg} = 2 + 1300 * 0.001 = 3.3$$

$$P_{yg} = 0 + 1700 * 0.001 = 1.7$$

$$P_p = 8 + 1500 * 0.001 = 9.5$$

Por lo tanto, P_p paso a ser el más fuerte. Este resultado indica que el P_p es el mayor por lo tanto el semáforo de los peatones pasará a verde mientras que el del eje X a rojo. Los semáforos permanecerán así al menos por el tiempo mínimo de verde (200 ticks) y luego, volverá a hacer los cálculos y la comparación.

Finalmente, si comparamos este algoritmo con el de semáforos de tiempos fijos destacamos que en este último no se realiza una evaluación de los pesos, sino que el cambio de luz se origina por el transcurso de los ticks pre-definidos.

D. Ambiente e interfaz de usuario

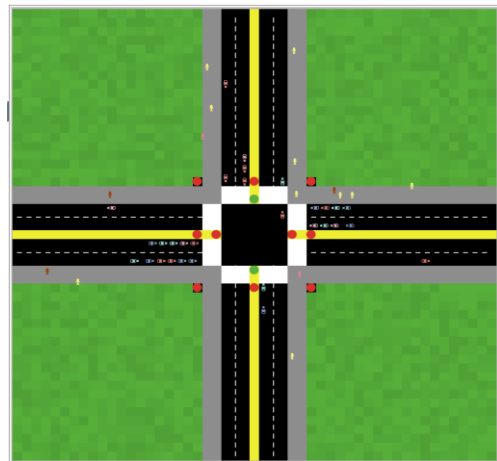


Fig. 2. Interfaz gráfica del modelo en NetLogo.

Se han considerado los siguientes supuestos, que están aplicados y pueden visualizarse en la Fig. 2:

(a) los únicos tipos de vehículos modelados son el auto y el taxi,

(b) las calles que conforman la intersección poseen una velocidad máxima que los autos respetan,

(c) las calles que conforman la intersección son doble mano,

(d) cuando un auto llega a la intersección puede cruzarla, girar a la derecha o a la izquierda,

(e) para doblar a la derecha los autos deben estar en el carril derecho de la calle,

(f) para doblar a la izquierda los autos deben estar en el carril izquierdo de la calle,

(g) los sensores de la intersección no fallan en ningún momento,

(h) no hay disturbios externos al sistema,

(i) el peatón cumple con las reglas de tránsito y cruza por donde es debido,

(j) cuando dos autos se acercan demasiado es considerado como un choque y el auto queda inmóvil en el lugar por un tiempo definido y luego es removido.

Recordando la definición de Wooldridge [24], un agente se encuentra en un ambiente e interactúa con él y con otros agentes, con los cuales se comunica, coopera y negocia. En este caso y debido a su modelización en NetLogo, está constituido por agentes pasivos o patches que se encargan inicialmente de decorar el entorno. También, está compuesto por dos ejes: el horizontal y el vertical, donde cada uno contiene una calle compuesta por dos manos y dos carriles en cada dirección (un total de 4 carriles). Las líneas amarillas que se visualizan en la Figura 2 son las encargadas de dividir las manos de una misma calle mientras que las blancas punteadas dividen los carriles. El verde alrededor de la calle es meramente decorativo y no posee ninguna interacción con los agentes.

El área crítica de la intersección es delimitada por las líneas blancas gruesas del centro de la imagen. Estas líneas indican hasta dónde pueden avanzar los autos si el semáforo de su eje se encuentra en rojo. Los semáforos son los puntos circulares rojos o verdes que se encuentran sobre la línea amarilla en el eje X, sirven para manejar el eje X y los que se encuentran sobre el eje Y para manejar el eje Y. Los semáforos internos son los que indican si los autos pueden seguir derecho o doblar a la derecha y los externos los que indican si se puede girar a la izquierda. Los semáforos que se encuentran al lado de la vereda son para los peatones.

