



CREACIÓN DE UN ALGORITMO DE INTELIGENCIA DE ENJAMBRES PARA LA SOLUCIÓN DE PROBLEMAS DE MOVILIDAD URBANA

CONVOCATORIA: Extraordinaria, junio 2024

MODALIDAD DEL TFG: Convencional

ALUMNO / A: Álvaro Martínez Parpolowicz

TUTOR / A: Dr. D. Georgy Nuzhdin Gelfand

GRADO: Matemática computacional

ÍNDICE

| | |
|---|-----------|
| RESUMEN | 4 |
| ABSTRACT..... | 5 |
| 1. INTRODUCCIÓN | 6 |
| 1.1 Motivación y contexto | 6 |
| 1.2 Planteamiento del problema..... | 6 |
| 1.3 Objetivos del trabajo | 7 |
| 2. ESTADO DE LA CUESTIÓN | 9 |
| 2.1 Marco Teórico del trabajo..... | 9 |
| 2.1.1 Sistemas de Transporte Inteligente | 9 |
| 2.1.2 Simulaciones de tráfico..... | 11 |
| 2.1.3 Inteligencia de enjambres | 11 |
| 2.1.4 Distancia de seguridad | 22 |
| 2.1.5 Relación entre agentes: aspectos físicos relevantes | 24 |
| 2.2 Trabajos relacionados. | 26 |
| 2.2.1 An Inverted Ant Colony Optimization approach to traffic | 26 |
| 2.2.2 Inverse Pheromone-based Decentralized Route Guidance | 27 |
| 2.2.3 An improved pheromone-based vehicle rerouting system to reduce traffic congestion | 28 |
| 2.2.4 Proactive eco-friendly pheromone-based green vehicle routing for multi-agent systems | 29 |
| 3. ASPECTOS METODOLÓGICOS | 31 |
| 3.1 Metodología | 31 |
| 3.1.1 Fase de investigación | 31 |
| 3.1.2 Formulación de la hipótesis | 31 |
| 3.1.3 Diseño del experimento | 31 |
| 3.1.4 Fase de experimentación..... | 32 |

| | | |
|-------|--|----|
| 3.1.5 | Análisis de los resultados | 32 |
| 3.2 | Tecnologías empleadas | 32 |
| 3.2.1 | Lenguaje de programación..... | 32 |
| 3.2.2 | Simulation of Urban Mobility..... | 33 |
| 3.2.3 | AITrafficLab | 33 |
| 4. | DESARROLLO DEL TRABAJO | 34 |
| 4.1 | Análisis de los algoritmos existentes | 34 |
| 4.2 | Elección del algoritmo base y planteamiento de modificaciones | 36 |
| 4.3 | Propuesta de idea: propagación de señales de comunicación | 37 |
| 4.4 | Estudio de un caso simplificado | 38 |
| 4.5 | Propagación de feromonas | 44 |
| 4.6 | Formulación de la propagación de feromonas sónicas | 47 |
| 4.7 | Modelado de la difusión de la señal en intersecciones | 51 |
| 4.8 | El coeficiente de absorción acústica | 54 |
| 4.9 | Formulación completa del algoritmo | 57 |
| 4.10 | Entorno de simulación y refinamiento de hiperparámetros del modelo | 58 |
| 4.11 | Análisis de los resultados..... | 61 |
| 5. | CONCLUSIONES | 72 |
| 5.1 | Resultados | 72 |
| 5.2 | Posibles líneas de trabajo futuras | 74 |
| 5.3 | Consideraciones finales | 76 |
| 6. | REFERENCIAS..... | 77 |
| 6.1 | Bibliografía. | 77 |
| 6.2 | Índice de imágenes..... | 81 |
| 6.3 | Índice de tablas | 83 |

RESUMEN

El desequilibrio entre los ritmos de crecimiento del número de vehículos y evolución de las infraestructuras en las grandes ciudades hace necesaria la búsqueda de soluciones capaces de gestionar el tráfico vehicular de manera óptima. Este Trabajo de Fin de Grado propone el desarrollo del algoritmo *Sonic Inverted Ant Colony Optimization* (SIACO), una versión modificada de un algoritmo de inteligencia de enjambres de la familia de algoritmos de optimización por colonias de hormigas. Su objetivo es dar solución a este problema prestando especial atención a la viabilidad real de implementación a gran escala.

La idea principal detrás de este algoritmo bio-inspirado es alterar el funcionamiento básico de las feromonas depositadas por las hormigas, de tal manera que su comportamiento se asemeje a la propagación de ondas sonoras en lugar de al original fenómeno químico. Los resultados apuntan a que el rendimiento del algoritmo desarrollado no difiere en gran medida del de los algoritmos del estado del arte, pero sí presenta mejoras en términos de eficiencia o seguridad.

ABSTRACT

The imbalance between the growth rates of the number of vehicles and the evolution of infrastructure in large cities claims for the search for solutions capable of optimally managing vehicular traffic. This Bachelor's Degree Final Project proposes the development of the Sonic Inverted Ant Colony Optimization (SIACO) algorithm, a modified version of a swarm intelligence algorithm from the Ant Colony Optimization (ACO) algorithm family. Its goal is to address this problem, focusing on the real feasibility of large-scale implementation.

The main idea behind this bio-inspired algorithm is to alter the fundamental behavior of the pheromones deposited by ants so that their mechanism resembles the propagation of sound waves rather than the original chemical phenomenon. The results indicate that the performance of the developed algorithm does not differ significantly from state-of-the-art algorithms, but it does present improvements in terms of efficiency or safety.

1. INTRODUCCIÓN

1.1 Motivación y contexto

La congestión de tráfico en las grandes ciudades se ha convertido en un problema en muchos aspectos de la vida de sus ciudadanos. No se trata solo de las consecuencias económicas y climáticas que provocan los grandes atascos de las horas punta, sino que, además, se está creando una crisis de salud pública debido tanto a la reducción de la calidad del aire (Jakubiak-Lasocka et al., 2014) como al aumento de los niveles de estrés (Hennessy & Wiesenthal, 1997).

Las nuevas tecnologías como *Internet of Things* (IoT), *edge computing* y la inteligencia artificial, han permitido diversas aproximaciones prometedoras al problema (Modi et al., 2021). En la intersección de las tres se encuentra una familia de algoritmos conocidos como “Inteligencia de Enjambres” (o *Swarm Optimization* en inglés). Se trata además de un tema que no se suele estudiar en profundidad en la carrera universitaria con, a priori, potencial suficiente para gestionar problemas tan complejos como la reducción de las congestiones vehiculares en entornos urbanos.

1.2 Planteamiento del problema

Desde un punto de vista matemático el problema se plantea de la siguiente manera:

Se puede representar una red vial como un grafo dirigido $G = (N, E)$, donde N representa el conjunto de nodos o intersecciones de la red y $E = \{(i, j) \mid i, j \in N \times N, i \neq j\}$ es el conjunto de aristas o vías entre dichos nodos. Existe, además, un tercer conjunto V que representa los vehículos que circulan por dicha red. Cada vehículo v tiene un par $(src, dst) \mid src, dst \in N \times N$, con el origen y el destino de su viaje. Este conjunto varía dinámicamente con el tiempo, puesto que aumenta cuando un nuevo vehículo comienza una ruta.

Con esta formulación, la resolución del problema se basa en la minimización del tiempo medio de viaje de todos los vehículos (Nguyen & Jung, 2021):

$$\text{Minimizar } F = \frac{1}{|V|} \sum_v \sum_{i,j}^N x_{ij}^v T_{ij}^v$$

$$\text{Sujeto a: } x_{ij}^v \in \{0,1\}; T_{ij}^v > 0 \forall i \neq j, i, j \in N$$

Donde x_{ij}^v representa si el vehículo v recorre la arista (i,j) y T_{ij}^v representa el tiempo tarda v en recorrer (i,j) .

Obtener analíticamente el valor de T_{ij}^v resulta complejo, puesto que depende de numerosos factores. Para resolver el problema del tráfico en las grandes ciudades, se propone entonces complementar esta búsqueda con el uso de distintas técnicas de computación. Así, varios requisitos adquieren una relevancia considerable si se pretende diseñar una solución viable para ser implementada:

1. Siendo poco realista la demolición, replanificación y reconstrucción de áreas urbanas, el problema está limitado por las características de las redes urbanas existentes. No obstante, la solución propuesta no debe ser dependiente de la topología concreta de la red urbana, el tamaño de esta o la cantidad de conductores o usuarios de la vía.
2. Si se quiere extrapolar la solución a todo tipo de redes, nuevamente, vuelve a ser poco realista disponer de capacidad de computación centralizada ilimitada que pudiera gestionar ciudades de características desconocidas. Por lo que el uso de técnicas compatibles con el escalado horizontal es indispensable.
3. Es imperativo atender al carácter, altamente sensible, de los datos manejados. La gestión del tráfico implica, en última instancia, manipular y conocer la ubicación en tiempo real de los usuarios y entidades de un entorno urbano.
4. Se han de tener en cuenta otros factores, propios de la naturaleza dinámica de este tipo de redes, como accidentes, cortes de calles o servicios de emergencia como ambulancias o policías.

1.3 Objetivos del trabajo

El objetivo principal de este trabajo es solucionar el problema de la gestión del tráfico en una ciudad a través del uso de la inteligencia de enjambres desde una óptica realista en lo que a la viabilidad de su implementación se refiere.

Para conseguir dicho fin se proponen los siguientes objetivos específicos:

1. Analizar los algoritmos de inteligencia de enjambres existentes.
2. Detectar aquellos aspectos que dificulten la puesta en marcha de dichas soluciones a gran escala.

3. Desarrollar un algoritmo, basado en los anteriores, que solucione en la medida de lo posible las desventajas encontradas.
4. Realizar un análisis comparativo entre los algoritmos existentes y el desarrollado.

Quedan por lo tanto fuera del alcance del trabajo:

- Definición, implementación o modificación de simuladores de tráfico o sus reglas para la gestión de vehículos.
- Debido a la falta de datos microscópicos, como rutas exactas seguidas por conductores o sus momentos de partida y llegada, y, a la complejidad de su obtención (por entrar en conflicto con la privacidad del usuario); también se excluye la definición o implementación de modelos predictivos que pudieran requerir la ingesta de este tipo de datos para su entrenamiento.

2. ESTADO DE LA CUESTIÓN

En este capítulo se presentarán diversos aspectos académicos íntimamente relacionados con el objeto de estudio. Por otro lado, se incluye una recopilación y análisis de investigaciones similares realizadas en el ámbito, con el doble objetivo de encontrar soluciones factibles o recabar características comunes a las soluciones encontradas para crear una propia.

2.1 Marco Teórico del trabajo.

2.1.1 Sistemas de Transporte Inteligente

Un Sistema de Transporte Inteligente, del inglés *Intelligent Transportation System* (ITS), se puede definir como el uso de las tecnologías de la información en el área del transporte terrestre para mejorar la calidad de vida de los ciudadanos en cuestiones de seguridad y la movilidad, al mismo tiempo que se disminuye el impacto ambiental del transporte (Mallik, 2014).

Existen multitud de trayectorias de aproximación a este tipo de sistemas dependiendo de las variables que se consideran modificables o alterables: control semafórico, monitorizando y sincronizando los distintos semáforos de una red para adaptar la duración de cada disco según las necesidades concretas del momento; diseño de la red de transporte, planificando de manera inteligente ampliaciones de las redes existentes; control sobre la toma de decisiones, utilizando estrategias de planificación de rutas que distribuyan homogéneamente los vehículos por la red para evitar aglomeraciones.

A su vez, dentro del campo de control sobre las rutas existen diversas aproximaciones de las cuales se creen relevantes para el objeto de estudio dos: *Multiple Traveling Salesman Problem* (mTSP), junto con su variante *Dynamic Vehicle Routing Problem* (DVRP) (Bektas, 2006), y *Traffic Assignment Problem* (TAP) (Patriksson, 1994).

El primero es una variación del conocido problema del viajante o problema del vendedor viajero (del inglés “*Traveling Salesman Problem*”). En el problema original, dado un grafo conexo con V vértices, un viajero busca encontrar el camino cerrado más corto que pasa exactamente una vez por cada uno de sus vértices (en principio ciudades). La variación que propone el problema mTSP es la siguiente:

- Existen n viajeros distintos, que comienzan en el mismo nodo inicial.

- Cada nodo v debe ser visitado al menos una vez por algún viajero.
- Los viajeros deben acabar su recorrido en el nodo inicial (camino cerrado).

La principal aplicación del problema en el campo de la gestión del tráfico es la del cálculo de rutas óptimas para un sistema de vehículos itinerantes (viajeros) que recogen pasajeros en determinados nodos. Este concepto de sistema de taxis o coches autónomos pretende reducir la densidad de tráfico mediante la disminución directa del número de vehículos en la red. Acepta también variaciones o restricciones como la capacidad de pasajeros del vehículo o usar el orden de llegada como orden de recogida. Estas características, junto con la posibilidad de crear dinámicamente nuevas necesidades de transporte a lo largo del tiempo, conforman el *Dynamic Vehicle Routing Problem*. No obstante, debido a la necesidad de limitar el alcance del trabajo, el estudio y resolución de estos problemas no se incluyen en este. El motivo de no elegir este enfoque radica en la necesidad de cambios socioculturales que permitieran a la sociedad deshacerse del concepto de vehículo propio muy arraigado actualmente.

Principal medio de transporte usado para ir al lugar habitual de trabajo de los españoles

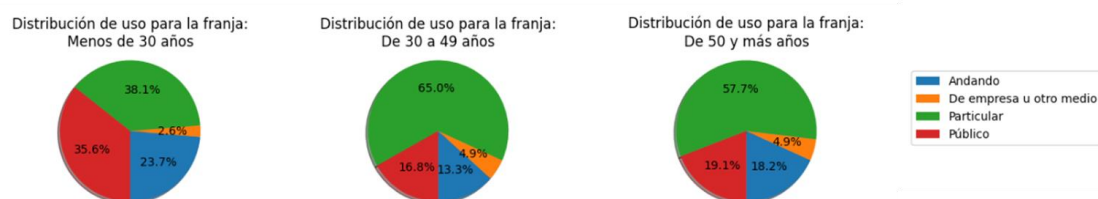


Ilustración 1. Principal medio de transporte usado por los españoles para ir a su lugar habitual de trabajo (Nota: Elaboración propia a partir de datos del Instituto Nacional de Estadística (INE), 2021).

Por otro lado, el TAP no pretende reducir la densidad vehicular general, sino distribuir los flujos de vehículos de manera homogénea a lo largo de la red para evitar atascos. Por lo que, dado un conjunto de parejas origen-destino, el objetivo es asignar a cada una de ellas la ruta óptima de acuerdo con cierto criterio como distancia mínima, tiempo de recorrido mínimo u otros más específicos (índices de contaminación, varianza de la aceleración, etc.).

Cabe destacar que la búsqueda de estas rutas depende de la elección del resto de rutas, puesto que, en general la métrica de coste empleada aumenta cuanto más congestionada esté una ruta concreta en una misma ventana de tiempo.

2.1.2 Simulaciones de tráfico

Atendiendo a la especificidad de los datos que se manejan se distinguen tres categorías. Su elección condicionará el nivel de generalidad del modelo construido, la capacidad de computación necesaria para llevar a cabo la simulación y el tipo de conclusiones que se pueden extraer del estudio. Estos son: nivel macroscópico, nivel microscópico y nivel mesoscópico (Dorokhin et al., 2020).

Una simulación de tráfico de nivel macroscópico busca representar los datos a gran escala. Generalmente se modelan los flujos de tráfico a lo largo de las calles, movimientos de unas zonas de una población a otras o directamente entre poblaciones.

Los datos necesarios para simulaciones microscópicas por su parte se oponen completamente a los anteriores. Estos buscan representar la información individual ofreciendo información específica como la velocidad instantánea, el tiempo exacto de trayecto o el estado concreto de las luces de tráfico en un momento dado. Además, se pueden obtener otros relacionados con el comportamiento de los conductores como la ruta escogida. Por lo tanto, este tipo de datos resultan de gran utilidad cuando se usan modelos que tienen en cuenta aspectos relacionados con la forma de conducción y las decisiones que se toman al volante.

Sin embargo, este nivel de detalle tiene sus propias desventajas, la más evidente de las cuales es el alto coste computacional. Esto se debe a que las simulaciones que utilizan este tipo de datos necesitan una gran cantidad de ellos. Algunos de estos datos problemáticos de obtener podrían ser el origen y destino de cada ruta particular o su hora exacta de salida y de llegada. Además, hacen falta numerosas simulaciones para obtener conclusiones estadísticamente significativas.

Finalmente, las simulaciones mesoscópicas son una mezcla de las dos ya descritas, que buscan paliar los defectos de ambas utilizando los puntos fuertes de la otra. De esta forma, las simulaciones de nivel mesoscópico representan información a nivel de vehículo individual, pero solo referente a datos estadísticos o agregados de este como la velocidad media del trayecto o el tiempo total empleado.

2.1.3 Inteligencia de enjambres

La inteligencia de enjambres (*Swarm Intelligence* o *Swarm Optimization* en inglés) es una rama de la inteligencia artificial que utiliza métodos bio-inspirados para la

resolución de problemas complejos mediante el uso de múltiples agentes que realizan tareas más simples (Chakraborty & Kar, 2017).

Se trata de algoritmos bio-inspirados porque surgen de la observación del comportamiento y las capacidades de algunas especies de seres vivos para resolver tareas complejas de manera colectiva. Algunos ejemplos de este tipo de tareas sería la construcción de lugares de resguardo, como madrigueras u hormigueros; defensa de la comunidad, frente a depredadores o inclemencias del tiempo; o búsqueda de recursos como comida, entre muchos otros. Para conseguir estas tareas cada individuo de la especie es incapaz de conseguirla por su cuenta. Es por esto por lo que se agrupan para lograrla colectivamente. De esta manera, cada uno realiza tareas considerablemente más simples, y, la potencia de este enfoque surge de las relaciones que se establecen entre dichos individuos.

Los algoritmos de inteligencia de enjambres comparten dos características comunes: la división de tareas y el carácter autoorganizado (Wahab et al., 2015). La división de tareas hace referencia a la ejecución simultánea de múltiples tareas simples por parte de cada una de las entidades del sistema. Por otro lado, la autoorganización del sistema se consigue cuando este es capaz en su conjunto de llevar a cabo la tarea compleja inicial sin ayuda externa. Por lo que son los propios individuos los que asumen los roles necesarios para llevar a cabo cada una de las tareas anteriores. Cabe destacar que la ejecución de estas tareas tiene un carácter descentralizado, es decir, no existe una figura central o principal que coordine a todas las demás o con la que todos los individuos se deban comunicar.

A continuación, se presentan algunos de las familias de algoritmos basados en inteligencia de enjambres más conocidos de acuerdo con Tang et al. (2021).

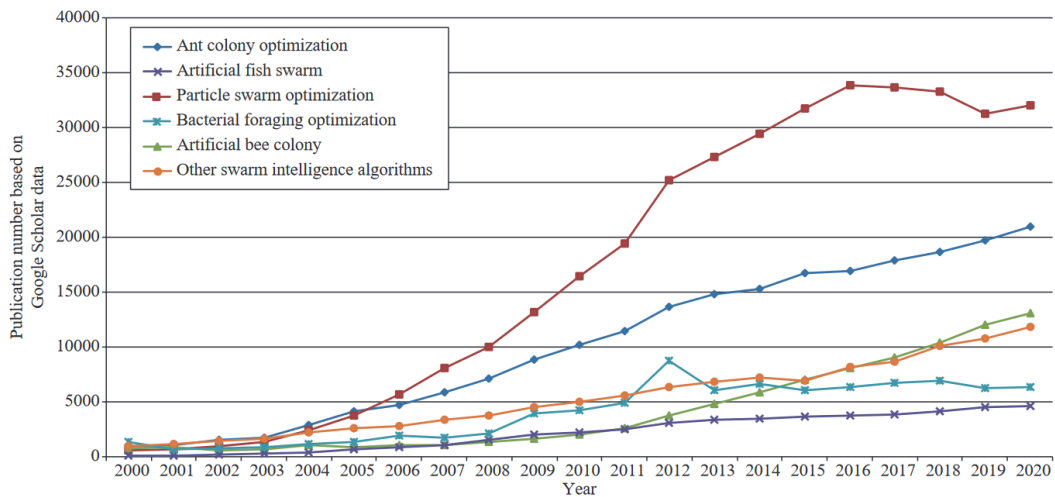


Ilustración 2. Indicador de Google Trends sobre el uso de algoritmos de inteligencia de enjambres estudiados en Tang et al. (2021) entre 2000 y 2020.