La interfaz gráfica del modelo incluye la interacción con diversas variables del sistema, permitiendo la realización del setup inicial. Los botones básicos son <setup>, <go>, <go once>. Se implementó un <select car> que permite ver el estado individual de un auto al hacerle click. Luego, el switch de inteligencia cambia el tipo de semáforo (tiempos fijos o inteligente). También, es posible cambiar el tiempo mínimo de las luces

de los semáforos y la frecuencia, de giro, de aceleración, de velocidad máxima, de desaceleración, y de paciencia máxima.

Al estar corriendo la simulación en NetLogo, es posible interactuar con estos controles en la interfaz para probar la variabilidad de los resultados en diversos escenarios.

IV. PRESENTACIÓN DE RESULTADOS.

Con el fin explícito de determinar la viabilidad del algoritmo desarrollado y evidenciar el comportamiento emergente, se han planteado los seis (6) escenarios siguientes, en los que se variará la frecuencia de agentes y el factor de ajuste para el semáforo inteligente. Se seleccionó un factor de ajuste 0.001 y 0.01. Aumentando el valor del factor para los semáforos de giro logramos aumentar el peso para estos semáforos, modificando su prioridad, dado que la selección del próximo semáforo a cambiar a verde, va a ser el semáforo con mayor peso.

Además, cada escenario, será evaluado con una frecuencia de giro de 20 y 80.

- **Escenario 1:** Misma frecuencia de agentes en todas las direcciones, mismo factor de semáforo inteligente (0.001) en todos los semáforos.
- **Escenario 2:** Misma frecuencia de agentes en todas las direcciones. Factor 0.001 de semáforo inteligente en los semáforos del eje X, eje Y, y peatones. Factor de 0.01 de semáforo inteligente en los semáforos de giro.
- **Escenario 3:** Frecuencia de generación de agentes de 10 en el eje Y, y frecuencia de generación de agentes de 3 en el eje X y peatones. Mismo factor de semáforo inteligente (0.001) en todos los semáforos.
- **Escenario 4:** Frecuencia de generación de agentes de 10 en el eje Y, y frecuencia de generación de agentes de 3 en el eje X y peatones. Factor 0.001 de semáforo inteligente en los semáforos del eje X, eje Y, y peatones. Factor de 0.01 de semáforo inteligente en los semáforos de giro.
- **Escenario 5:** Frecuencia de generación de agentes de 15 en el eje Y, y frecuencia de generación de agentes de 1 en el eje X y

peatones. Mismo factor de semáforo inteligente (0.001) en todos los semáforos.

- **Escenario 6:** Frecuencia de generación de agentes de 15 en el eje Y, y frecuencia de generación de agentes de 1 en el eje X y peatones. Factor 0.001 de semáforo inteligente en los semáforos del eje X, eje Y, y peatones. Factor de 0.01 de semáforo inteligente en los semáforos de giro.

Los valores pre-establecidos para todos los escenarios son los siguientes:

1. set-ticks-to-remove-collisions: 320
2. green-length: 194
3. turn-freq: 38
4. acceleration: 0.008
5. max-speed: 0.5
6. deceleration: 0.06
7. max-patience: 50
8. pedestrian-turn-freq: 86
9. taxi-freq: 78
10. waiting-for-taxi-freq: 29
11. collisions-enabled: on

Ahora bien, los valores variables para los escenarios se ven reflejados en la Tabla V.

A. Resultados obtenidos

En cada uno de los escenarios se efectivizará la comparación de la utilización de un semáforo fijo o inteligente, con o sin giro; vislumbrando además lo que ocurre con una frecuencia de giro de 20 y una de 80. Las tablas a continuación presentan un resumen claro de los resultados y caracterización de los escenarios.

TABLA V
VALORES FIJOS DE LOS ESCENARIOS

Variables	E1	E2	E3	E4	E5	E6
freq-east	10	10	3	3	1	1
freq-north	10	10	10	10	15	15
freq-south	10	10	10	10	15	15
freq-west	10	10	3	3	1	1
pedestrian-freq	10	10	3	3	1	1
x-factor	0.001	0.001	0.001	0.001	0.001	0.001
y-factor	0.001	0.001	0.001	0.001	0.001	0.001
x-turn-factor	0.001	0.01	0.001	0.01	0.001	0.01
y-turn-factor	0.001	0.01	0.001	0.01	0.001	0.01
pedestrian-factor	0.001	0.001	0.001	0.001	0.001	0.001

En las Tablas VI y VII se puede observar el tiempo promedio de espera en ticks de los vehículos.

TABLA VI
TIEMPO PROMEDIO DE ESPERA, CON 20% GIRO IZQUIERDA

Tiempo promedio de espera (ticks)	Inteligente	Tiempo fijo	
factor de 0.001 en todas las direcciones	E1	413	1332
	E3	393	1258
	E5	416	1353
Factor de 0.01 en los semáforos de giro y 0.001 en el resto	E2	647	1332
	E4	692	1258
	E6	532	1353

TABLA VII
TIEMPO PROMEDIO DE ESPERA, CON 80% GIRO IZQUIERDA

Tiempo promedio de espera (ticks)	Inteligente	Tiempo fijo	
factor de 0.001 en todas las direcciones	E1	618	1060
	E3	561	913
	E5	442	1213
Factor de 0.01 en los semáforos de giro y 0.001 en el resto	E2	917	1060
	E4	384	913
	E6	402	1213

En las Tablas VIII y IX, se describe el tiempo máximo de espera en ticks. Es el tiempo más largo que espero un vehículo para poder cruzar. Sirve para controlar que ningún vehículo haya estado esperando un tiempo excesivo en el cruce.

TABLA VIII
TIEMPO MÁXIMO DE ESPERA, CON 20 GIRO IZQUIERDA

Tiempo promedio de espera (ticks)	Inteligente	Tiempo fijo	
Factor de 0.001 en todas las direcciones	E1	>10000	5202
	E3	>10000	4397
	E5	>10000	4978
Factor de 0.01 en los semáforos de giro y 0.001 en el resto	E2	6210	5202
	E4	5481	4397
	E6	8281	4978

TABLA IX
TIEMPO MÁXIMO DE ESPERA, CON 80 GIRO IZQUIERDA

Tiempo promedio de espera (ticks)	Inteligente	Tiempo fijo	
Factor de 0.001 en todas las direcciones	E1	6050	3538
	E3	4256	3737
	E5	7160	3302
Factor de 0.01 en los semáforos de giro y 0.001 en el resto	E2	4472	3538
	E4	2505	3737
	E6	7096	3302

Las celdas sombreadas en las Tablas VII y IX indican el mejor escenario en el que puede ser utilizado el semáforo inteligente, debido a que reduce considerablemente su tiempo promedio de espera y el tiempo máximo de espera.

En el caso de los choques y los taxis, estos influyen en menor medida en los resultados presentados, debido a que obstaculizan el tránsito parcialmente, dado que no impiden totalmente la circulación de autos.

Los peatones tienen una incidencia mayor porque al no cruzar en paralelo con los autos esto hace que los autos tengan menos tiempo su semáforo en verde, aumentando su tiempo de espera en el cruce.

B. Descripción de los escenarios

En el caso del escenario 1, el tiempo promedio de espera se reduce fuertemente con el semáforo inteligente. El problema es que aumenta el valor máximo de espera. En el caso del giro con frecuencia 20, el tiempo de espera máximo no llegó a encontrarse en los 10000 ticks. Esto ocurre porque al tener un factor de giro tan bajo y tan pocos autos puedan doblar, el peso de ese semáforo nunca llega a ser más alto que otros. Con la frecuencia de 80 de giro, los autos doblan con más rapidez, pero esperan más tiempo que los autos controlados por un semáforo de tiempo fijo.

En el escenario 2, con otro factor de semáforo, los valores cambian ya que aunque el valor de tiempo de espera no se reduce tanto como con el escenario anterior, el valor máximo de espera toma un valor acotado y menor al valor del escenario 1.