2.1.3.1 Particle Swarm Optimization

Si bien es cierto que los algoritmos de esta familia no son ampliamente utilizados en el contexto de la resolución de problemas de tráfico, dos sutiles aspectos si han servido de inspiración para el desarrollo del trabajo: la propagación de información entre partículas y la “componente social”. A continuación, se explican, no solo estos conceptos sino también otros necesarios para entender el algoritmo en su conjunto.

Los algoritmos de optimización por enjambre de partículas están basados en el algoritmo *Particle Swarm Optimization* (PSO) (Kennedy & Eberhart, 2002). Este algoritmo está inspirado en el comportamiento gregarios de algunas especies animales. Ejemplos clásicos animales que presentan estas conductas serían las bandadas de pájaros o los bancos de peces, que se agrupan en grandes números para protegerse de depredadores, para buscar alimento u otros motivos como la reproducción. Desvinculado de una especie concreta, este algoritmo denomina a cada individuo dentro del conjunto como “partícula”.

Este algoritmo permite resolver problemas optimización multidimensional de manera iterativa. Para ello cada partícula se modelará como un candidato o posible solución inicial al problema. Dichas partículas “explorarán” el espacio de soluciones basándose en diversas normas de comportamiento y leyes físicas. Para optimizar su valor, cada partícula utilizará tanto el “conocimiento” individual que genere como el generado por el conjunto o enjambre. Tras cada iteración, se evalúan los resultados de cada una de las partículas y se modifica el valor del óptimo global.

A continuación, se definen diversos aspectos relevantes para comprender el funcionamiento del algoritmo. Cada una de las explicaciones parte del aspecto biológico o físico en el que está inspirado.

Se considera que el comportamiento individual de los individuos del conjunto está determinado por tres simples reglas: separación, alineamiento y coherencia (Reynolds, 1987):

- Separación: cada partícula evita la colisión con otras partículas del conjunto.
- Alineamiento: cada partícula ajusta su velocidad y dirección para adecuarse a la velocidad y dirección promedio del conjunto.
- Coherencia: tendencia de las partículas a moverse en dirección al centro del conjunto.

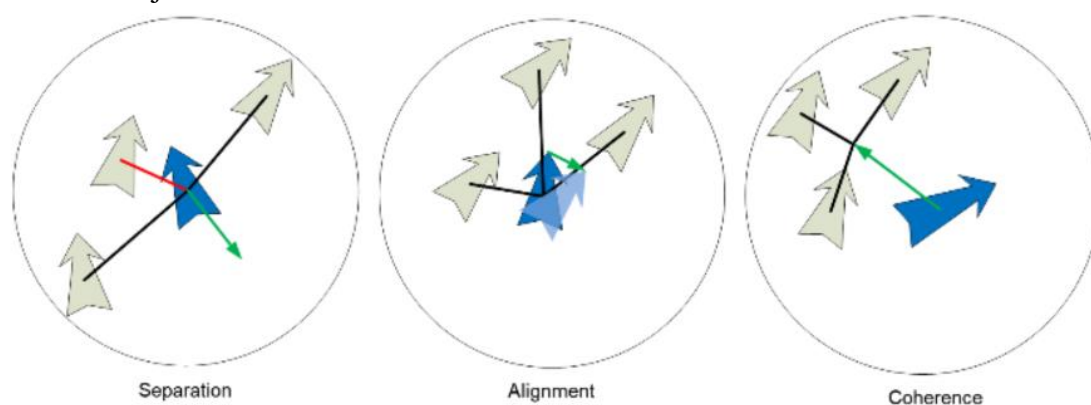


Ilustración 3. Reglas de las bandadas de pájaros: Separación, alineamiento y coherencia. (Khamis, 2024)

Mediante estas reglas, grandes conjuntos son capaces de responder a estímulos captados por individuos concretos. Así, si una partícula encuentra una ruta óptima hacia una fuente de alimento o divisa un depredador, modificará su comportamiento para adaptar su movimiento. Esto propagará, de manera indirecta, el estímulo al resto del conjunto, que podrán adaptarse para modificar sus respectivas trayectorias.

El objetivo, por tanto, del PSO es replicar este comportamiento para obtener un éxito similar. El algoritmo se define entonces de la siguiente manera:

Algoritmo 1. Particle Swarm Optimization (PSO)

Inicializar *enjambre* de partículas

Para cada *partícula* en el *enjambre*:

 Inicializar posición aleatoriamente

 Inicializar velocidad aleatoriamente

Fin Para

Mientras no se cumpla el criterio de parada:

Para cada *partícula* en el *enjambre*:

 Calcular *valor local* de la función en el punto

Si *valor* es mejor que el mejor *valor local* conocido por la *partícula*:

 Actualizar mejor *valor local* conocido

 Actualizar mejor *posición local* conocida

Si *valor* es mejor que el mejor *valor global* conocido por el enjambre:

 Actualizar mejor *valor global* conocido por el enjambre

 Actualizar mejor *posición global* conocido por el enjambre

Fin Si

Fin Si

Fin Para

Para cada *partícula* en el *enjambre*:

 Calcular nueva velocidad de la *partícula*

 Actualizar *posición* de la *partícula*

Fin Para

Fin Mientras

Devolver mejor *posición global*

Para ello, Khamis (2024) define el vector de movimiento de cada partícula del enjambre como:

$$v_{k+1}^{id} = \omega \times v_k^{id} + c1r1_k^d(pbest_k^{id} - x_k^{id}) + c2r2_k^d(gbest_k^d - x_k^{id})$$

Donde:

- k es el número de iteración.
- i y d son el número de partícula y dimensión respectivamente.
- v_k^{id} es el vector de movimiento de una partícula i , en una dimensión d en la iteración k .
- x_k^{id} es el vector de posición de una partícula i , en una dimensión d en la iteración k .
- ω es el vector de pesos de inercia.
- $c1, c2$ son los coeficientes de aceleración.
- $r1_k^d, r2_k^d$ son números aleatorios entre 0 y 1 generados en cada iteración k , comunes para todas las partículas en una misma iteración.
- $pbest_k^{id}$ es el mejor vector de posición conocido por una partícula i en una dimensión d , hasta la iteración k .

- $gbest_k^d$ es el mejor vector de posición conocido por todo el enjambre en una dimensión d, hasta la iteración k.

La ecuación anterior se divide en tres componentes que contribuyen al movimiento de una partícula: componente de inercia, cognitiva y social, respectivamente.

El primer término de la expresión se corresponde con la componente de inercia. Esta modela la incapacidad de un animal real de modificar bruscamente su trayectoria. Este término suaviza entonces las modificaciones que generaría la recepción de nuevos impulsos. El ajuste de los pesos del vector de inercia permite controlar la relación entre exploración y explotación.

La siguiente componente se conoce como componente cognitiva. Este término proporciona la partícula información sobre el conocimiento propio que ha adquirido hasta el momento. De esta manera, se fomenta la exploración de áreas que el agente considera relevantes, con el objetivo de refinar aún más los resultados obtenidos en una zona prometedora. Esta componente ignora el conocimiento colectivo.

Finalmente, el último término añade ese carácter colectivo al algoritmo, por lo que se le conoce como componente social. Esta componente atrae a las demás partículas hacia el mejor óptimo conocido hasta la fecha por el enjambre, lo que modificará sus comportamientos individualistas para que exploren los entornos más prometedores encontrados.

2.1.3.2 Ant Colony Optimization

La optimización por colonias de hormigas, *Ant Colony Optimization* (ACO), es otra de las familias de algoritmos de optimización por enjambres. Como su nombre indica, la lógica de estos algoritmos está basada en el comportamiento de las hormigas, particularmente en su proceso de búsqueda y recolección de alimentos. El primer algoritmo, precursor del resto de la familia, fue propuesto por Dorigo et al. (1996) y fue conocido como *Ant System* (AS). El objetivo de este algoritmo es encontrar la ruta de coste mínimo entre dos puntos.

En un entorno real, una hormiga es incapaz de resolver la tarea de buscar alimento por su cuenta. Por ello se coordina con otras hormigas para hacer una búsqueda conjunta que facilite la tarea. Sin embargo, dicha coordinación es imposible sin un método de comunicación apropiado. Para ello se introduce un nuevo concepto, diferente al de

agente o partícula vistos en el apartado anterior, que modela la interfaz de comunicación que existe entre hormigas reales: la feromona.

Definición 1: *Feromona.*

Sustancia química segregada por las hormigas (entre otras variedades de insectos y demás animales) cuya localización e intensidad puede ser percibida por otras hormigas.

Se puede **modelar digitalmente** mediante un valor numérico que contiene algún tipo de información. Dicho valor es almacenado por el entorno y puede ser modificado por las hormigas digitales.

Este compuesto que depositan en el medio se evapora paulatinamente, por lo que, si no hay otras hormigas que depositen encima sus propias feromonas, estas acabarán por desaparecer del medio con el tiempo. La comunicación entre hormigas se realiza entonces de manera indirecta, pues no existe una comunicación hormiga-hormiga. En su lugar existe la comunicación hormiga-entorno y entorno-hormiga. A este concepto se le conoce como estigmergia (Grasse, 1959):

Definición 2: *Estigmergia.*

Comunicación que permite la colaboración entre distintos agentes a través del medio físico.

Por medio de esta herramienta, las hormigas son capaces de marcar el camino a seguir para llegar a una fuente de alimento. Así, sucesivas hormigas que salen del hormiguero en busca de comida solo deben seguir el rastro de feromonas dejado por sus predecesoras.

Lo que requiere de nuestra atención y el principio en el que se basan los algoritmos de ACO, es el proceso de formación de dicho camino hasta la fuente de alimento:

1. Múltiples hormigas abandonan inicialmente el hormiguero en busca de comida. No existen feromonas previas depositadas, por lo que se produce una tarea de exploración.

2. Cuando una hormiga encuentra una fuente de alimento, dado que es capaz de recordar el camino que ha recorrido desde el hormiguero, vuelve a este depositando feromonas en su camino de vuelta.
3. Nuevas hormigas abandonan el hormiguero. A diferencia de la última vez, ahora disponen de una pista en forma de feromonas, dejadas por otras hormigas, sobre la ubicación y el camino a seguir hasta la fuente de alimento. Cada hormiga elegirá entonces un camino basándose en los niveles de feromona de este.
4. Las nuevas hormigas, en su retorno al hormiguero depositarán nuevas feromonas que reforzarán el camino que eligieron y compensarán aquellas que se evaporaron. Comienza de nuevo en el paso 3.

La explicación anterior puede ser representada también mediante el siguiente pseudocódigo:

Algoritmo 2. Ant Colony Optimization (ACO)

Inicializar *matriz de feromonas*

Definir *nodo de salida* y *nodo de llegada*

Mientras no se cumpla el criterio de parada:

Inicializar *hormigas* en el *nodo de salida*

Para cada *hormiga*:

Mientras la *hormiga* no haya alcanzado el *nodo de llegada*:

Calcular la probabilidad de transición a cada nodo adyacente

Seleccionar siguiente nodo

Actualizar lista de nodos visitados

Fin Mientras

Actualizar *matriz de feromonas* basado en la lista de nodos visitados

Fin Para

Evaporar feromonas

Fin Mientras

Seleccionar *mejor camino* basado en la *matriz de feromonas*

Devolver *mejor camino*

Con lo explicado hasta ahora, una colonia de hormigas es capaz de encontrar uno o varios caminos hasta una fuente de alimento. No obstante, este proceso también conlleva intrínsecamente la optimización de dicho camino hasta que sea el más corto posible. A este fenómeno se le conoce como comportamiento autocatalítico (Khamis, 2024) y sucede debido a que cuantas más hormigas recorren el camino más corto, más atractivo se vuelve para las siguientes hormigas. Este término deriva del concepto químico de autocatálisis: un proceso químico cuyos productos funcionan también como catalizadores de la propia reacción. Para comprender por qué se produce este fenómeno, consideremos el siguiente ejemplo:

- En el momento inicial ($t = 0$) dos hormigas exploradoras acaban de regresar al hormiguero desde una fuente de alimento por dos caminos distintos A y B. El camino A mide una unidad espacial, es decir, una hormiga es capaz de recorrer el camino A en una iteración. El camino B mide dos unidades espaciales. En $t = 0$, tanto el camino A como B tienen depositadas una unidad de feromona cada uno. Para este ejemplo no se contemplará de momento la evaporación de feromona.
- A continuación, en $t = 1$, dos nuevas hormigas abandonan el hormiguero para buscar comida. Como la intensidad de feromona en ambos caminos es idéntica, cada hormiga tiene un 50% de probabilidad de elegir cada camino. Supongamos que cada una elige uno distinto. Llamaremos a a la hormiga que eligió el camino A y b a la que eligió el B.
- En $t = 2$, la hormiga a habrá llegado al alimento, y la hormiga b estará a mitad de camino. La hormiga a emprende su camino de vuelta depositando una unidad de feromona. Si dos nuevas hormigas salieran en este momento, se produciría el mismo caso que en $t = 1$.
- En $t = 3$, la hormiga a regresa al hormiguero y el camino A tiene ahora depositadas dos unidades de feromona, mientras que el B tiene una. La hormiga b acaba de llegar a la fuente de comida y emprende su regreso. Si ahora dos nuevas hormigas salieran del hormiguero, tendrían una probabilidad de $2/3$ de elegir el camino A y $1/3$ de elegir B.

Con este pequeño ejemplo es fácil observar que, debido a que A es reforzado más a menudo, cada vez es más probable que una hormiga elija el camino A, por lo que es a su vez más susceptible de ser reforzado nuevamente. Ello no quita que, probabilísticamente hablando, el camino B no sea elegido nunca. Aquí es donde interviene la evaporación. Si en ambos caminos las feromonas se evaporan al mismo ritmo, el camino que sea reforzado menos veces acabará por desaparecer por completo. Este fenómeno puede apreciarse en la siguiente figura, donde, después de explorar, las hormigas acaban eligiendo el camino con mayor feromona:

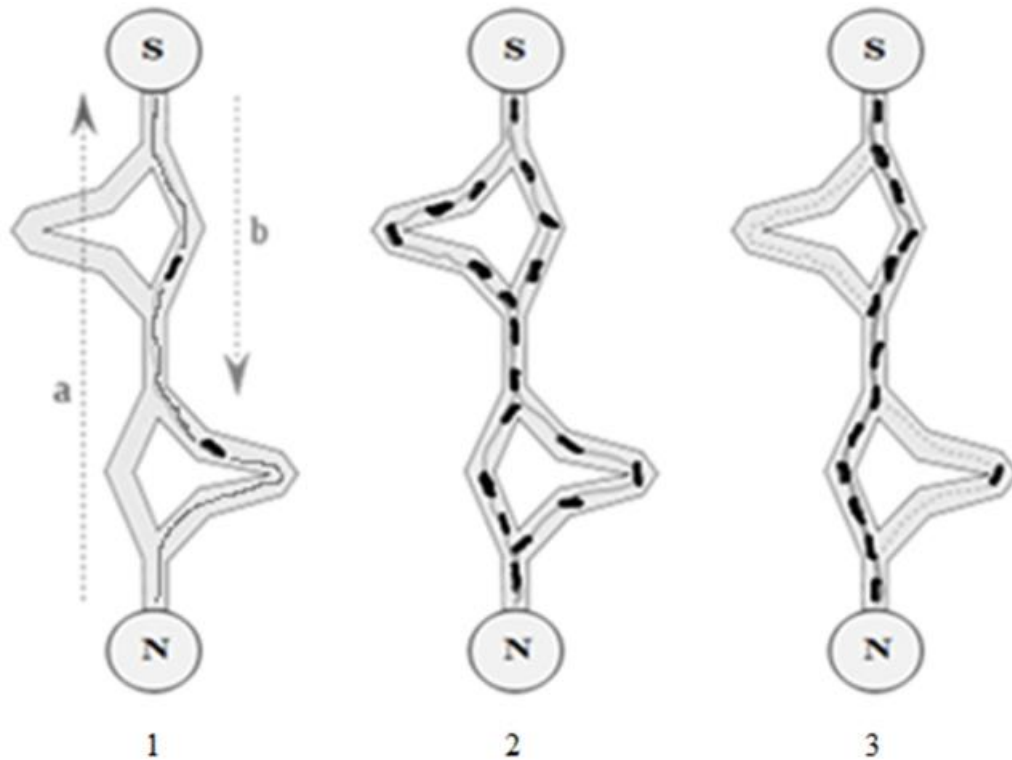


Ilustración 4. Proceso de optimización por colonias de hormigas. (Wahab et al., 2015)

2.1.3.3 Artificial Bee Colony

La tercera familia de algoritmos de optimización por enjambres más conocida es la optimización por colonias de abejas artificiales, *Artificial Bee Colony* (ABC) (Karaboga, 2005). De este algoritmo será relevante un único concepto: la comunicación directa e individual entre abejas inspirada por la “danza del aleteo”. Sin embargo, se explicarán también otros conceptos no tan relevantes para el desarrollo del trabajo para poder comprender el algoritmo en su conjunto.

Este algoritmo se centra de nuevo en las estrategias adoptadas, esta vez por enjambres de abejas melíferas, para la recolección de alimento. Para la implementación de este algoritmo se utilizan tres de las funciones llevadas a cabo por las abejas reales y se denominan: abeja exploradora, abeja trabajadora y abeja observadora.

Para la correcta coordinación de las tareas, tanto de abejas reales como artificiales, es necesaria la definición de una nueva interfaz de comunicación. Es esta ocasión, abandonamos el concepto de stigmergia para adoptar una comunicación directa abeja-abeja inspirada en la “danza del aleteo” o “danza de la abeja” (Von Frisch, 1950).

Mediante esta danza, las abejas denominadas “exploradoras” son capaces de informar a las abejas “observadoras” sobre la ubicación exacta de fuentes de alimento y su calidad. Es entonces cuando las abejas observadoras pueden realizar la función de recolección. Esta división de tareas evita que se desperdicie tiempo y esfuerzo por parte de la colmena en búsquedas constantes de alimento.

Las funciones desempeñadas por las abejas no son constantes, es decir, pueden variar en función de las necesidades de la colmena y la disponibilidad actual de alimento y fuentes de este. Las abejas trabajadoras están inspiradas en este concepto, de tal manera que una abeja exploradora que ha encontrado fuentes explotables regresa a la colmena y, tras informar a las abejas observadoras, asume la función de abeja trabajadora. Su función es convencer con su danza y guiar a abejas observadoras para visitar constantemente la misma fuente de alimento hasta que este se haya agotado. Una vez que se haya agotado, asumirá de nuevo la función de abeja exploradora y buscará nuevas flores.

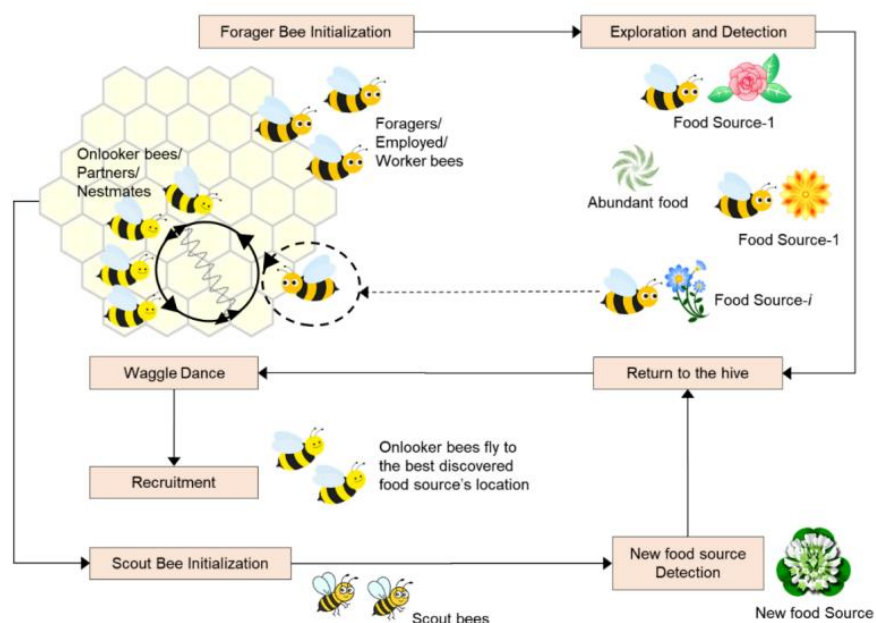


Ilustración 5. Algoritmo Artificial Bee Colony. (Khamis, 2024)

El proceso anterior se puede entender también con el siguiente pseudocódigo:

Algoritmo 3. Artificial Bee Colony (ABC)

Inicializar *abejas_exploradoras*
Inicializar *abejas_trabajadoras*
Inicializar *abejas_observadoras*
Definir *limite_máximo*

Mientras no se cumpla el criterio de parada:

Para cada *abeja_exploradora*:

 Obtener una *posición* aleatoria
 Convertir en abeja trabajadora
 Definir *entorno* basado en la *posición*
 calidad_entorno = Evaluar calidad de la solución

Fin Para

Para cada *abeja_observadora*:

 Elegir *entorno* de búsqueda basado en *calidad_entorno* de *abejas_trabajadoras*
 Obtener *posición* aleatoria en el *entorno*
 Evaluar calidad de la solución

Fin Para

Para cada *abeja_trabajadora*:

 Obtener *posición* aleatoria en el *entorno* propio
 Evaluar calidad de la solución
 nueva_calidad = Obtener la calidad máxima obtenida por todas las abejas en el entorno
 calidad_entorno = *nueva_calidad*
 Si *nueva_calidad* > *calidad_entorno*:

límite = 0

Si no:

límite = *limite* + 1

Si *límite* >= *límite_máximo*:

 Convertir en abeja exploradora

Fin Si

Fin Si

Fin Para

Fin Mientras

Devolver *posición* de *abeja_trabajadora* en el mejor entorno

2.1.4 Distancia de seguridad

De acuerdo con el artículo 22, apartado 2, de la Ley de Tráfico y Seguridad Vial:

“El conductor de un vehículo que circule detrás de otro debe dejar entre ambos un espacio libre que le permita detenerse, en caso de frenado brusco, sin colisionar con él, teniendo en cuenta especialmente la velocidad y las condiciones de adherencia y frenado”.

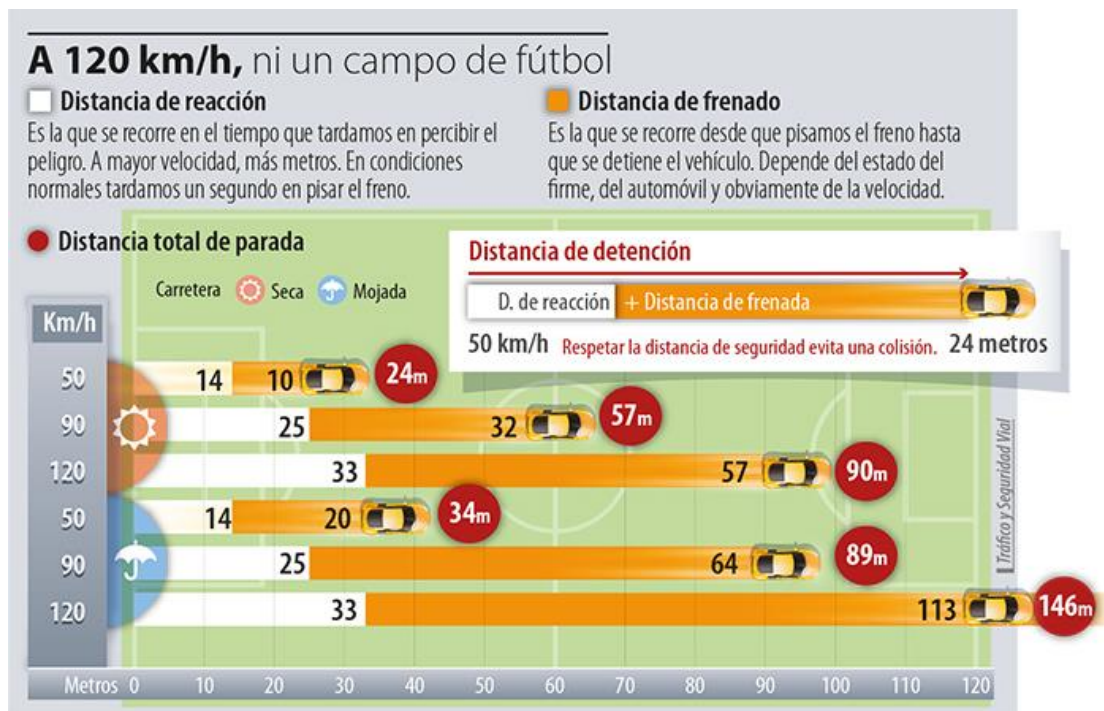


Ilustración 6. Velocidad, distancia de reacción y distancia de frenado (Dirección General de Tráfico [DGT], 2015)

Aunque no se especifica cuantitativamente en el texto legal, la Dirección General de Tráfico hace algunas recomendaciones sobre la forma de cuantificar la distancia de detención o de seguridad. La primera es la conocida como regla del cuadrado, donde se indica que un conductor debe guardar como distancia mínima con el coche que le precede (medida en metros) el resultado del cuadrado de la división entera de la velocidad actual del vehículo (en km/h) entre diez (Dirección General de Tráfico [DGT], 2023). Esta distancia debe duplicarse en calzadas mojadas. No obstante, dado que el cálculo de dicha operación y la visión espacial requieren ambos de esfuerzo mental, existe también otra norma, conocida como regla de los tres segundos. Esta, por otro lado, se basa en el cálculo de la distancia de seguridad utilizando la relación entre velocidad, distancia y tiempo. Así, recomiendan dejar una distancia temporal de entre dos y tres segundos (dependiendo de las condiciones climáticas y de la vía) con el vehículo que precede al conductor (Dirección General de Tráfico [DGT], 2013).

Aunque ambas reglas son algo imprecisas desde un punto de vista matemático, en este trabajo se utilizará la regla de los tres segundos.

Ahora bien, en lo que concierne al estudio, la distancia de seguridad cobra relevancia a la hora de medir la influencia individual de un vehículo sobre otro. Para comprender la forma de medir esta influencia, supongamos el siguiente caso: en condiciones ideales,

un vehículo A y un vehículo B que le sigue, circulan por una vía de un solo carril, sin posibilidad de adelantamiento.

Si ambos guardaran inicialmente, al menos, la distancia de seguridad correspondiente a la velocidad máxima de la vía ambos podrían circular a máxima velocidad sin influenciarse mutuamente. La única posibilidad de interacción en este caso es si el vehículo A reduce su velocidad por algún motivo.

En el momento en el que el vehículo B se encuentre a una distancia menor a la distancia de seguridad, este se verá obligado a reducir su propia velocidad hasta que esta sea suficiente para mantener una nueva distancia de seguridad proporcional. En este caso consideraremos que el vehículo B ha invadido el área de influencia del vehículo A, puesto que su velocidad se ve condicionada por la del vehículo A.

No sería hasta que el vehículo A acelerase hasta alcanzar la velocidad máxima y se restaurara la distancia de seguridad asociada a dicha velocidad, que el vehículo B podría acelerar libremente hasta esa misma velocidad, quedando entonces libre de la influencia de A.

Por lo tanto, el área de influencia individual de un vehículo sobre otro no depende del comportamiento de estos, sino de las características de la vía. Más concretamente de la velocidad máxima permitida.

Cabe destacar que esta distancia de influencia se mide desde el vehículo predecesor, en dirección de la vía y sentido opuesto a la marcha, ya que el vehículo B no puede influenciar en el comportamiento del conductor del vehículo A.

2.1.5 Relación entre agentes: aspectos físicos relevantes

A la hora de desarrollar un algoritmo, es conveniente que este requiera la menor cantidad de información posible para su correcto funcionamiento, sin que ello repercuta negativamente en su desempeño. Por este motivo se busca agregar gran parte de la información de la red para tratarla usando la mínima cantidad de datos posibles.

Es por ello necesario encontrar un equilibrio entre la información provista por pocas entidades como el vehículo de enfrente o la vía en la que se encuentra, puesto que proporciona información insuficiente; y, la totalidad de automóviles de una gran ciudad, ya que se trata de una cantidad de información inabarcable.

Con este objetivo en mente se ha buscado la forma de establecer una analogía con magnitudes físicas como la temperatura o la intensidad acústica. El motivo de establecer esta analogía está inspirado en otros fenómenos físicos que ya se modelan en los algoritmos de inteligencia de enjambres, como la mencionada “evaporación” de feromonas.

Precisamente, como se verá más adelante, ha sido necesaria la búsqueda de fenómenos capaces de sustituir la función de evaporación de feromonas. Si recordamos, el concepto de feromona se usaba en los algoritmos de la familia ACO como memoria del sistema sobre la cantidad de hormigas que habían elegido determinada ruta y el tiempo que hacía que la habían seleccionado. Así, la función de evaporación dotaba al sistema de la capacidad de “olvidar” información con el tiempo para evitar que nuevas hormigas recorriesen caminos que hormigas previas descubrieron que eran poco rentables. Las investigaciones realizadas en este ámbito se centran por tanto en encontrar funciones existentes que permitan replicar este comportamiento.

Hemos planteado dos soluciones prometedoras: la atenuación de la intensidad acústica en su propagación debida a las características de un medio, conocida como fenómeno de absorción acústica; y, la disipación del calor debido a la transmisión de energía térmica de regiones de mayor temperatura hacia otras con menor temperatura, conocida como difusión del calor. De entre estas, solo se desarrollará la idea de propagación de ondas sonoras, por lo que es conveniente definir primero algunos conceptos. No obstante, también se comentarán más adelante algunos aspectos relacionados con la otra idea y el motivo de su descarte.

Para comprender la pérdida de energía acústica a medida que una onda se propaga es necesario conocer distintos aspectos. Por un lado, la potencia sonora se define como la cantidad de energía acústica emitida por una fuente por unidad de tiempo (Kinsler et al., 2000). Es un concepto relacionado con la intensidad acústica, pero no el mismo. La intensidad acústica hace referencia a la densidad de potencia por unidad de superficie. Están relacionadas siguiendo la siguiente ecuación:

$$P = I S$$

Donde P se refiere a la potencia, I a la intensidad y S es la superficie. Observando esta ecuación es razonable llegar a la conclusión de que si la cantidad de energía emitida por una fuente se mantiene constante en el tiempo (potencia constante), la intensidad

disminuye a medida que aumenta la superficie. A este fenómeno se le conoce como atenuación y se debe precisamente a que a medida que se propaga una onda, su energía se reparte cada vez entre mayor superficie. Esto puede modelarse mediante:

$$\text{Ondas planas: } P = I r; I_1 r_1 = I_2 r_2$$

$$\text{Ondas tridimensionales: } P = I r^2; I_1 r_1^2 = I_2 r_2^2$$

Donde ahora r es el radio o distancia al foco de emisión de la onda. La atenuación reduce entonces la intensidad sonora si a medida que se propaga, pues cada vez abarca mayor superficie. No obstante, para el caso de ondas unidimensionales (como la propagación de una onda a través de un cable vibrando), la intensidad se mantiene constante, puesto que la superficie de reparto de energía se mantiene también constante a medida que la onda se aleja del foco de emisión.

Existe por otro lado, otro fenómeno que causa la disminución de la intensidad sonora a medida que esta se propaga. Esta vez el causante de esta disipación es el propio medio en el que se propaga la onda. Las causas de esta disipación pueden dividirse en dos categorías: aquellas intrínsecas al medio y aquellas relacionadas con los límites del medio (Kinsler et al., 2000). Para comprender los desarrollos más adelante efectuados, nos centraremos en comprender la absorción acústica producida por el propio medio: A medida que una onda se propaga por un medio, parte de su energía se disipa en forma de calor debido al rozamiento. La disipación es por tanto mayor según esta avanza por el medio en cuestión. Puede modelarse de la siguiente manera:

$$I = I_0 e^{-\alpha r}$$

Donde ahora I_0 representa la intensidad inicial, α es el coeficiente de absorción acústica del medio y r vuelve a ser la distancia recorrida por la onda a través del medio.

2.2 Trabajos relacionados.

En este apartado se muestran y analizan otros trabajos desarrollados previos que han abordado una problemática similar a este estudio.

2.2.1 An Inverted Ant Colony Optimization approach to traffic

Hasta ahora la deposición de feromonas por parte de una hormiga implicaba un único hecho, otra hormiga había tomado esa ruta para llegar a su destino en un momento anterior. Dicha información era utilizada por el resto de las hormigas para seguir

aquellos caminos más populares. Así, se podría decir que las hormigas reales son una especie animal especializada en formar un gran atasco ordenado.

Este artículo propone entonces una interesante modificación del algoritmo básico de ACO para lograr resolver el problema TAP. La idea principal se basa en invertir el comportamiento de las feromonas depositadas por las hormigas: ahora las feromonas, en lugar de atraer a otras hormigas, repelen a sus homónimas (Dias et al., 2014). Lo que bautizan como *Inverted Ant Colony Optimization* (IACO).

Esta modificación tiene todo el sentido, ya que usando la misma información que ACO (la feromona indica que una hormiga previa ha tenido que depositarla), consigue precisamente que las hormigas (en este caso vehículos) lleguen a sus respectivos destinos evitándose las unas a las otras.

No obstante, dado que las hormigas necesitan un objetivo más allá de evitarse las unas a las otras, se opta por un algoritmo híbrido que combina el clásico algoritmo de Dijkstra con esta modificación del modelo ACO. En el artículo sugieren que esto atiende a un propósito doble, orientar a las hormigas hacia su destino y evitar que en situaciones sin tráfico (como primera hora de la mañana) las hormigas se pierdan al carecer de información de feromonas dejada por hormigas previas (Dias et al., 2014).

Sin embargo, este artículo deja algunos aspectos, relacionados con su implementación, sin aclarar, como la especificación de la función de deposición de feromona.

$$\text{Acumulación de feromonas: } pher_{edge} = pher_{edge} + DEP_PHER$$

$$\text{Evaporación de feromonas: } pher_{edge} = pher_{edge} - minTT_{edge} * DEP_PHER$$

Donde $pher_{edge}$ es el nivel de feromona de la arista $edge$, $minTT_{edge}$ el tiempo mínimo para recorrer una arista y DEP_PHER es desconocido. Más adelante se estudia cómo resolver este problema.

2.2.2 Inverse Pheromone-based Decentralized Route Guidance for Connected Vehicles

Este artículo propone un enfoque muy similar al anterior, basándose en el concepto de feromona invertida. Se ha incluido porque, acusando también las carencias del artículo anterior, busca proponer una solución completa, aunque ligeramente distinta (Nguyen & Jung, 2021).

En esta ocasión se utiliza un enfoque probabilístico eligiendo la siguiente arista del camino basado en:

$$P_{ij}^v(t) = \frac{[\tau_{ij}(t)^{-1}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{l \in allowed} [\tau_{il}(t)^{-1}]^\alpha \cdot [\eta_{il}]^\beta}$$

Donde $P_{ij}^v(t)$ es la probabilidad de elegir la arista (i, j) en el momento t , $\tau_{ij}(t)$ el nivel de feromona en (i, j) en el tiempo t , η_{ij} es la inversa de la longitud de (i, j) , y, α y β ajustan la relevancia de cada componente (Nguyen & Jung, 2021).

Para atajar el problema de orientar las hormigas hacia su destino, en lugar de usar el algoritmo de Dijkstra, limitan las posibles aristas que se pueden elegir mediante el conjunto *allowed*. Esto evita que los vehículos vaguen sin rumbo por la red, condicionando así sus decisiones (Nguyen & Jung, 2021). Sin embargo, el método de obtención de este conjunto vuelve a no estar definido en el artículo. Una vez más, se propondrá una solución más adelante.

2.2.3 An improved pheromone-based vehicle rerouting system to reduce traffic congestion

Este artículo propone la creación de un algoritmo, basado también en hormigas, denominado *Proactive Travel-time based Pheromone Rerouting* (PTPR). A diferencia de los algoritmos anteriores, que tenían un carácter reactivo, este propone un comportamiento predictivo (Ho et al., 2019). Esto implica que las hormigas no reaccionan a cambios en el entorno modificando sus rutas, sino que planifican sus rutas con la información disponible en el presente para evitar futuras congestiones.

Para conseguir dicho enfoque predictivo proponen un modelo basado en redes neuronales recurrentes de tipo *Long Short-Term Memory* (LSTM) (Hochreiter & Schmidhuber, 1997) de acuerdo con las modificaciones propuestas por Gers et al. (2000). Esta red toma como información tres tipos distintos de feromonas para predecir el tráfico futuro. Los tres tipos de feromonas son: la feromona de tiempo de viaje, que depositan los vehículos basándose en el tiempo que llevan en su calle actual; la feromona de densidad, depositada en base a la densidad vehicular actual de la calle (calculada en función de la cantidad de vehículos y las dimensiones de la calle); y, la feromona de densidad de flujo, un concepto nuevo y algo más abstracto que se basa en la cantidad de agentes que han proporcionado información al sistema en dicha iteración

(Ho et al., 2019). Definiendo reglas sobre qué vehículos pueden depositar feromonas en cada momento el último tipo de feromona permite tener en cuenta información sobre otros factores como las luces de tráfico (ejemplo propuesto: los coches ante un semáforo en rojo no pueden depositar feromona (Ho et al., 2019)).

Otras modificaciones interesantes que proponen son la capacidad de una hormiga para depositar feromonas en aristas que no son inmediatamente adyacentes a su posición en un momento dado, o la proposición de una variación del algoritmo de k-caminos más cortos de Yen (Yen, 1971) para que adquiera un carácter localizado y dinámico, reduciendo así el coste computacional (Ho et al., 2019).

2.2.4 Proactive eco-friendly pheromone-based green vehicle routing for multi-agent systems

Esta publicación ataca el problema desde una perspectiva algo distinta, se busca minimizar el tiempo que un vehículo se encuentra detenido ante un semáforo. El motivo de elección principal de este enfoque radica en la reducción de los niveles contaminación más que en una reducción del tiempo de viaje (Soon et al., 2019).

Nuevamente, el algoritmo *Eco-friendly Pheromone-based Green Vehicle Routing* (E-PGVR) utiliza un enfoque predictivo para evitar la congestión vehicular y, esta vez, reducir al máximo las emisiones contaminantes. Para ello utiliza información sobre la intensidad de feromona depositada (con un efecto de repulsión también), la velocidad media de la carretera y la duración restante del disco verde del próximo semáforo. Con esto consigue elaborar una predicción a corto plazo del comportamiento del tráfico que utiliza para modificar iterativamente la ruta del vehículo (Soon et al., 2019).

No obstante, uno de los aspectos más llamativos es la óptica realista de la interfaz de comunicación que existe entre las diferentes entidades del sistema. Este algoritmo, además de tener un carácter descentralizado, intrínseco a su naturaleza como algoritmo de inteligencia de enjambres, busca limitar la cantidad de información de la que disponen los agentes. Con esto se pretende aumentar la escalabilidad, eficiencia y robustez del algoritmo. Esto facilitaría la implementación en un entorno real de un sistema que emplease este modelo. Dicha comunicación tiene una estructura jerarquizada con cuatro tipos de agentes repartidos en tres niveles. Este modelo ha sido nombrado como *Hierarchical Multi-Agent Pheromone-based System* (HMAPS) (Soon et al., 2019).