Para el escenario 3, el valor promedio de espera se reduce considerablemente con el sistema inteligente, en más de un 68% con respecto al semáforo inteligente. El tiempo de espera máximo aumenta, debido a que hay pocos autos que giren, y considerando que el factor de giro es igual al resto esto tarda mucho en ocurrir; en 10000 ticks no ocurrió nunca. Una mayor frecuencia de giro permite un mejor flujo de giro, y aunque el promedio de espera no se reduce tanto como con 20, el valor máximo de espera es similar al que se tiene con el semáforo no inteligente.

Ahora bien, en el escenario 4, los semáforos inteligentes también mejoran el promedio de espera con respecto al semáforo inteligente para ambas frecuencias de giro. Se obtiene el menor

tiempo de espera promedio de todas las simulaciones para el caso de una frecuencia de 80 de giro; se reduce aproximadamente un 58% reduciendo también el tiempo de espera máximo en un 33%, con respecto al semáforo fijo. Con el 20 de giro el tiempo promedio de espera se reduce menos y aumenta demasiado el tiempo máximo de espera.

Luego, en el escenario 5, ocurre algo similar al escenario 3, los tiempos de espera promedio presentan una mejora aún mayor. Al mismo tiempo, se puede ver que en situaciones con flujos muy distintos el factor 0.001 puede dejar esperando mucho tiempo a un auto.

Finalmente, en el escenario 6, para una brecha más grande de flujo de autos (1-15), el factor 0.01 tampoco logra hacer mucha diferencia con el otro factor 0.001, pero logra reducir el valor máximo de espera cuando el flujo de giro es bajo. De todos modos, la mejora de promedio de espera obtenida entre el semáforo inteligente y el de tiempo fijo en todos los escenarios es destacable (media 42%, variando [13%-68%]).

C. Análisis de los escenarios

El semáforo inteligente en todos los escenarios mejora el promedio de espera de los vehículos (autos y taxis) en el semáforo. Se obtuvo un mejor resultado con una mayor frecuencia de giro (80 con respecto a 20), evitando que los vehículos queden esperando un tiempo excesivo. El factor de 0.001 introducido en el artículo previo de los autores [15] no es recomendable para este ambiente ya que si no hay muchos autos que doblen, estos autos tendrán una espera prolongada. Disminuido el factor a 0.01 en los semáforos de giro y manteniendo 0.001 en el resto, mejora el resultado. El sistema se optimiza cuando hay más autos que quieren doblar a la izquierda y ejercen más peso para hacerlo.

Al definir en qué tipo de flujos de vehículos funciona mejor habría que analizar cómo influye el tiempo promedio de espera. En el caso de un flujo similar en todas las direcciones, el tiempo de espera promedio es ligeramente menor pero el costo del tiempo de espera extra es mínimo así que, en una intersección con este tipo de flujo sería conveniente si hay muchos autos que giran. En el caso de que el flujo sea superior en una intersección pero no excesivamente superior, es conveniente sin importar cuantos autos doblen a la izquierda porque la mejora promedio es muy

alta (para un giro de 20 baja de 1258 a 692 y para un giro de 80 baja de 913 a 384).

En el escenario en el que los flujos son muy dispares no es conveniente ya que aunque el tiempo promedio baja casi tres veces en ambos casos, el tiempo máximo de espera sube demasiado (para un giro de 20 de 4978 a 8281 y para un giro de 80 de 3302 a 7096), por lo que no es conveniente.

Concluimos que el algoritmo inteligente logra reducir el tiempo de espera promedio y máximo para una diferencia de frecuencia media entre las dos direcciones y un factor 0.01. Si la frecuencia de agentes en las dos direcciones es similar o mayor, se disminuye el tiempo de espera medio con respecto al semáforo de tiempos fijos, pero el tiempo de espera máximo va a ser un poco mayor para frecuencias de agentes similares y un poco más del doble para frecuencias muy disímiles entre los ejes X e Y.