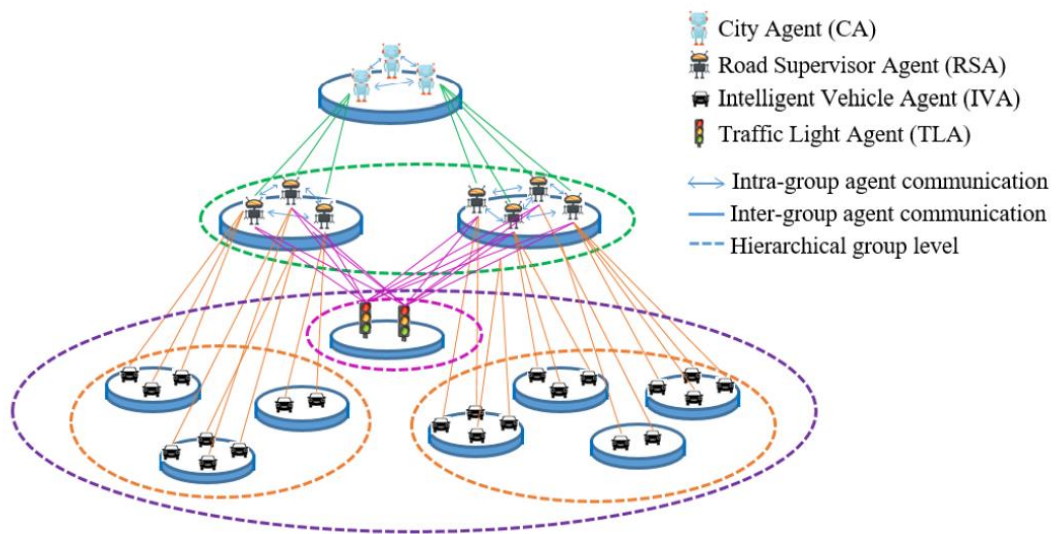


Ilustración 7. Hierarchical Multi-Agent Pheromone-based System (HMAPS) (Soon et al., 2019).

3. ASPECTOS METODOLÓGICOS

En este capítulo se presenta el planteamiento de las distintas fases de investigación a seguir. Se desarrollará cada una de las etapas del proyecto y se explicará la relación que guardan dichas fases con cada uno de los objetivos planteados al inicio del documento.

Por otro lado, se introducirán brevemente las tecnologías utilizadas, haciendo énfasis en el motivo de su elección.

3.1 Metodología

El proyecto ha sido dividido en diversas fases, inspiradas en el método científico, para facilitar el avance pautado de la investigación.

3.1.1 Fase de investigación

Partiendo de que el problema a resolver y los objetivos ya han sido planteados se lleva a cabo la fase de investigación. Tras estudiar múltiples algoritmos se evalúan las ventajas e inconvenientes que plantean cada uno de ellos. A continuación, se estudiará la posibilidad de combinar distintas características de algunos de ellos con el objetivo de proponer alguna mejora.

3.1.2 Formulación de la hipótesis

Con los resultados del apartado anterior se procederá a la formulación de un algoritmo que compense algunas de las desventajas de los algoritmos del estado del arte.

La hipótesis se basará entonces en determinar si las modificaciones propuestas son capaces de mejorar en alguno de los aspectos o métricas descubiertas en el apartado anterior, alguno de los modelos en los que se basan.

3.1.3 Diseño del experimento

Tras describir el algoritmo y plantear la hipótesis se pretende diseñar un experimento capaz de ofrecer evidencias suficientes para aceptar o no dicha hipótesis.

En esta fase se implementarán además los algoritmos necesarios para su posterior estudio.

3.1.4 Fase de experimentación

Una vez preparado el entorno de simulación se probarán todos los algoritmos implementados. En caso de que fuera necesario, esta experimentación se realizará de manera iterativa, refinando el valor de los hiperparámetros de los modelos que así lo requirieran.

Una vez se disponga de los modelos finales, se recogerán los datos pertinentes.

3.1.5 Análisis de los resultados

Con los datos recolectados en la fase anterior se realizarán distintos estudios estadísticos que permitan validar la hipótesis inicial. En función de los resultados observados se obtendrán conclusiones pertinentes, analizando las posibles causas y sugiriendo líneas de mejora.

3.2 Tecnologías empleadas

En este apartado se listan y describen las diferentes tecnologías que han sido necesarias para el desarrollo del trabajo.

3.2.1 Lenguaje de programación

El lenguaje de programación elegido para llevar a cabo las implementaciones y estudios estadísticos ha sido Python en su versión 3.10.9 (Python Software Foundation, 2023).

Inicialmente se contemplaron otras opciones como C++, debido a que los simuladores de tráfico estudiados están escritos en este lenguaje; R, dado que era previsible tener que llevar a cabo estudios estadísticos y análisis de resultados; o alguna combinación de estos tres lenguajes. No obstante, se eligió utilizar exclusivamente el primero debido, principalmente a su extensa adopción en el mundo del aprendizaje automático. Evitar el uso combinado de distintos lenguajes facilita también el desarrollo y su posterior uso por otros estudios.

Cabe destacar además el uso de algunas librerías que no forman parte del lenguaje base que facilitarán el desarrollo del trabajo: Optuna, para la optimización de hiperparámetros de los modelos implementados (Akiba et al., 2019), Scipy para la realización de estudios estadísticos (Virtanen et al., 2020) y Matplotlib para la visualización de resultados (Hunter, 2007).

3.2.2 Simulation of Urban Mobility

Simulation of Urban MObility (SUMO) es un paquete de simulación de tráfico microscópico y continuo que da la posibilidad de diseñar y manejar grandes redes viales. Permite realizar simulaciones intermodales, es decir, incluyendo distintos tipos de entidades que se relacionan entre sí como peatones o vehículos. Es un proyecto de código abierto desarrollado principalmente por empleados del Instituto de Sistemas de Transporte del Centro Aeroespacial Alemán (Lopez et al., 2018).

Se utilizará como simulador de tráfico subyacente debido a diversos motivos. El primero es la gran cantidad de herramientas que provee (muchas de las cuales no serán utilizadas), como la visualización de las simulaciones o las herramientas de diseño de redes urbanas, entre otros. El segundo es su uso extendido entre la comunidad científica. Todos los artículos analizados han probado sus resultados usando SUMO, por lo que utilizarlo también garantizará obtener resultados en las mismas condiciones. El último motivo está relacionado con su compatibilidad con la herramienta de gestión multiagente AITrafficLab.

3.2.3 AITrafficLab

AITrafficLab (Martínez Parpolowicz, 2024) es una librería de Python especialmente diseñada para el entrenamiento de modelos basados en sistemas multiagente en redes viales urbanas. Utiliza un enfoque basado en grafos, que permite discretizar las redes de tráfico, e implementa un gestor multiagente. Con él se pueden personalizar los comportamientos de los conductores mediante la definición de una o varias clases de tipo agente.

A fecha de elaboración de este trabajo, la librería solo contempla la posibilidad de crear modelos cuyos agentes son los propios conductores de la red y no tiene en cuenta otros factores o posibles agentes como los viandantes o los semáforos. No obstante, ninguna de ellas supone un impedimento para la realización del trabajo.

4. DESARROLLO DEL TRABAJO

En el siguiente apartado se detalla el proceso que se ha seguido, de acuerdo con la planificación marcada en la sección de metodología, para llevar a cabo la investigación.

4.1 Análisis de los algoritmos existentes

Después de haber llevado a cabo un estudio preliminar sobre los diferentes algoritmos de inteligencia de enjambres se observó que la mayoría pertenecían a la familia de optimización por colonias de hormigas. A continuación, se procede entonces a la realización de un análisis más en profundidad de los algoritmos mencionados en la sección de trabajos relacionados.

El primer aspecto relevante encontrado es el esfuerzo que dedica E-PGVR a la creación de un sistema teniendo en cuenta no solo el aspecto de optimización de rutas, sino también su viabilidad de implementación en un entorno real. Como se ha comentado, el sistema jerárquico diseñado limita el número de interacciones entre agentes. Ello provoca que los agentes de este sistema no sean omniscientes (de ahora en adelante agentes ignorantes), es decir, cuenta con una perspectiva limitada sobre la situación global del sistema.

De manera similar, el algoritmo PTPR usa el algoritmo LDkSP (Ho et al., 2019) para limitar la cantidad de información que conoce cada agente evitando la pérdida de rendimiento. Esto es algo que por ejemplo no sucede en IACO donde cada agente conoce en tiempo real la actualización de las feromonas de cada una de las aristas del grafo viario; puesto que, sin dicha información no podría evaluar correctamente todos los posibles caminos mediante el algoritmo de Dijkstra.

Se puede establecer entonces que, en igualdad de condiciones y resultados obtenidos por un modelo con agentes omniscientes y uno con agentes ignorantes, el modelo con agentes ignorantes tiene una implementación real más viable. Al menos en lo que a la complejidad de infraestructura de comunicación se refiere.

Otra disyuntiva que se presenta es la elección entre un enfoque predictivo o uno reactivo. Los resultados de los algoritmos que tratan de predecir las condiciones de tráfico futuro ponen de manifiesto la imposibilidad de realizar predicciones de situaciones a largo plazo o que abarquen una gran extensión debido a problemas de escalabilidad y rendimiento. Por ello, ambos trabajos limitan sus predicciones a

entornos cercanos al vehículo. Compensan la limitación en sus predicciones presentando características de comportamiento propias de algoritmos reactivos. Así, ambos algoritmos proponen obtener predicciones iterativamente para adaptar su comportamiento localmente.

Limitar la percepción de un agente a un entorno local puede conllevar el estancamiento en óptimos locales. Para comprender este concepto en un contexto de optimización de rutas se presenta el siguiente ejemplo sobresimplificado, donde un vehículo debe salir del nodo de salida (S) y llegar al nodo de destino (D) y los pesos de las aristas representan el tráfico o estimación de tiempo debido al mismo:

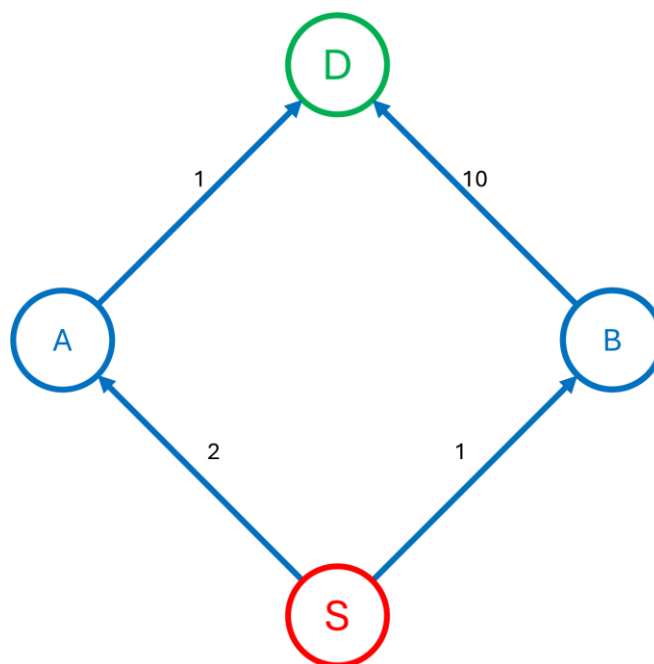


Ilustración 8. Ejemplo simple de tráfico discretizado.

En este trivial ejemplo se puede observar que, si un agente en S solo dispusiera de información a un salto de distancia, elegiría el camino al nodo B. Una vez en B estaría obligado a circular por la arista BD para alcanzar D, provocando que el tiempo total del viaje aumente (respecto a haber elegido el nodo A como primer destino).

Si bien es cierto que en un entorno urbano rara vez las decisiones tomadas son completamente irrevocables, este ejemplo puede fácilmente extrapolarse a entornos reales. Por ejemplo, se ha de atravesar una población desde un extremo hasta el opuesto, pudiendo elegir entre los dos sentidos de una carretera de circunvalación. Sin embargo, el conductor en cuestión desconoce que ha ocurrido un accidente en uno de los sentidos.

Para solventar este problema, una posible trayectoria de aproximación sería introducir el concepto de “componente social” de un algoritmo PSO. Mediante el empleo de este concepto, los algoritmos de PSO compensan el estancamiento de óptimos locales causado por la “componente cognitiva” (que extrapolado a este caso serían las predicciones a corto plazo de los agentes).

Finalmente, una ventaja que tienen los enfoques reactivos frente a los predictivos surge precisamente de un evento mencionado anteriormente: los accidentes de tráfico. Y no solo este tipo de eventos, sino, en general, cualquier suceso anómalo que pueda no entrar dentro de los parámetros de un modelo predictivo, como cortes de calles o inclemencias del tiempo, entre otros.

4.2 Elección del algoritmo base y planteamiento de modificaciones

En base a las ventajas e inconvenientes vistos en el apartado anterior sobre cada aspecto de los algoritmos relacionados se plantea modificar el algoritmo IACO. Los motivos principales de haber elegido IACO son:

1. Es un algoritmo de carácter reactivo. Este hecho reduce la cantidad de datos necesaria para llevar a cabo el diseño y entrenamiento del modelo. Esto, además, aumenta su flexibilidad, especialmente si se dan sucesos no contemplados en la fase de diseño.
2. De entre los estudiados, es el algoritmo más simple. Dado que no se dispone de ningún de modelo preentrenado de entre los estudiados, haría falta rehacer el proceso de entrenamiento, lo que podría suponer un problema teniendo en cuenta la limitación existente en cuanto a recursos computacionales. Por otro lado, proponer modificaciones podría desembocar en el aumento de la complejidad del algoritmo.
3. Utiliza información global. Es capaz de tener en cuenta el tráfico a mayor distancia que si se limitase al uso de información local.

A partir de ahora la investigación se centrará en proponer modificaciones para este algoritmo, utilizando aspectos y conocimientos inspirados en el resto de los algoritmos, en pos de los siguientes objetivos específicos:

1. Crear una interfaz de comunicación entre agentes que permita la obtención de información global de manera realista. Para esto se propone la introducción de

una “componente social”, favoreciendo la comunicación directa entre agentes y limitando la colaboración mediante estigmergia.

2. Evitar una pérdida de rendimiento significativo respecto del modelo original.

4.3 Propuesta de idea: propagación de señales de comunicación

Si bien es cierto que la estigmergia es el concepto clave que permite el funcionamiento de los algoritmos de la familia ACO, en este caso concreto se trata de un problema a resolver: Es precisamente mediante esta comunicación con el medio que las hormigas de IACO son capaces de encontrar el camino menos congestionado a su destino, sin embargo, para su correcto desempeño, las hormigas deben obtener en cada iteración información de toda la red en su conjunto interactuando con el medio (obteniendo el nivel de feromonas depositado en todas las aristas de la red urbana). En un entorno puramente teórico y mediante la ejecución de una simulación en un ordenador, esto es tan simple como obtener o modificar el valor de una variable compartida o accesible por todos los agentes. Pero, como ya se ha mencionado anteriormente, recrear este concepto a escala metropolitana resultaría más complejo.

Es por este motivo que se plantea buscar una forma de codificar la información de manera que un agente pueda reconstruir la información global con la mínima pérdida posible.

Si revisamos el caso simplificado propuesto anteriormente, para solucionar el problema que se producía por obtener solo información local bastaría con hacer el peso de las aristas dependiente del peso de otras aristas en la red:

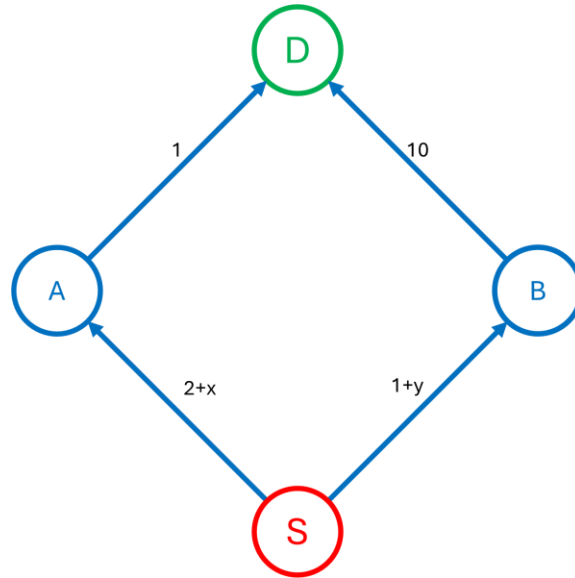


Ilustración 9. Ejemplo simple de tráfico discretizado con pesos dependientes.

En este nuevo ejemplo, según el valor que tomasen x e y , un vehículo podría evitar el atasco en \overline{BD} . La complejidad del problema reside ahora en diseñar una función que permita modelar la influencia de unas aristas sobre otras.

4.4 Estudio de un caso simplificado

Para estudiar cómo influye el nivel de tráfico de unas calles en otras, se ha optado por estudiar primero un caso considerablemente más simple que la realidad. Las reglas que rigen este nuevo escenario son las siguientes:

1. Cada arista ha sido discretizada, subdividiéndolas en múltiples aristas sus respectivos nodos intermedios. La longitud (o peso) de todas las nuevas aristas es idéntica y mayor o igual a la distancia de seguridad necesaria para evitar colisiones. Los “nodos originales” serán representados como hasta ahora. Los nuevos “nodos intermedios” tendrán siempre grado dos y serán representados más pequeños y mediante contorno discontinuo. En los casos en los que no sean necesarios o distraigan del verdadero objetivo de la visualización, se representará el conjunto de todos los n nodos intermedios de una arista mediante un único nodo más grande y con contorno discontinuo.

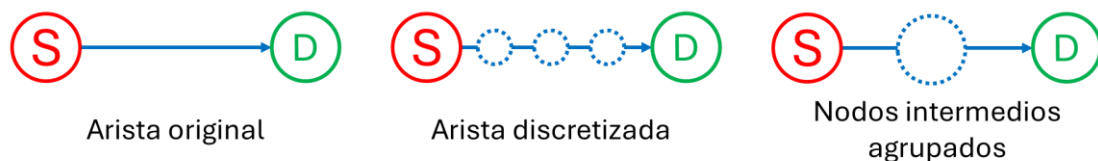
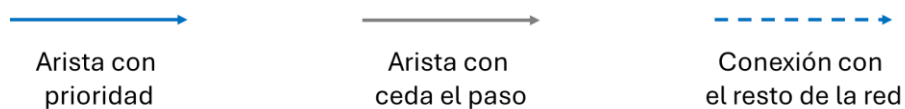


Ilustración 10. Representación visual de aristas.

2. Dado que ahora las calles no son ahora un espacio continuo, cada vehículo solo podrá existir en un nodo en cada iteración, ya sea un nodo original o un nodo intermedio. No podrá haber dos vehículos en un mismo nodo en una misma iteración.
3. En cada iteración, un vehículo solo podrá moverse a un nodo inmediatamente adyacente. Esta regla no puede contradecir la anterior, por lo que, si todos los nodos adyacentes están ocupados, el vehículo en cuestión no podrá moverse.
4. Cuando dos vehículos en nodos intermedios intenten acceder a un mismo nodo original, solo podrá acceder aquel que se encuentre en la arista con prioridad (marcadas en azul). El que se encuentre en la arista con ceda el paso (marcadas en gris) deberá por tanto esperar en su nodo actual y reintentarlo en la siguiente iteración. Las aristas discontinuas representan la conexión del subgrafo representado con el resto de la red.



tráfico, existen flujos vehiculares externos. Un vehículo ocupa el nodo A o E en la iteración t con probabilidad $P_t(A)$ y $P_t(E)$ respectivamente.

Desde S existen dos posibles caminos al nodo D:

- Camino a : $\{S, \{\text{nodos en } \overline{SB}\}, B, \{\text{nodos en } \overline{BD}\}, D\}$
- Camino b : $\{S, \{\text{nodos en } \overline{SC}\}, C, \{\text{nodos en } \overline{CD}\}, D\}$

Si un vehículo llega a A en la iteración t , dado que A y B son adyacentes, siempre llegará a B en la iteración $t+1$, puesto que la arista \overline{AB} tiene prioridad sobre la arista \overline{SB} . Por lo tanto, tomando en cuenta solo vehículos que provienen de A, la probabilidad de que B esté ocupado en la iteración $t+1$ es $P_{t+1}(B) = P_t(A)$. Para este ejemplo asumiremos también que la probabilidad de aparición de un vehículo en A no depende del tiempo y es constante. Por lo tanto, solo tomando en cuenta vehículos que provienen de A, la probabilidad de que B esté ocupado es $P(B) = P(A)$. Lo mismo pasa en los nodos C y E, aunque $P(A) \neq P(E)$.

Ahora, si un vehículo saliera de S y tomara el camino a , necesitaría, en cierta iteración t ocupar el nodo B. Solo podrá hacerlo si no va a ser ocupado por un vehículo de proveniente de la arista con prioridad. De estarlo, deberá esperar una iteración y volver a reintentarlo en $t+1$. Este ejemplo se puede visualizar en la siguiente figura, donde un triángulo verde representa el vehículo estudiado:

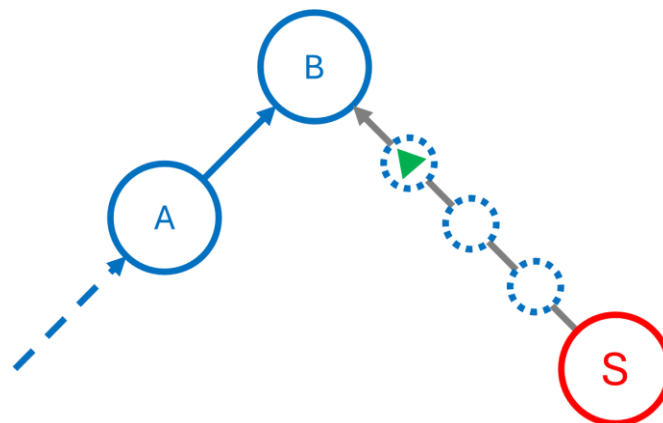


Ilustración 13. Ejemplo de incorporación discretizada.

Por lo tanto, podemos considerar cada intento de incorporación como un experimento de Bernoulli, donde la probabilidad de éxito es complementaria a la probabilidad de ocupación del nodo B por parte de un vehículo proveniente de A. Cada vehículo de S solo necesita un éxito para incorporarse a B. La cantidad de intentos necesarios hasta obtener el primer éxito puede ser modelada mediante una distribución geométrica con

probabilidad de éxito $1 - P(A)$. Así, el coste total del camino a teniendo en cuenta el tráfico se puede obtener mediante:

$$E[C'_a] = C_a + E[Geom(1 - P(A)) - 1] = C_a + \frac{1}{1 - P(A)} - 1$$

Donde C_a es el coste básico de a si no hubiera ningún otro vehículo circulando (medido en número de iteraciones), C'_a es el coste de a teniendo en cuenta el tráfico, $Geom(p)$ es una distribución geométrica de parámetro p (siendo p la probabilidad de éxito), y, $E[x]$ es la esperanza matemática. Cabe destacar que, si el éxito se produce en el primer intento, el vehículo de S espera un total de cero iteraciones; si se produce en el segundo intento, esperará una iteración y se incorporará en la segunda; y, así sucesivamente. Por lo que el coste de la espera asociada al tráfico es igual a la cantidad de fracasos (intentos totales exceptuando un éxito).

Pasando a un caso más complejo, se puede cuantificar el tiempo de espera en la arista sin prioridad debido a la acumulación de esperas de distintos vehículos. En la siguiente figura, el triángulo verde sigue representando el vehículo estudiado, mientras que los triángulos naranjas representan otros vehículos:

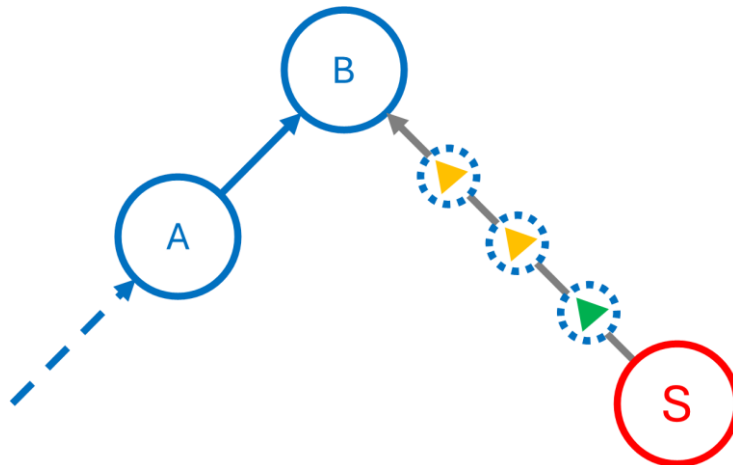


Ilustración 14. Ejemplo de incorporación discretizada con densidad vehicular máxima.

En este nuevo caso, se observa como la arista \overline{SB} tiene una densidad vehicular máxima, es decir, no podrían acceder nuevos vehículos desde S a un nodo intermedio de esta arista hasta que algún vehículo se desplazara a B . De igual manera, el vehículo verde no podrá acceder a B hasta que los dos anteriores lo hayan hecho. Necesitará, por tanto, un total de tres éxitos de incorporación para poder abandonar la arista \overline{SB} . Así, el número de intentos necesarios hasta obtener el k -ésimo éxito se puede modelar

mediante una distribución binomial negativa. Nuevamente el tiempo empleado por el vehículo verde será igual al número de intentos de incorporación fallidos, es decir, el número total de intentos menos k intentos exitosos:

$$\begin{aligned}
 E[C'_a] &= C_a + E[BN(n_{\overline{SB}}, 1 - P(A)) - n_{\overline{SB}}] = C_a + \frac{n_{\overline{SB}}}{1 - P(A)} - n_{\overline{SB}} \\
 &= C_a + \sum_{i=0}^{n_{BE}} E[Geom(1 - P(A)) - 1] = C_a + \sum_{i=0}^{n_{BE}} \left(\frac{1}{1 - P(A)} - 1 \right)
 \end{aligned}$$

donde ahora $BN(k, p)$ es una distribución binomial negativa de parámetros k (número de éxitos) y p (probabilidad de éxito), y, $n_{\overline{SB}}$ es el número de vehículos en la arista \overline{SB} (que, para este caso, tomaría el valor 3). Será útil más adelante notar que, al ser la distribución binomial negativa una extensión de la distribución geométrica para para k éxitos, también podemos expresar la cuantificación del coste esperado de a (teniendo en cuenta el tráfico) utilizando esta última distribución. Esta última igualdad puede ser interpretada como que **el coste esperado es igual al coste básico del camino más la acumulación de retrasos esperados de cada vehículo que circula delante del vehículo estudiado**.

Se propone un último escenario que no se analizará en profundidad debido a su complejidad:

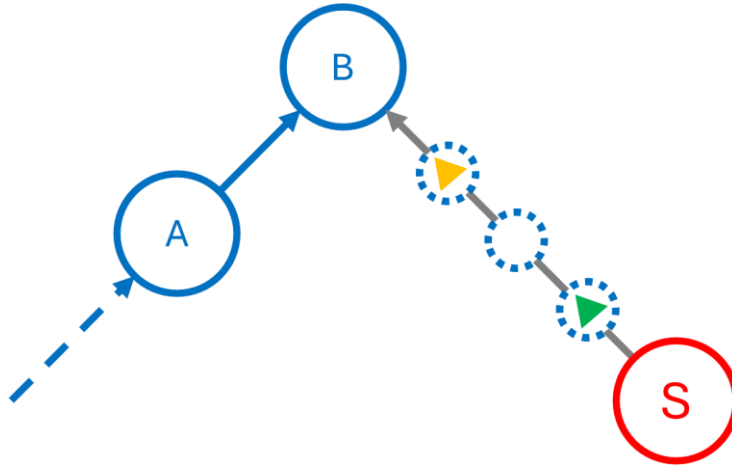


Ilustración 15. Ejemplo de incorporación discretizada con densidad vehicular no máxima.

En esta ocasión la densidad vehicular de la arista \overline{SB} no es máxima, lo que tiene consecuencias directas en el coste esperado del camino. Por un lado, es fácil ver que, independientemente de si el vehículo amarillo accede a B o no, en la siguiente iteración, el vehículo verde avanzará al siguiente nodo adyacente al suyo en el sentido de la

marcha. Por otro lado, el vehículo predecesor podría acceder a B o deberá esperar en su nodo actual.

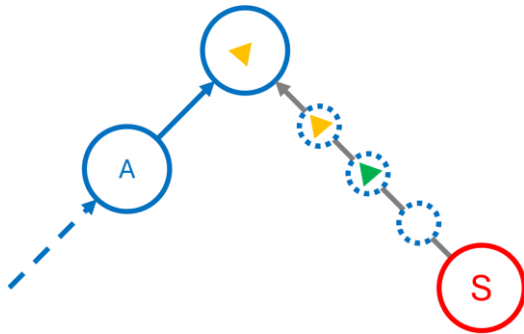


Ilustración 16. Caso 1: Otro vehículo accedió a B desde la arista con prioridad en $t+1$.

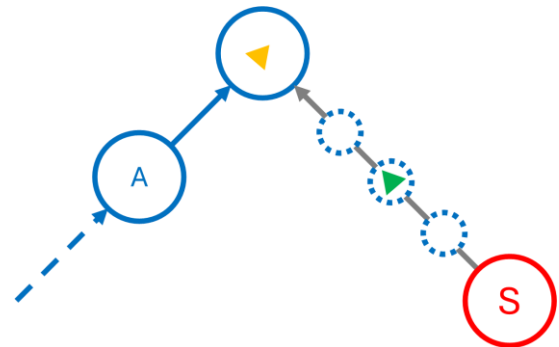


Ilustración 17. Caso 2: Nadie accedió a B desde la arista con prioridad en $t+1$.

En el caso 1, el vehículo estudiado no se verá afectado por los retrasos ocasionados por el otro vehículo, por lo que el nuevo coste actualizado de a dependerá solo de su propio retraso ocasionado por no poder acceder a B. Mientras que, en el caso 2, el vehículo estudiado deberá, además esperar las iteraciones necesarias para que su vehículo predecesor acceda a B. El nuevo coste esperado puede modelarse como:

$$E[C'_a] = C_a + P(\text{Alcance}) * C_T + P(\overline{\text{Alcance}}) * 0$$

Donde $P(\text{Alcance})$ es la probabilidad de que el coche estudiado alcance al coche predecesor y C_T es el coste total del retraso asociado a alcanzar al vehículo predecesor. Ahora, si bien es cierto que para el caso concreto planteado no sería complejo obtener esta esperanza, la generalización de esta ecuación tiene un carácter recursivo. La probabilidad de que el vehículo alcance al vehículo que circula delante suyo depende de si ha llegado a una intersección sin prioridad o si este se detiene porque alcanza al vehículo que a su vez precediera a este, lo que a su vez dependería nuevamente de las mismas condiciones. Por otro lado, el coste total variaría también en función de en qué iteración alcanzara cada vehículo a su predecesor.

La obtención de la solución general de este escenario consumiría tiempo y podría acabar no siendo relevante para el desarrollo del algoritmo. Lo que sí que es relevante, sin embargo, es haber descubierto precisamente este carácter recursivo, donde **el coste esperado de cada vehículo depende del tráfico sufrido por el vehículo inmediatamente anterior en su arista.**

Esto desemboca en otra conclusión relevante, aunque quizás trivial para cualquier tipo de conductor, el coste asociado a recorrer una arista **solo depende del nivel de tráfico delante del vehículo y no del que hubiera detrás**. Esto se debe a que esta dependencia recursiva afecta de uno en uno a los vehículos de una calle en orden de llegada a dicha arista. O, lo que es lo mismo, **en orden inverso al sentido de la marcha** (sentido de la arista).

Precisamente tratando de cuantificar el coste real, los casos básicos mencionados hasta ahora tenían en cuenta que el coste asociado a un fracaso de acceso al nodo B era de una iteración. Se podría entonces considerar esta medida como un indicador del nivel de tráfico. Así, **al multiplicar el indicador de tráfico por su coste temporal, se obtiene una cuantificación en unidades de tiempo** del coste esperado del camino:

$$E[C'_a] = C_a + C_r E[BN(n_{\overline{SB}}, 1 - P(A)) - n_{\overline{SB}}]$$

Donde ahora C_r es el coste del retraso asociado a cada fallo de incorporación a la calle con prioridad (medido en unidades temporales). Esto podría llevarse más allá, utilizando un coste distinto por vehículo e iteración para modelar los procesos de aceleración y frenado de cada uno:

$$E[C'_a] = C_a + \sum_{i=0}^{n_{BE}} C_{r_i}^t E[Geom(1 - P(A)) - 1]$$

Donde ahora $C_{r_i}^t$ es el coste del retraso del vehículo i en la iteración t .

4.5 Propagación de feromonas

El estudio simplificado del apartado anterior dista mucho de poder ser una solución completa al problema, entre otros motivos porque se hacen varias asunciones que lo convierten en un caso poco realista. No obstante, si se combinan los aspectos resaltados en su estudio con los conocimientos obtenidos sobre optimización por enjambres, puede llegar a proponerse una solución completa.

Para poder elegir entre distintos caminos, es necesario obtener el coste esperado de cada uno de ellos. En el caso de estudio anterior se llegó a la conclusión de que la información asociada al coste de recorrer cada arista puede propagarse recursivamente de vehículo en vehículo en sentido contrario al de la marcha de este. Pudiendo entonces

un vehículo elegir entre dos caminos si conociera únicamente la información del último vehículo de cada arista saliente del nodo en el que toma la decisión.

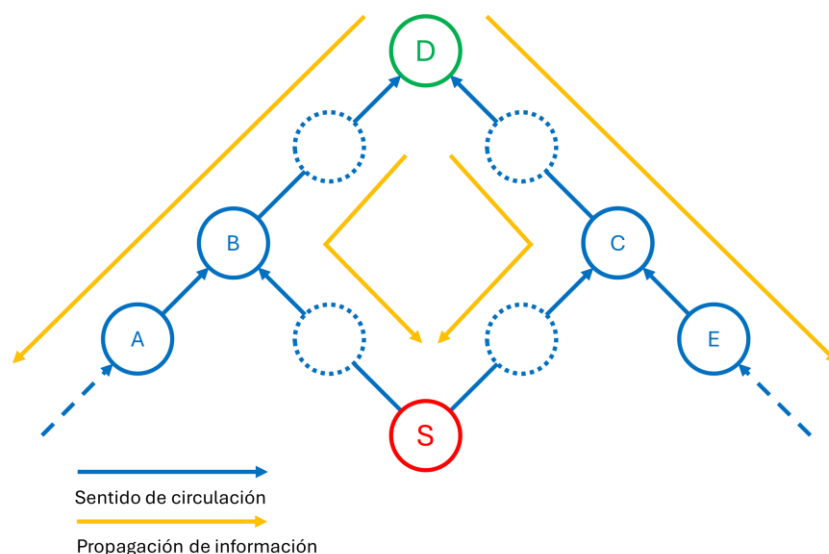


Ilustración 18. Propagación de la información en el ejemplo discretizado.

Teniendo en cuenta que en el problema real las calles son espacios continuos, no es posible estudiar la probabilidad de que un vehículo alcance a otro analizando todas las posibilidades. Por lo que, en su lugar, se ha buscado una forma alternativa basada en el mismo concepto de feromona, que pueda funcionar como indicador de tráfico.

El problema de las feromonas que se presentan tanto en IACO como en IPR es que están inspirada por un fenómeno químico, y permanece estática en cada una de las aristas. Por lo que se propone la idea principal del trabajo: **la propagación de la feromona por la red**. La idea de propagación es algo que puede entrecerarse en el algoritmo PTPR, donde las hormigas son capaces de depositar feromonas con menor intensidad en aristas más alejadas.

Este nuevo comportamiento de la feromona debería mantener sus cualidades originales, es decir, la acumulación y evaporación; a la vez que permitir influenciar el nivel de feromonas en otras aristas. Para ello se estudiaron distintos fenómenos físicos que pudieran servir de inspiración para crear esta nueva interfaz de comunicación entre los agentes:

1. Fenómeno de difusión del calor: la idea básica de este modelo sería tratar a los vehículos como fuentes de calor. Cada vehículo tendría un comportamiento similar a una partícula caliente sobre un plano con menor temperatura. Otros vehículos tratarían de encontrar la “ruta más fría” a su destino, evitando así a

otros vehículos. La principal ventaja que se encontró en este enfoque es que el calor se disipa tanto con la distancia como con el tiempo, con lo que la señal térmica emitida cumpliría con las funciones de “evaporación” y “propagación”. No obstante, sería necesario redefinir la función de acumulación o encontrar una forma de que la señal térmica fuera sumable. Es decir, ya que juntar dos cuerpos a cierta temperatura no hace que se duplique la temperatura del conjunto, sería necesario encontrar un método que permitiera acumular este calor para indicar un mayor nivel de tráfico o densidad vehicular.

2. Fenómenos de atracción y repulsión electrostática: la idea básica de este modelo sería tratar a los vehículos como partículas cargadas negativamente y sus destinos como partículas cargadas positivamente. Cada vehículo entonces se vería atraído por su destino y repelido por el resto de los vehículos. Si bien el campo eléctrico no desaparece con el tiempo, sí que lo hace con la distancia; por lo que, quizás la carga de cada partícula podría ser dependiente del tiempo. Dos vehículos adyacentes podrían además considerarse como una única partícula cargada con su suma como carga. Pese a ser prometedor, no se encontró una forma de modelar el peso de las aristas dentro de este modelo o de equilibrar las cargas positivas y negativas para que un vehículo siempre se dirigiera a su destino por mucho que se aproximara a una carga negativa.
3. Fenómeno de absorción acústica: la idea básica de este modelo es tratar a cada vehículo como una fuente que emite ruido. Así, otros vehículos tratarían de encontrar la “ruta más silenciosa” hasta su destino. Dos fenómenos permiten que una señal acústica se disipe a medida que esta se propaga por un medio: la atenuación y la absorción acústicas. Al considerarse el grafo como un conjunto de hilos (aristas) por el que se propagan las vibraciones, no existiría atenuación (puesto que se está propagando una onda unidimensional), por lo que se eligió el fenómeno de absorción.

De las tres posibilidades estudiadas inicialmente, fue necesario elegir una para desarrollar el algoritmo, puesto que las características propias de cada uno de los modelos comenzaban a ser demasiado diferentes como para crear las tres versiones, implementarlas, analizarlas y después tomar la decisión. Así, se optó por basar el nuevo comportamiento de la feromona en señales acústicas. No obstante, la creación de otros

algoritmos basados en las ideas expuestas puede ser una potencial línea de trabajo futuro.

4.6 Formulación de la propagación de feromonas sónicas

Antes de comenzar con la formulación del algoritmo vamos a elaborar una explicación más detallada de nuestra modificación. Como se ha comentado, la idea principal consiste en cambiar el comportamiento básico de las feromonas para que, en lugar de que funcionen como señales químicas, tengan características más similares a señales acústicas. Nace así el concepto de feromona sónica:

Definición 3: *Feromona sónica.*

Valor numérico que codifica la cantidad de tráfico que hay en una calle. Comparte el efecto de repulsión de las feromonas invertidas y su acumulación provoca una intensificación del efecto de repulsión.

A diferencia de las anteriores, las feromonas sónicas no reducen su intensidad mediante la “evaporación”, sino que se propagan sufriendo el efecto de absorción acústica, atenuándose a medida que se alejan del foco emisor.

A menudo la feromona sónica será también referida como **señal acústica**.

Intuitivamente se podría entender el algoritmo de la siguiente manera: cada conductor utiliza el claxon de su vehículo para avisar al resto de conductores de su presencia. De esta manera, un conductor que tuviera que tomar una decisión en una bifurcación, elegiría aquella en la que se escuchase menos ruido. Esto se debe a que una menor cantidad de ruido indicaría que o bien la congestión vehicular es menor por dicha calle, o bien que se encuentra más lejos. De ser el segundo motivo, sería posible entonces que se haya disipado el atasco a su llegada, que su destino se encuentre antes de alcanzar el atasco, o, que tenga margen de maniobra suficiente para tomar otra ruta antes de llegar a la aglomeración.