V. CONCLUSION

El tráfico es una realidad que afecta a muchas personas, y existen evidencias de la disminución del tiempo de espera de los conductores si se implementan sistemas dinámicos y adaptativos de control de semáforos. El enfoque orientado a agentes tiene características esenciales que facilitan su modelización y simulación. Existen diversas aplicaciones, sistemas y simulaciones que posibilitan que el área se mantenga activa, debido a que el incremento en la demanda de la sociedad en pos de una mejora no solo es en la infraestructura sino también en los sistemas de gestión.

El sistema de control de semáforos en una intersección compleja descripto, logra optimizar el tiempo promedio de espera de los vehículos en una intersección de calles -con doble carril- aplicando un algoritmo que evalúa los pesos para las direcciones de tráfico (X e Y), los giros (derecha e izquierda), y los peatones.

La reducción del tiempo de espera en el semáforo se logra con un factor de ajuste del 0.01 para los giros a derecha e izquierda y un factor de ajuste de 0.001 para el resto de las direcciones y una frecuencia de giro de 80. El escenario con mejor optimización se obtiene con flujos de vehículos en dirección X e Y, medianamente diferente.

El análisis de los diferentes escenarios posibilita comprender cómo afecta la variación de flujo de vehículos (autos y taxis) con la circulación de peatones, y la variación de frecuencia de giro. El entorno orientado a agentes brinda un contexto adecuado para realizar este análisis. Permite la extensión de los agentes como ser bicicletas, transporte público, ambulancias, motos, entre otros.

Destacamos que la simulación realizada pone en evidencia que es factible realizar una reducción sensible en los tiempos de espera de un semáforo, en una intersección de calles complejas, aplicando un algoritmo simple, aspecto que tiene un impacto positivo en la vida de una ciudad. A futuro se prevé realizar el análisis de una intersección compleja real, incorporando al modelo frecuencias reales de flujo de vehículos y peatones en una intersección existente. También será posible extender el modelo a una red de tráfico más amplia.

AGRADECIMIENTOS

El presente proyecto se ha realizado con el apoyo de la Facultad de Ingeniería, Universidad Austral.

REFERENCIAS

- [1] J. Long, Z. Gao, H. Ren, and A. Lian. Urban traffic congestion propagation and bottleneck identification. *Science in China Series F: Information Sciences*, 51(7), 948-964, 2008.
- [2] A. M. Rao, and K. R. Rao. Measuring Urban Traffic Congestion-A Review. *International Journal for Traffic & Transport Engineering*, 2(4), 2012.
- [3] F. Tan, J. Wu, Y. Xia, and K. T. Chi. Traffic congestion in interconnected complex networks. *Physical Review E*, 89(6), 062813, 2014.
- [4] J. A. Lindley. Urban freeway congestion: quantification of the problem and effectiveness of potential solutions. *ITE journal*, 57(1), 27-32, 1987.
- [5] R. Bauza, and J. Gozávez, J. Traffic congestion detection in large-scale scenarios using vehicle-to-vehicle communications. *Journal of Network and Computer Applications*, 36(5), 1295-1307, 2013.
- [6] P. Lopez-Garcia, E. Onieva, E. Osaba, A. D. Masegosa, and A. Perallos. A hybrid method for short-term traffic congestion forecasting using genetic algorithms and cross entropy. *IEEE Transactions on Intelligent Transportation Systems*, 17(2), 557-569, 2016.
- [7] D. A. Hennessy, and D. L. Wiesenthal. Traffic congestion, driver stress, and driver aggression. *Aggressive behavior*, 25(6), 409-423, 1999.
- [8] R. Putha, L. Quadrioglio, and E. Zechman. Comparing ant colony optimization and genetic algorithm approaches for solving traffic signal coordination under oversaturation conditions. *Computer - Aided Civil and Infrastructure Engineering*, 27(1), 14-28, 2012.
- [9] D. Teodorović, and M. Dell'Orco. Mitigating traffic congestion: solving the ride-matching problem by bee colony optimization. *Transportation Planning and Technology*, 31(2), 135-152, 2008.
- [10] A. L. Bazzan, and F. Klügl. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review*, 29(3), 375-403, 2014.
- [11] B. Chen, and H. H. Cheng. A review of the applications of agent technology in traffic and transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 11(2), 485-497, 2010.