Las ideas detrás de este algoritmo están inspiradas por algunos de los conceptos que se proponen en el proyecto *Screaming insects* (Simulife Hub, 2023). Este proyecto no está relacionado con la resolución de problemas de tráfico, sino que se trata de un algoritmo planteado para encontrar la ruta más corta entre dos puntos. En él, muchos agentes “ciegos” buscan el origen y el destino de la ruta. Cuando un agente encuentra

accidentalmente uno de los dos, emite una señal perceptible en cierto radio (proporcional a la intensidad). Cuando otro agente percibe la señal, aumenta la intensidad de la señal y la reemite, aumentando el radio de alcance. Gracias a esto, todos los agentes acaban enterándose de la localización de ambos nodos mientras optimizan la ruta entre ellos. Inspirado por este modelo de comunicación surge el concepto de impulso:

Definición 4: *Impulso.*

Valor numérico utilizado para aumentar la intensidad de la feromona sónica y permitir que esta sea perceptible en un radio mayor antes de que se atenúe por completo.

Con este concepto se define una nueva lógica de funcionamiento para los agentes basada en los siguientes puntos:

1. La señal se transmitirá en dirección opuesta al sentido de la marcha y se irá atenuando a medida que se propaga.
2. Basado en el concepto de impulso propuesto, cada agente percibirá y emitirá, en cada iteración, una señal acústica variable. La intensidad de la señal será igual a la intensidad de la feromona percibida por el agente más el incremento causado por el impulso.
3. Cada agente percibirá exclusivamente la señal emitida por el agente que le precede en su arista. Después, modificará dicha señal y volverá a emitirla para que la reciba el siguiente agente.
4. Los agentes utilizarán la información codificada en las feromonas para evitar las zonas con mayor densidad vehicular.

El primer punto está relacionado con el fenómeno de absorción acústica. Es cierto que difiere en parte del fenómeno real en cuanto a que la propagación del sonido se dará solo en una dirección (aquella contraria al sentido de la marcha), pero su formulación matemática se ha mantenido igual. La ecuación que modela la intensidad percibida por un agente es la siguiente:

$$I_n = I_{n-1} * e^{-kx}$$

Donde I_n representa la intensidad percibida por el vehículo n , I_{n-1} la intensidad emitida por el vehículo $n-1$ (predecesor de n en su arista), k es el coeficiente de

absorción acústica y x es la distancia que hay entre $n-1$ y n . I es además una medida del retraso esperado, por lo que se mide en unidades de tiempo (s).

El valor del coeficiente de absorción acústica depende de las características del medio en el que se propaga la onda. En este caso, consideraremos a la propia calle como un medio, sustancia o material por el que se propaga la onda. La determinación de este valor se discute más adelante.

El segundo punto modela el impacto individual que tiene cada uno de los vehículos sobre el tráfico. Siguiendo la idea de impulso, antes de emitir de nuevo la señal, cada agente aumentará la intensidad una cantidad igual a I_0 :

$$I_n = I_{n-1} * e^{-kx} + I_0$$

La intensidad inicial I_0 puede considerarse como la cuantificación del impacto que tiene cada vehículo en el nivel de tráfico general y cómo se traduce esto en retraso para otros vehículos. Con todo esto ahora I_n es la intensidad de la señal o feromona que enviará el vehículo n después de percibir la de su predecesor. Esta señal será percibida por su sucesor, que volverá a modificar la señal antes de enviarla, y así sucesivamente. Tenemos entonces reconstruida una función recursiva que indica a cada vehículo el coste temporal adicional de una arista ocasionado por el tráfico. Sin embargo, aún es necesario conocer la definición de la intensidad inicial. Para ello es necesario definir la formulación de impulso:

$$\delta * e^{-k_a x_{seg a}} = \epsilon$$

$$\delta = \epsilon * e^{k_a x_{seg a}}$$

En la ecuación anterior, δ es aquello a lo que llamaremos impulso, $x_{seg a}$ representa la distancia mínima de seguridad que se debería guardar en la arista a si la velocidad fuera la máxima permitida, ϵ es un valor que llamaremos umbral perceptible y k_a el valor del coeficiente de absorción en la arista a .

Definición 5: *Umbral perceptible.*

Intensidad de feromona mínima que los agentes son capaces de percibir. Su valor es próximo a cero.

Cuando una feromona se propaga suficiente como para que su intensidad alcance el umbral perceptible, se considerará que se ha “disipado” por completo, pues ningún vehículo será capaz de escucharla.

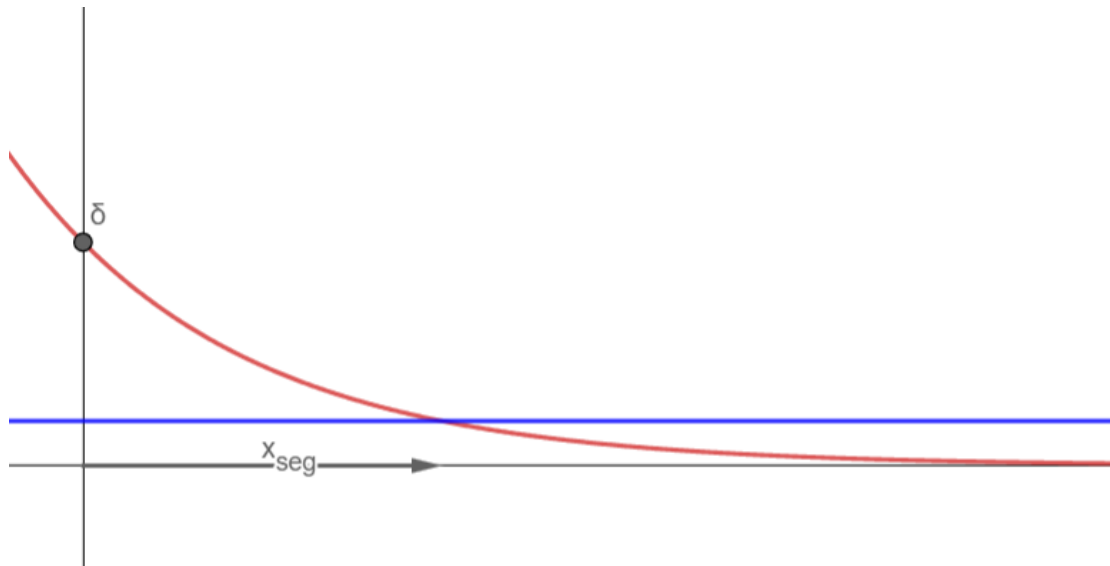


Ilustración 19. Función de atenuación y umbral perceptible.

En la gráfica anterior se pueden observar todos los elementos antes mencionados. La curva roja es la función de absorción acústica para determinado k . La recta azul es la constante $y = \epsilon$. Se muestran también la distancia de seguridad y el valor impulso en gris. Como se puede observar, un impulso emitido en $x = 0$ alcanza el umbral perceptible al cabo de x_{seg} distancia. Consideraremos que la feromona emitida con intensidad igual a un impulso se ha disipado y no ejerce influencia sobre otros vehículos.

Así, como se comentó en el estudio de la distancia de seguridad, si un agente emitiera solo un impulso, su área de influencia se limitaría a precisamente a esta distancia. Cualquier vehículo que estuviera invadiendo este espacio, tendría, necesariamente, que estar circulando a una velocidad inferior a la máxima de la vía para evitar colisiones. Esta reducción de velocidad acabará creando atascos si suficientes vehículos se ven obligados a adoptarla. En ese caso, utilizando las primeras ecuaciones vistas, la feromona acústica comenzaría a acumularse y cada vehículo no solo propagaría el impulso, sino una mayor cantidad de feromona.

No obstante, las hormigas no incrementan la señal solo usando un impulso. Esto se debe a que el retraso generado por un vehículo que circula a una velocidad de 10 km/h no es el mismo que genera uno que circula a 30 km/h aunque ambos estén circulando a velocidades inferiores a la máxima permitida. Por otro lado, el impulso es una medida adimensional, por lo que, al tener la ecuación de absorción carácter recursivo, la intensidad I también sería adimensional. Como se ha mencionado la intensidad es una

cuantificación del retraso esperado (medido en segundos), por lo que I_0 también debe estar en segundos. Por este motivo se define I_0 como:

$$I_0 = C_r * \delta$$

$$C_r = \frac{x_a - x}{v} - \frac{x_a - x}{v_{max, a}}$$

Donde C_r es la estimación del retraso que sufriría el vehículo si circula a su velocidad actual hasta que llegue al final de la arista, x_a es la longitud total de la arista a (en m), x es la distancia recorrida por el vehículo en a , v es la velocidad actual del vehículo (en m/s) y $v_{max, a}$ es la velocidad máxima permitida en a . El resultado es que C_r mide el retraso que sufre el vehículo debido exclusivamente al tráfico y no al tiempo básico que le llevaría desplazarse por la arista. Con todo esto tanto C_r como I_0 tienen unidades temporales.

4.7 Modelado de la difusión de la señal en intersecciones

Con lo visto hasta ahora el algoritmo está definido para todos aquellos casos en los que cada agente tiene un único predecesor y sucesor. Pero este no es siempre el caso, pues surgen ambigüedades en todos aquellos nodos con grado mayor que uno tanto de salida como, de entrada. Así, en un entorno en el que las aristas son espacios continuos y los nodos puntos discretos, se presentan las siguientes preguntas:

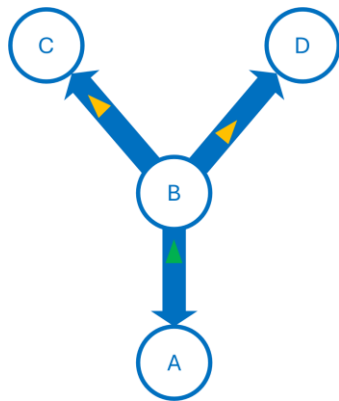


Ilustración 22. Nodo con grado de salida mayor que uno. ¿Quién es el predecesor del vehículo verde?

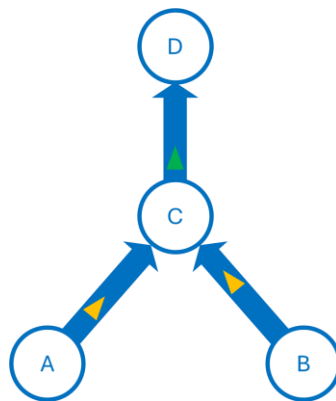


Ilustración 21. Nodo con grado de entrada mayor que uno. ¿Quién es el sucesor del vehículo verde?

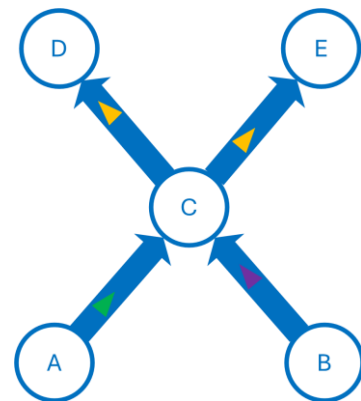


Ilustración 20. Nodo con grados de entrada y salida mayor que uno. ¿En qué se diferencia el tráfico que percibe el vehículo verde del que percibe el morado?

Ante la dificultad para responder a estas preguntas, hace falta definir el comportamiento de la feromona en las intersecciones. Por lo que surge un nuevo tipo de agente además de las hormigas o vehículos: las intersecciones o nodos inteligentes. El algoritmo empleado por estas debería, idealmente, ser genérico e independiente del grado de entrada y salida de cada nodo, puesto que existen diversos tipos de intersecciones.

Diversas aproximaciones fueron estudiadas para solucionar este problema. Las más simples se basaban en utilizar funciones como la media aritmética, la mediana, el máximo o el mínimo de las feromonas de las aristas salientes de un nodo, y, mandar una copia de la señal por cada arista entrante. Sin embargo, todas ellas compartían un mismo problema: dificultad de gestión en casos en los que la cantidad de vehículos que entran o salen por cierta arista de un nodo está desbalanceada respecto al resto de aristas de ese mismo nodo.

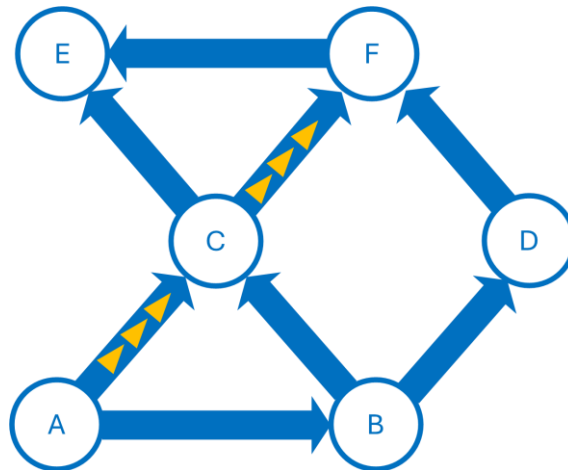


Ilustración 23. Caso de estudio con densidades vehiculares desbalanceadas

En este ejemplo, muchos vehículos comienzan su viaje en A y acaban en F. En general todos estos métodos fallan a la hora de informar a un vehículo de que la ruta {B, C, E} está poco congestionada (salvo el método de feromonas mínimas, que presenta otro problema). Suponiendo que el coste temporal básico de todas las aristas es el mismo, un vehículo en B que debe decidir su ruta hasta E, debería tomar el camino {B, C, E}. Sin embargo, si el nodo C tomara la media, la mediana o el máximo de las feromonas que percibe por sus aristas salientes, mandaría por la arista \overline{BC} una señal que indica la existencia de tráfico. El vehículo tomaría entonces equivocadamente la ruta {B, D, F, E} en su lugar. De manera similar ocurriría si otro vehículo quisiera ir de A a F y C

mandara el mínimo de feromona que recibe por sus aristas salientes, es decir 0, ya que el nuevo vehículo volvería a escoger la ruta congestionada {A, C, F}.

Se contemplaron también otros métodos más complejos, tratando por ejemplo de replicar el comportamiento de una onda cuando cambia de medio (buscando inspiración en los fenómenos de reflexión y refracción). No obstante, presentaban el mismo tipo de problemas, puesto que las leyes físicas que rigen su comportamiento también son constantes.

Es por ello por lo que se buscó una solución especialmente diseñada para el contexto dinámico del tráfico, de tal manera que el comportamiento pudiera variar para ajustarse a las condiciones de tráfico concretas de cada intersección. Esto no entra en conflicto con lo mencionado anteriormente porque se pretende utilizar una función cuyo comportamiento sea independiente del grado del nodo, pero que se adapte a las condiciones de tráfico. O, al menos, en la medida de lo posible.

Por esto se llegó a la conclusión de que lo mejor era utilizar un método frecuentista que permitiera utilizar datos históricos para ofrecer la mejor aproximación posible. Así, cada vez que un vehículo atraviesa una intersección, se guarda un registro de la arista incidente por la que entró al nodo y de la arista saliente que tomó a continuación. Esta información se almacena en una matriz de flujos F con dimensiones (n, m) donde n es el grado de entrada del nodo y m el grado de salida. De esta manera, cada elemento f_{ij} representa el número de viajes que se tomaron desde la arista incidente i hacia la arista saliente j .

Utilizando la matriz F cada nodo emite una señal acústica por cada arista i por medio de la siguiente media ponderada:

$$I_i = \frac{\sum_{j=0}^m I_j f_{ij}}{\sum_{k=0}^n \sum_{j=0}^m f_{kj}}$$

Donde I_i es la cantidad de feromona emitida por la arista incidente i e I_j es la cantidad de feromona percibida por la arista j .

Un factor a tener en cuenta es que no se ha estudiado el comportamiento a largo plazo de esta matriz. A corto plazo se adapta bien a los cambios dinámicos de tráfico, pero es posible que cuando añadir un nuevo viaje apenas modifique la media ponderada, este

método pueda perder efectividad. Este estudio y la posibilidad de almacenar valores hasta cierto horizonte temporal se deja por tanto como una línea de trabajo futuro.

4.8 El coeficiente de absorción acústica

El último parámetro que queda por analizar es el coeficiente de absorción acústica k . El valor de k determinará lo rápido que se atenúa la señal acústica a medida que se aleja del agente que la emitió. Como se puede comprobar en la siguiente gráfica, en la que se analiza la atenuación de la intensidad a medida que aumenta la intensidad con distintos valores del coeficiente, cuando mayor es su valor, más rápido alcanzaría el umbral perceptible:

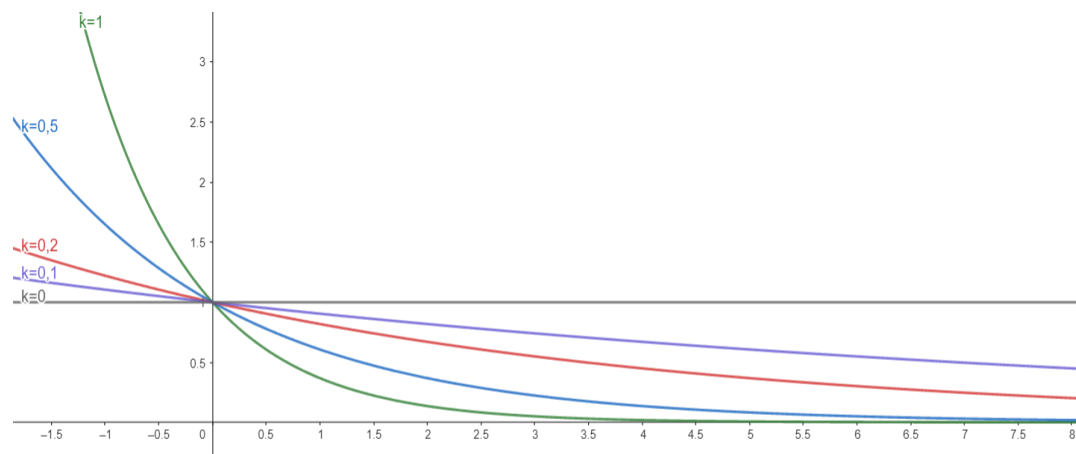


Ilustración 24. Comparación de la función de atenuación con distintos valores de k .

Determinar un valor óptimo para este parámetro es crucial para el correcto desempeño del algoritmo. Elegir un valor muy elevado favorecería la rápida disipación de las feromonas emitidas, ello provocaría que los agentes tuvieran dificultad a la hora de percibir los atascos, puesto que, aunque la feromona pudiera acumularse con cada vehículo, se disiparía rápidamente en cuanto se propagara por una calle vacía.

Lo contrario también es problemático. Un valor muy bajo del parámetro provocaría que las señales tardasen más en desaparecer. Esto tiene tres consecuencias apreciables:

1. Si el nivel general de feromonas en la red se acumula más rápido de lo que se disipa, los agentes comenzarían a alejarse de los atascos, independientemente de si ello los aleja también de sus destinos. Una consecuencia derivada de esta,

que se ha observado empíricamente, es la decisión de algunos agentes de circular en bucles cerrados con poco tráfico debido a que los niveles de feromonas fuera de ese bucle sugieren en cada iteración que la mejor opción es esperar dentro del bucle en busca de una ruta mejor.

2. Si el medio es incapaz de absorber las feromonas y estas se siguen propagando por la red, se observan fenómenos de “atasco fantasma”. Esto se produce cuando a cierta altura de una calle se produce un atasco, momento en el que los agentes comienzan a evitar dicha arista. Sin embargo, varias iteraciones después de que se haya disipado el atasco, áreas que conectan con la zona que estaba atascada dejan de ser atractivas para los agentes, aunque no estén atascadas.

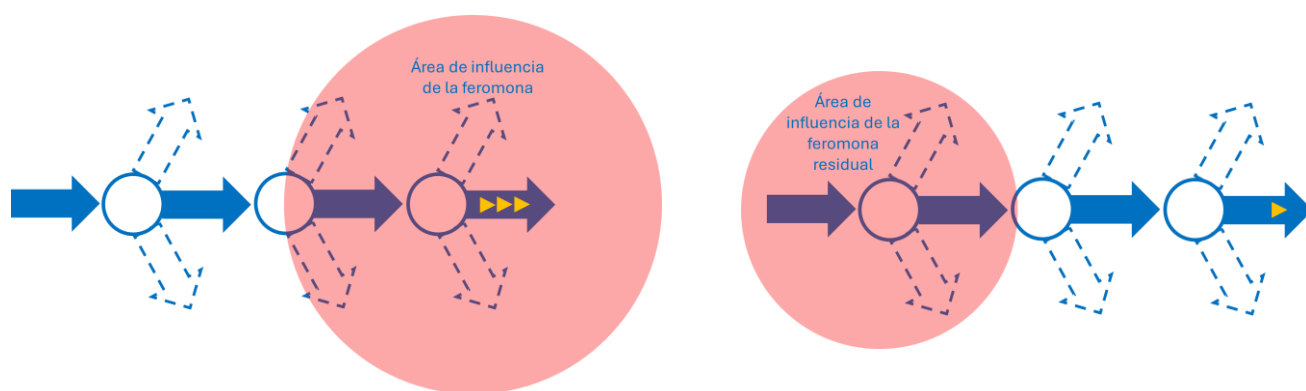


Ilustración 25. Efecto de la feromona residual.

3. En redes altamente conectadas, el aumento del nivel de feromonas comienza a concatenarse, provocando que señales que se bifurcaron en cierta intersección no se disipen y acaben juntándose de nuevo en otra, favoreciendo nuevamente el aumento del ruido generalizado en la red. Esto acaba confundiendo a los agentes, que tardan más en llegar a sus destinos, emitiendo mayor cantidad de feromonas en el proceso, que favorecen la confusión generalizada.

Para determinar un valor óptimo del coeficiente, se intentó también buscar una función inspirada en métodos de determinación del coeficiente para materiales reales. Sin embargo, al tratarse de un concepto muy relacionado con ámbitos como la arquitectura o el interiorismo (habitualmente estudiado para minimizar el ruido en los espacios como restaurantes o mejorar la acústica de otros como auditorios), es común que este se determine de forma empírica en cámaras reverberantes de acuerdo con lo especificado en la norma ISO 354:2003.

Por ello se trató de estudiar las vías urbanas como un material o sustancia sintética con sus propias propiedades únicas: número de carriles, velocidad máxima, longitud, inclinación, curvatura, su centralidad en la red, la oferta y demanda, etc. El objetivo de este estudio era desarrollar una función real de variable vectorial que permitiera obtener analíticamente el valor del coeficiente de absorción para cada calle. Inicialmente se concibió utilizar distintos coeficientes bajo la hipótesis de que las características intrínsecas de cada vía hacen que sean más o menos propensas a generar atascos.

Sin embargo, no fue posible llegar a una fórmula analítica debido a que se desconoce el impacto marginal o conjunto de cada uno de estos factores (si es que son los únicos) sobre el tráfico. Es decir, intuitivamente cualquier conductor podría entender que una calle con pendiente positiva es más probable que se atasque antes, o que un atasco se prolongue más en el tiempo, que una calle plana en las mismas condiciones debido a que toma un tiempo ligeramente superior iniciar la marcha cuesta arriba. El problema está en que no existen cuantificaciones matemáticas de dicho impacto sobre el retraso medio generado.

Por si fuera poco, existen otras variables de las que se desconoce tan siquiera si influyen de manera proporcional o inversamente proporcional en el tráfico. Un ejemplo de esto podría ser el número de carriles. A priori puede parecer que aumentar el número de carriles favorece la circulación de más vehículos y evita la congestión. Sin embargo, otras ideas igual de intuitivas ponen en duda esta afirmación: los desplazamientos laterales de los vehículos en la vía también ocasionan tráfico, pues un vehículo que se desplace lateralmente cuatro carriles en poco tiempo, por falta de previsión, puede provocar que otros tengan que frenar. Por otro lado, también se crean cuellos de botella cuando una vía reduce su número de carriles, lo que también puede ocasionar congestiones.

Todo esto, sin tener en cuenta la correlación que existe entre estas variables o cómo agregar información en distintas unidades como longitud, velocidad inclinación y número de carriles, que añade mayor complejidad al problema. Es por esto por lo que se asumió que todas estas variables guardaban una relación lineal, cuyos coeficientes serían hiperparámetros del modelo que habría que ajustar de manera empírica al igual que los coeficientes de evaporación y deposición en IACO. Aunque de este ajuste se hablará más adelante.

4.9 Formulación completa del algoritmo

Llegado a este punto, suficientes conceptos han sido explicados para entender el algoritmo en su conjunto. A continuación, se ensamblan cada una de las partes comentadas en un único pseudocódigo que permitiría implementar el algoritmo:

Algoritmo 4. Sonic Inverted Ant Colony Optimization (SIACO)

```
Inicializar hormigas (tipo de agente)
Inicializar nodos (tipo de agente)
Inicializar aristas

Para cada nodo:
    Inicializar matriz de flujos
Fin Para
Para cada arista:
    Calcular k
    Calcular impulso
Fin Para

Mientras no se cumpla el criterio de parada:
    Añadir nuevas hormigas
    Para cada nodo:
        Para cada arista_saliente en nodo:
            Percibir feromona del último agente
            Almacenar feromona percibida para arista_saliente
        Fin Para
        Calcular feromonas a emitir
        Para cada arista_entrante en nodo:
            Emitir feromona
        Fin Para
    Fin Para

    Para cada arista:
        Para cada hormiga en arista:
            Percibir feromona del agente predecesor
            Si hormiga va a cambiar de arista:
                Recalcular ruta
                Actualizar matriz de flujos del nodo que atraviesa
            Fin Si
            Emitir feromona
        Fin Para
    Fin Para
Fin Mientras
```

Al igual que IACO, el algoritmo en el que se basa el desarrollado, el algoritmo SIACO también necesita recalcular la ruta de cada hormiga cuando esta vaya a tomar una decisión. El momento de toma de decisión se produce solo justo antes de llegar a un nodo por falta de utilidad en cualquier otro caso. En este trabajo también se utilizará el algoritmo de Dijkstra para determinar la ruta más eficiente al destino. Dado que las feromonas también se miden en unidades temporales, esta es directamente sumable con el peso de cada una de las aristas:

$$E[C_a'] = C_a + I_a$$

Donde C_a' vuelve a ser el coste de la arista a teniendo en cuenta el tráfico, C_a el coste básico de a (el tiempo necesario para recorrerlas a la velocidad máxima permitida), e, I_a es la intensidad de feromona de la arista saliente a percibida por el nodo en el que se toma la decisión. De esta manera, desde la perspectiva de un agente, todos los pesos de la red se mantienen igual salvo los de las aristas salientes del nodo. Con esto se busca agregar la información de toda la red en dichas aristas para tomar la decisión con la menor cantidad de información posible.

4.10 Entorno de simulación y refinamiento de hiperparámetros del modelo

Una vez implementada la lógica del modelo, es necesario encontrar el valor óptimo de cada uno de los hiperparámetros o coeficientes que se han ido mencionando a lo largo del desarrollo. Para ello se llevará a cabo una exploración empírica del espacio de hiperparámetros por medio de la librería de Python Optuna. Esta se encargará de proponer múltiples combinaciones de valores para estos coeficientes, seleccionar los mejores candidatos y continuar explorando en torno a las áreas más prometedoras. El proceso de optimización se llevará a cabo tratando de minimizar el tiempo medio de llegada de los vehículos a sus destinos.

Con tal fin se ha creado un entorno de simulación en el que se puedan llevar a cabo los ensayos. Cabe destacar que, debido al alto coste computacional asociado a gestionar grandes redes y suficientes vehículos para que estas se congestionen, se ha acabado optando por la utilización de una red significativamente más simple. Esto permitirá conducir numerosos experimentos, obtener sus resultados y analizarlos a un bajo coste.

Así, se ha decidido emplear una red vial sintética y genérica con forma de cuadrícula de tamaño 5x5:

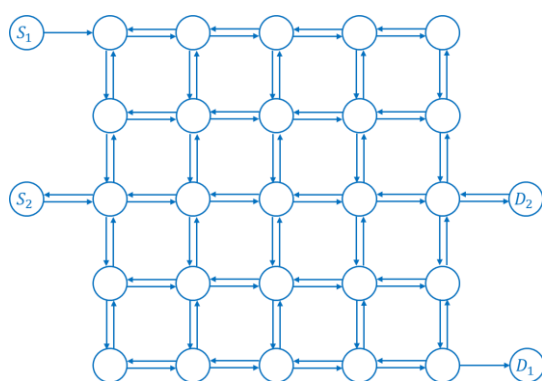


Ilustración 27. Digrafo asociado a la red creada.

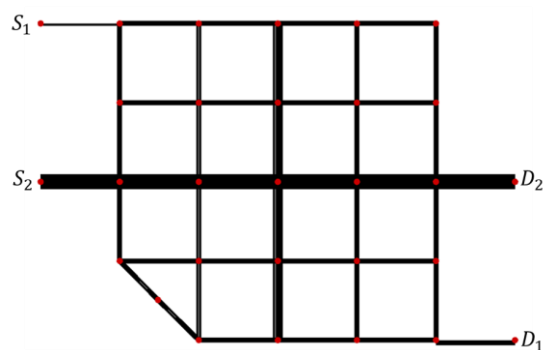


Ilustración 26. Red sintética creada usando la herramienta Netedit del paquete de SUMO.

En la anterior red, los nodos o intersecciones están representados mediante círculos rojos. Las aristas o calles están representadas mediante líneas negras. Su grosor indica la cantidad de carriles de cada una. En general todas las calles tienen un carril en cada sentido, salvo las dos avenidas principales notablemente más gruesas. La avenida horizontal tiene tres carriles de circulación en cada sentido. La avenida vertical tiene dos en cada uno. La otra excepción son las aristas conectadas con S_1 y D_1 , que tienen un único carril de circulación, como se puede ver en el grafo asociado. Cabe destacar que la esquina inferior izquierda de la red ha sido modificada para reducir la longitud de las aristas de dicho nodo. De este modo se crea una única ruta más corta entre S_1 y D_1 .

Al respecto de estos nodos nombrados, se han diseñado dos flujos de tráfico constantes. Un primer flujo de vehículos, y más relevante para ser estudiado, tiene como origen el nodo S_1 y destino el nodo D_1 . Con el mero pretexto de dificultar la libre circulación de los vehículos del flujo anterior, se ha creado un segundo flujo que circula por la avenida horizontal principal entre S_2 y D_2 .

En cuanto a la generación de vehículos, en cada iteración un nuevo vehículo comienza su viaje con una probabilidad del 50%. A su vez, este vehículo pertenece al primer flujo o al segundo con una probabilidad del 50% para cada uno. Utilizar un enfoque probabilístico en lugar de determinístico favorece que se creen situaciones con tráfico heterogéneo y evita el sobreajuste de los modelos a una situación concreta constante. Para que se creen suficientes situaciones diferentes aleatorias se ejecutará cada simulación un total de 1000 iteraciones.

Dado el carácter aleatorio de una sola simulación, la calidad del resultado obtenido después de una sola simulación podría deberse a la casualidad. Es por este motivo que con cada combinación de hiperparámetros se llevarán a cabo varios ensayos y se determinara la calidad de la combinación obteniendo el valor promedio de todas ellas. Como a priori se desconoce la distribución que siguen los resultados de los experimentos con cada combinación, se llevará a cabo un total de 30 simulaciones por cada combinación. Con todo esto, se realizaron un total de 200 ejecuciones para obtener la mejor combinación.

A lo largo del estudio se llegó a la conclusión de que la asunción hecha sobre la relación lineal guardada entre las características de una vía era poco realista. Las características que se emplearon para la obtención del coeficiente de absorción acústica fueron: el número de carriles, la velocidad máxima permitida, la longitud y la centralidad de intermediación de los nodos sobre los que incidía la arista. Se eligieron estos porque eran los únicos parámetros disponibles con los que permitía interactuar el software de simulación. Para evitar problemas con las unidades, todos se normalizaron para que se encontraran en un rango $[0,1]$. Cada coeficiente oscila entonces en un rango $[-1,1]$.

Los resultados acabaron sobreajustados al entorno propuesto, llegando a coeficientes de absorción negativos en algunas aristas que alteraban el comportamiento de los agentes. Para alcanzar este diagnóstico fue suficiente con hacer pruebas en otras redes similares, como la que se muestra en la ilustración 30, y comprobar que el tiempo de viaje promedio aumentaba considerablemente.

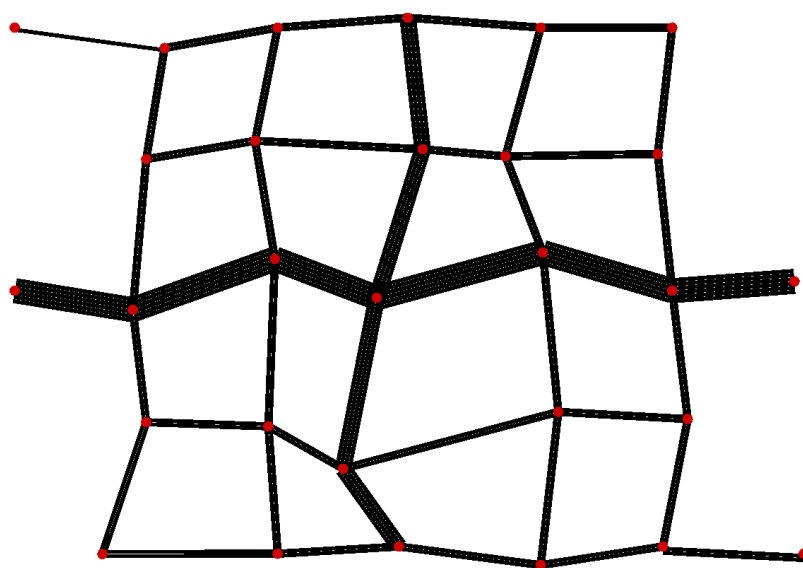


Ilustración 28. Red irregular creada a partir de modificaciones aleatorias en la red sintética de la ilustración 29.

La red anterior mantiene la misma topología que la red de entrenamiento, pero modifica la longitud de las calles y su velocidad máxima. La distancia en línea recta entre los nodos de origen y destino es la misma que la red de entrenamiento. En esta red, usar los mejores parámetros del entrenamiento ofrece resultados pésimos. Experimentos similares se llevaron a cabo en otras redes que mantuvieran las longitudes y velocidades, pero modificaran la topología de la red o el número de carriles. Las conclusiones fueron equivalentes.

Por ello, se descartó la idea de obtener un coeficiente de absorción característico de cada vía para este trabajo. No obstante, encontrar una buena relación entre los valores de las características de la vía constituye una interesante línea de trabajo futura. Así, se repitió la investigación realizando 200 nuevas ejecuciones con sus 30 ensayos cada una optimizando un único valor de k para todas las vías. Los mejores candidatos para este parámetro se encuentran todos alrededor de 0,2 (siendo 0.1895 el mejor encontrado).

Esta investigación reveló otro parámetro al que es sensible el modelo: el umbral perceptible, por lo que se incorporó a los experimentos para ajustarlo en un rango (0,1]. Los mejores candidatos para este parámetro también se encuentran próximos a 0,2 (siendo 0.2370 el mejor encontrado).

4.11 Análisis de los resultados

Ahora que el algoritmo está definido por completo, es necesario evaluar su desempeño. Llevar a cabo esta evaluación atendiendo únicamente a los resultados del algoritmo SIACO puede ser algo complicado de medir de manera absoluta. Es por ello por lo que se equiparán los resultados a los obtenidos por otros algoritmos. Para poder evaluar en igualdad de condiciones, en lugar de reutilizar los resultados de cada uno de los artículos en los que se describen, se ejecutarán nuevas simulaciones en el mismo entorno definido en el apartado anterior.

El motivo más relevante por el que se conduce este nuevo experimento es comprobar que el rendimiento del algoritmo propuesto no difiere significativamente de los rendimientos de los algoritmos del estado del arte. De ser así, al ser SIACO un algoritmo presumiblemente más realista de implementar, podríamos concluir que las modificaciones propuestas suponen un avance. De lo contrario, lo propio es cuantificar la pérdida de rendimiento y llevar a cabo un segundo estudio, relacionado con el

impacto económico de implementar cada uno de los algoritmos a gran escala, y decidir si esta reducción de rendimiento compensa el ahorro de coste.

Los algoritmos del estado del arte contra los que se va a comparar el algoritmo desarrollado son IACO e IPR, debido principalmente a que tienen una naturaleza similar y a que es considerablemente más sencillo reproducir sus resultados que los del resto de algoritmos (por lo motivos ya comentados en apartados anteriores). Cabe destacar aun así que debido a que no están completamente especificados en sus respectivos artículos, nos hemos visto en la necesidad de completar las partes faltantes. Para ello se ha hecho uso de los conocimientos adquiridos hasta el momento sobre el desarrollo de algoritmos de inteligencia de enjambres:

- Algoritmo IACO: En el artículo original, la función concreta de deposición de feromona no queda especificada. En su lugar la deposición y acumulación de feromona en una arista se define mediante la siguiente ecuación:

$$pher_{edge} = pher_{edge} + DEP_PHER$$

Que podría indicar que se trata de una deposición constante, en la que, en cada iteración cada vehículo añade una cantidad de feromona igual a DEP_PHER. Sin embargo, su valor concreto no queda especificado. No obstante, esta asunción contradice otra sección del artículo en la que se menciona que “El aumento del nivel de feromonas es, por tanto, proporcional al número de vehículos presentes en la arista, pero inversamente proporcional a su velocidad” (Dias et al., 2014). En el artículo se mencionan también la “ratio de deposición de feromonas” y la “ratio de evaporación”, que son establecidos de manera empírica (Dias et al., 2014), aunque no se especifica su valor.

Teniendo todo esto en cuenta se ha completado la falta de información con la siguiente función de deposición de feromonas:

$$DEP_PHER = \alpha \frac{x_a - x}{v}$$

Donde α es la ratio de deposición, x_a vuelve a ser la longitud de la arista a , x longitud de a recorrida por el vehículo en cuestión (por lo que el término $x_a - x$ es la distancia de a que falta por recorrer), y, v es la velocidad actual del vehículo.

Los valores concretos de las ratios de deposición y evaporación se han determinado empíricamente siguiendo el mismo procedimiento de optimización de hiperparámetros que en SIACO. Los valores óptimos para ambas variables en este contexto son ambos aproximadamente 0,5.

- Algoritmo IPR: en el artículo original se menciona el término arista permitida “*allowed edge*”, que aparece en la función de probabilidad de elegir cada una de las aristas (Nguyen & Jung, 2021), posiblemente también limitadas a este conjunto de aristas permitidas. Se define el término $allowed_i^v$ como “los segmentos de carretera permitidos para viajar hacia el destino del vehículo v desde el nodo i ”. La clave de esta definición está en la palabra “hacia”. El criterio para decidir si una arista apunta hacia el destino o no, no queda especificado.

Esto es problemático porque este algoritmo no se sirve de otros como hace IACO con el algoritmo de Dijkstra para acercarse a su destino, pues elige su camino arista a arista. Si se considera que el conjunto *allowed* lo constituyen las aristas que salen de cada nodo, cada vehículo vagaría sin rumbo tomando los caminos con menos tráfico. Casi se podría establecer una analogía entre su comportamiento y el del robot SPD-13 o “*Speedy*” en *Yo, Robot* (Asimov, 1950), puesto que se ha observado que cuanto más se acercan los vehículos al destino, más se repelen entre ellos por la acción de la feromona y más se alejan. Pero, cuando se alejan, vuelven a sentirse repelidos en el otro extremo de la red y vuelven a aproximarse al destino. Esto se produce porque, al igual que en la ficción, entran en conflicto sus dos instrucciones básicas “acércate al destino” y “evita a otros coches”.