- [12] P. A. Ehlert, and L. J. Rothkrantz. Microscopic traffic simulation with reactive driving agents. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE* (pp. 860-865). IEEE, 2001.
- [13] S. Tissue, and U. Wilensky. NetLogo: Design and implementation of a multi-agent modeling environment. In *Proceedings of agent*, vol. 2004, pp. 7-9.
- [14] R. J. Allan. Survey of agent based modelling and simulation tools (pp. 1362-0207). Science & Technology Facilities Council, 2010.
- [15] T. F. Battolla, S. Fuentes, J. I. Illi, J. Nacht, M. Falco, G. Pezzuchi, and G. Robiolo. Sistema dinámico y adaptativo para el control del tráfico de una intersección de calles: modelación y simulación de un sistema multi-agente. En: *Simposio Argentino de Inteligencia Artificial (ASAI) – Jornadas Argentinas de Informática, Universidad de Palermo, Septiembre de 2018*.
- [16] S. F. Smith, G. J. Barlow, X. F. Xie, and Z. B. Rubinstein. Smart Urban Signal Networks: Initial Application of the SURTRAC Adaptive Traffic Signal Control System. In *ICAPS 2013*.
- [17] T. Nagatani. Vehicular traffic through a sequence of green-wave lights. *Physica A: Statistical Mechanics and its Applications*, 380, 503-511, 2007.
- [18] K. H. N. Bui, J. E. Jung, and D. Camacho. Game theoretic approach on Real - time decision making for IoT - based traffic light control. *Concurrency and Computation: Practice and Experience*, 29(11), 2017.
- [19] K. H. N. Bui, D. Camacho, and J. E. Jung. Real-time traffic flow management based on inter-object communication: a case study at intersection. *Mobile Networks and Applications*, 22(4), 613-624, 2017.
- [20] F. Daneshfar, J. RavanJamJah, F. Mansoori, H. Bevrani, and B. Z. Azami. Adaptive fuzzy urban traffic flow control using a cooperative multi-agent system based on two stage fuzzy clustering. In *Vehicular Technology Conference, 2009. VTC Spring 2009. IEEE 69th* (pp. 1-5). IEEE, 2009.
- [21] J. C. Burguillo-Rial, P. S. Rodríguez-Hernández, E. C. Montenegro, and F. G. Castiñeira. History-based self-organizing traffic lights. *Computing and Informatics*, 28(2), 157-168, 2012.
- [22] A. Guerrero-Ibanez, J. Contreras-Castillo, R. Buenrostro, A. B. Marti, and A. R. Muñoz. A policy-based multi-agent management approach for intelligent traffic-light control. In *Intelligent Vehicles Symposium (IV), 2010 IEEE*, pp. 694-699. IEEE.
- [23] S. Abar, G. K. Theodoropoulos, P. Lemarinier, and G. M. O'Hare. Agent based modelling and simulation tools: a review of the state-of-art software. *Computer Science Review*, 24, 13-33, 2017.
- [24] M. Wooldridge. An introduction to multiagent systems. John Wiley & Sons, 2009.
- [25] A. Zeid, A UML Profile for Agent-Based Development, in: *Lecture Notes in Computer Science*, vol. 2641, 2003, pp. 161-170.
- [26] E. Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences*, 99 (suppl 3), 7280-7287, 2002.
- [27] A. Banos, C. Lang, and N. Marilleau. Agent-based spatial simulation with NetLogo, vol. 1, Elsevier, 2015.
- [28] U. Wilensky, and W. Rand. An introduction to agent-based modeling: modeling natural, social, and engineered complex systems with NetLogo. MIT Press, 2015.