La solución propuesta también es análoga, se aumenta la importancia de la instrucción “acércate al destino”. Dicha solución por supuesto fue propuesta por Nguyen & Jung (2021) con el conjunto *allowed*, por lo que ahora solo se definirá una posible regla para delimitar el conjunto. Con esta regla se consigue limitar la cantidad de caminos que un vehículo podría elegir para evitar que se aleje de su destino. El conjunto se obtendrá entonces mediante el siguiente algoritmo:

1. Se obtendrán k posibles caminos desde el nodo en el que se toma la decisión (S) hasta el destino (D). Para ello se empleará el algoritmo de los K caminos

más cortos de Yen (o algoritmo de Yen) (Yen, 1971). El hiperparámetro k se ha obtenido empíricamente siguiendo el mismo proceso que en SIACO, siendo su valor óptimo 20:

$$[\overline{SA}, \overline{AB}, \dots, \overline{XD}]$$

$$[\overline{SA}, \overline{AC}, \dots, \overline{YD}]$$

$$[\overline{SE}, \overline{EF}, \dots, \overline{ZD}]$$

⋮

$$[\overline{SG}, \overline{GH}, \dots, \overline{ZD}]$$

2. Se obtendrá la primera arista de cada camino y se añadirá al conjunto *allowed*. Por supuesto las aristas repetidas son ignoradas. Ahora desde el nodo S solo se pueden tomar las siguientes aristas permitidas:

$$allowed = \{\overline{SA}, \overline{SE}, \dots, \overline{SG}\}$$

Adicionalmente a los algoritmos del estado del arte, se han hecho comparaciones también con algoritmos más básicos y que bien podrían modelar el comportamiento real de algunos conductores:

- Algoritmo de Dijkstra: Cada vehículo toma la ruta más corta hasta su destino. En el caso de estudio concreto se forman grandes atascos puesto que la ruta más corta es conocida por todos los conductores y, por ende, acaba congestionada.
- Algoritmo de Yen: Cada vehículo toma una ruta aleatoria de entre las 20 más cortas. A diferencia de la modificación propuesta para IPR, esta decisión se toma una vez al comienzo del viaje. Este algoritmo busca modelar las decisiones basadas en criterios subjetivos por parte de los conductores. Pensamientos como “yo creo que por aquí es más corto”, “esta ruta es un atajo” o “seguro que el resto no ha pensado en venir por aquí” pueden llevar a los conductores a elegir sus rutas de manera algo impredecible. Sin embargo, todas ellas tienen en común que buscan de una forma u otra acercarse a su destino.

Así, se ha llevado a cabo una comparación entre los cinco algoritmos mencionados (Dijkstra, Yen, IACO, IPR y SIACO) ejecutando 50 simulaciones de cada uno en las mismas condiciones descritas en el apartado de optimización de hiperparámetros. De cada simulación se ha obtenido el tiempo de viaje medio, medido en número de

iteraciones (siendo cada iteración equivalente a un segundo real) y se ha analizado su distribución:

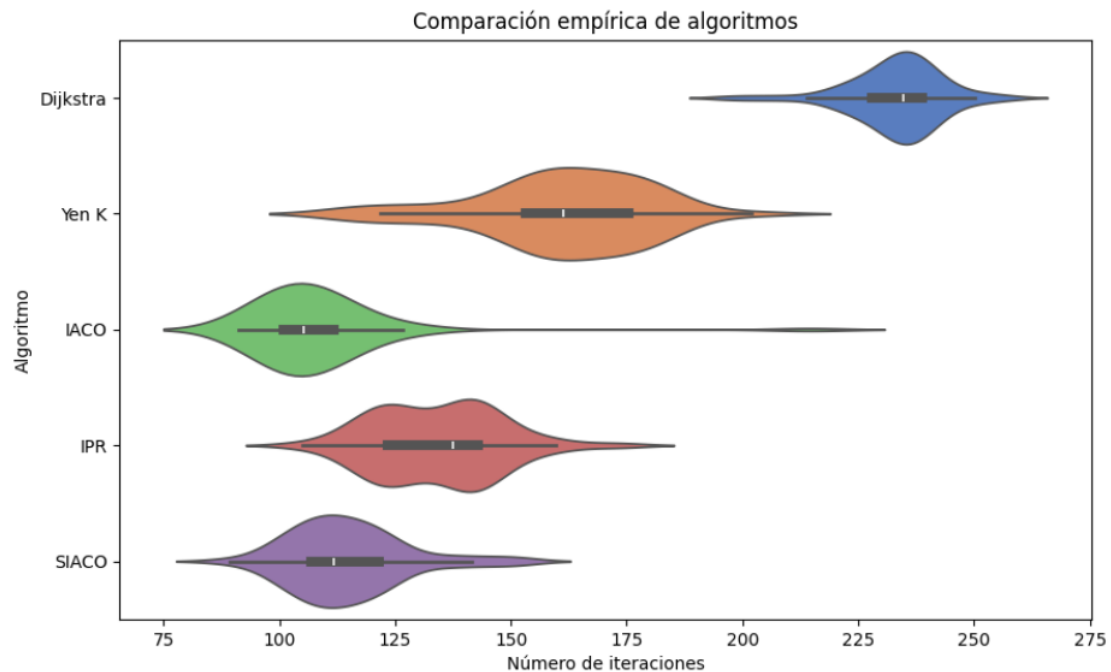


Ilustración 29. Comparación empírica mediante diagramas de violín de los resultados obtenidos por algoritmo.

Como se puede observar en los diagramas de violín, no todos los rendimientos son iguales. Como era de esperar el algoritmo de Dijkstra tiene el peor desempeño, puesto que no se aprovechan la mayoría de las vías para redistribuir el tráfico:

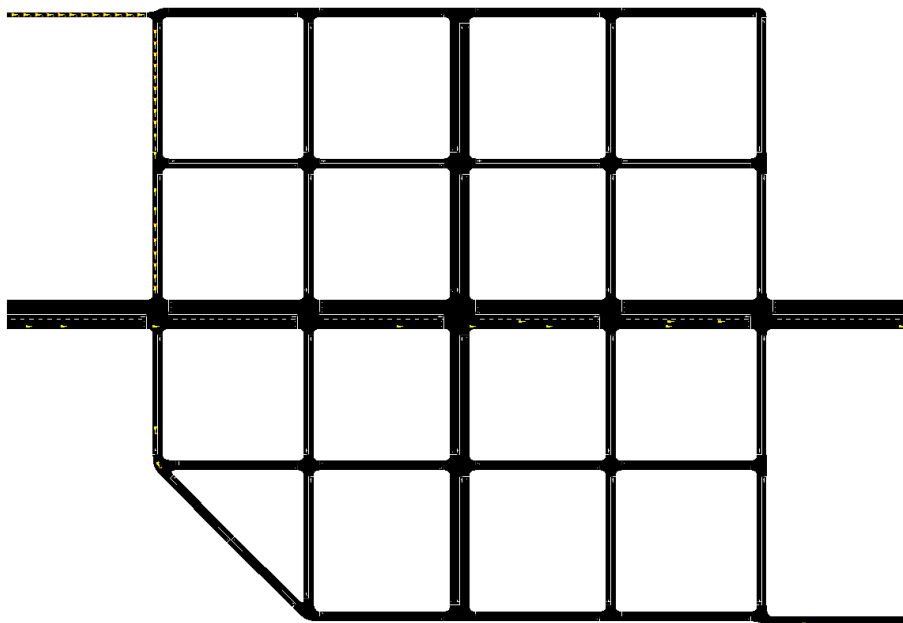


Ilustración 30. Estado de una simulación en la iteración 500 con el algoritmo de Dijkstra.

El algoritmo de Yen presenta mejores resultados, puesto que no todos los agentes se acumulan en las mismas aristas. Sin embargo, dado que sus elecciones no se basan en el nivel de tráfico, aparecen aleatoriamente zonas de congestión, habiendo vías vacías:

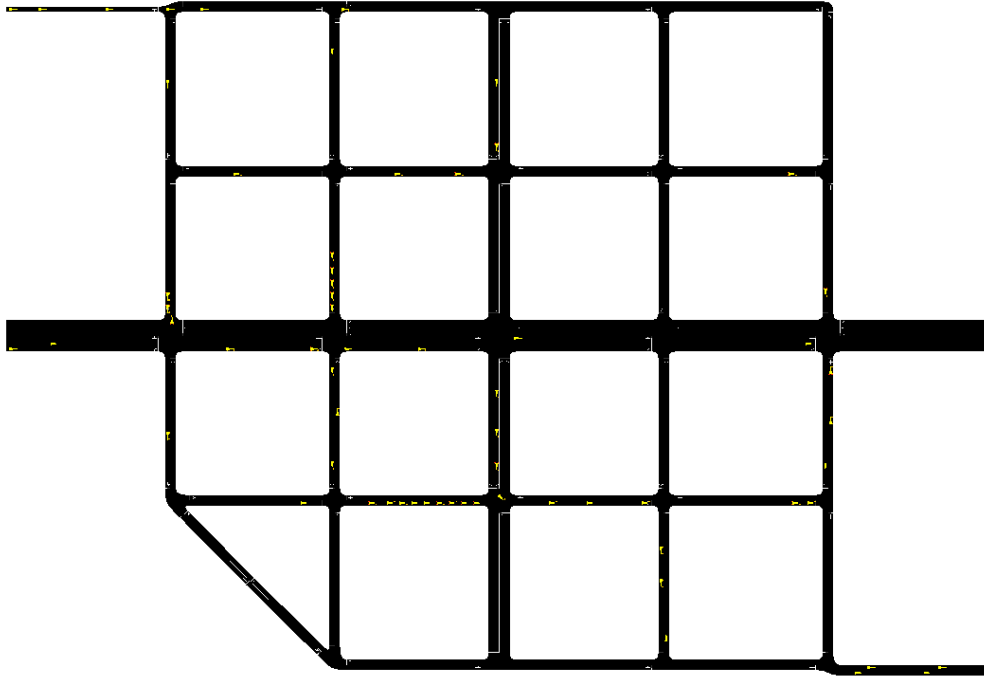


Ilustración 31. Estado de una simulación en la iteración 500 con el algoritmo de Yen.

Por otro lado, los resultados del algoritmo IACO, parecen a priori los mejores. Sus vehículos se distribuyen homogéneamente por la red minimizando la cantidad de vehículos que espera en cada arista antes de cruzar la avenida horizontal:

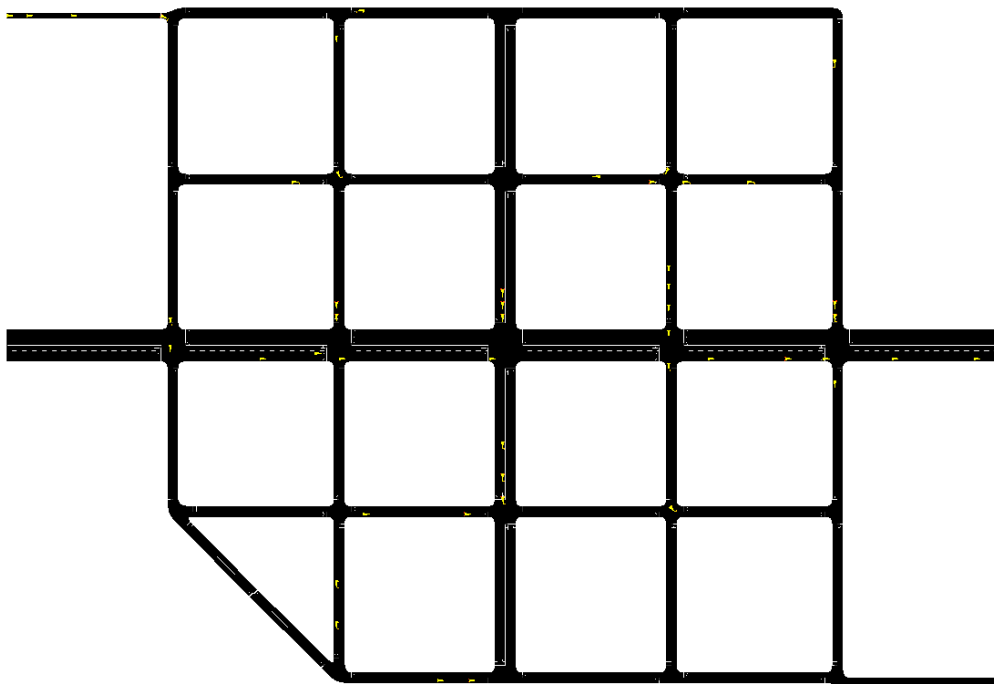


Ilustración 32. Estado de una simulación en la iteración 500 con el algoritmo IACO.

A diferencia del anterior, el algoritmo IPR presenta un rendimiento peor de lo esperado. Dado que es un algoritmo cuya feromona no se propaga y que solo utiliza información local, provoca que los vehículos no puedan tomar decisiones con suficiente antelación:

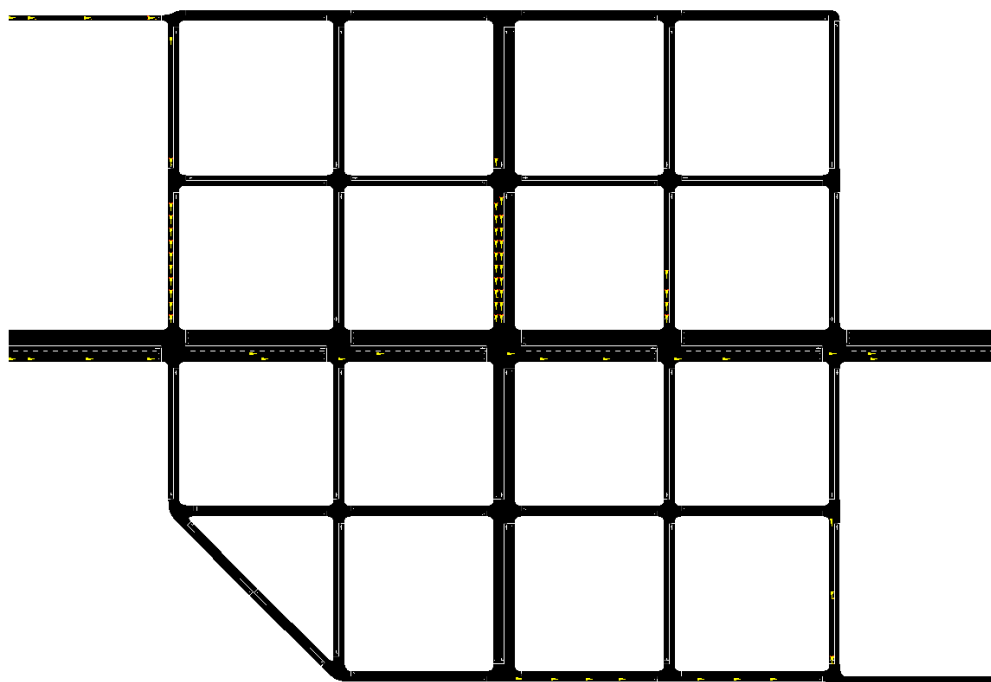


Ilustración 33. Estado de una simulación en la iteración 500 con el algoritmo IPR.

En la ilustración 35 se observa claramente como algunas rutas acaban congestionadas mientras que otras se encuentran vacías. Esto, además de por las explicaciones anteriores puede deberse a que las modificaciones propuestas no sean las óptimas y/o ello conlleve la necesidad de alterar alguno de los parámetros o funciones descritas en el artículo original.

Por último, el algoritmo propuesto, SIACO, presenta, a priori, uno de los mejores resultados, superando a los obtenidos por IPR. No obstante, al no disponer de información global la distribución a lo largo de la red no es igual de homogénea. Como se puede ver a continuación, los vehículos de SIACO suelen evitar el tráfico más inmediato en busca de posibles rutas futuras menos congestionadas. Esto provoca que se acumule más tráfico en aquellas aristas en las que los vehículos ya no pueden seguir esperando a encontrar una ruta mejor:

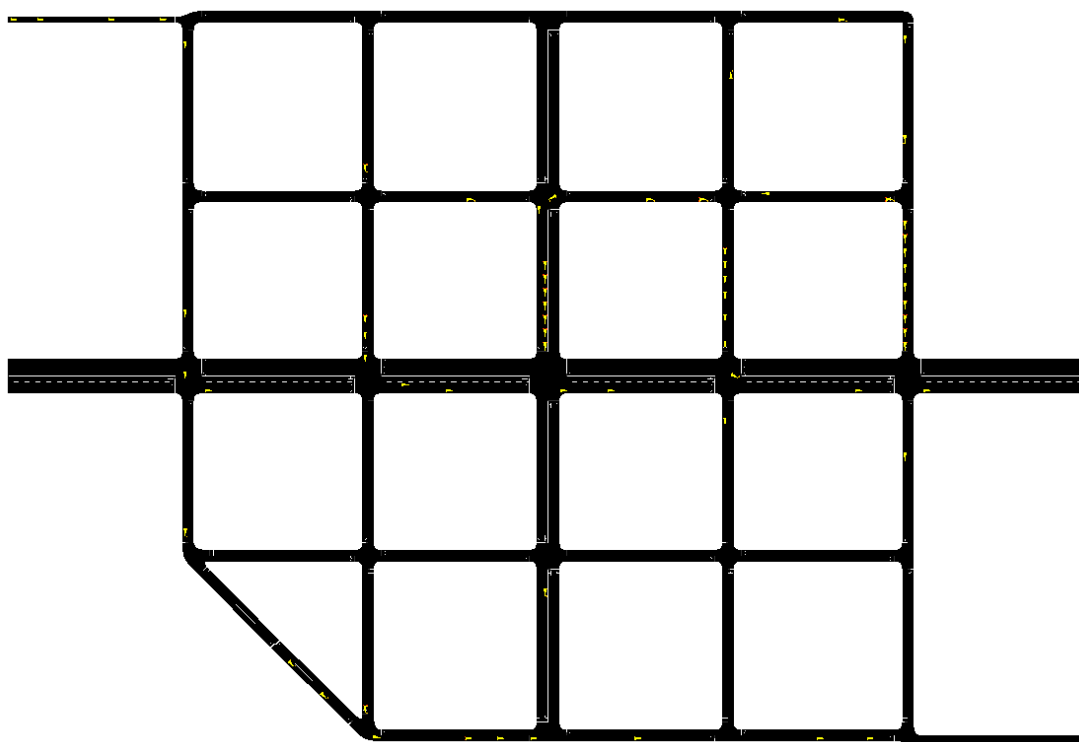


Ilustración 34. Estado de una simulación en la iteración 500 con el algoritmo SIACO.

Esto sugiere la posibilidad de que la feromona se disipa más rápido de lo que lo hace el nivel de tráfico. Por lo que, la información que recibe un vehículo en el nodo S_1 es que el tráfico de las aristas del lado derecho de la cuadrícula posiblemente se haya disipado para cuando este llegue. Es por lo tanto posible que los hiperparámetros puedan seguir mejorando o que sea necesario modificar la función de atenuación de intensidad sonora para que tenga en cuenta este aspecto.

Existe, no obstante, la posibilidad de que esto no sea un fallo o un funcionamiento poco eficiente, sino todo lo contrario, un comportamiento social con una componente más colaborativa. Se ha observado como en algunas ejecuciones de IACO, que todos los agentes se distribuyeran homogéneamente en todas las aristas verticales a la espera de cruzar la avenida horizontal generaba aún más retrasos. Esto se debe a que los vehículos de la primera arista empezando por la izquierda a veces eligen circular por la avenida horizontal, lo que aumenta la densidad vehicular de la misma para la segunda, tercera, cuarta y quinta aristas. Los de la segunda solo aumentan la densidad para la tercera, cuarta y quinta. Y así con las demás. Por lo que los vehículos de la quinta arista, empezando por la izquierda, sufren el tráfico de las cuatro anteriores también. Es posible que los vehículos de SIACO estén por tanto percibiendo la señal de las hormigas de la avenida horizontal y adaptándose a esta circunstancia.

Dada la dificultad de resolver un problema TAP para un humano, incluso en un escenario tan simple, resulta imposible determinar si es un comportamiento óptimo o que se debe pulir. Sería interesante por tanto realizar un estudio que demostrara de cuál de las dos se trata. Sin embargo, atendiendo exclusivamente a los hechos, el desempeño del algoritmo SIACO es considerablemente bueno. Es por tanto imprescindible cuantificar cuanto, es decir, si difiere significativamente de los algoritmos básicos y no es significativamente peor que los algoritmos del estado del arte.

El primer paso, antes de llevar a cabo estas comparaciones, es comprobar la normalidad de las distribuciones para elegir un contraste de hipótesis adecuado. Se ha estudiado la normalidad de las distribuciones de tiempos de viaje medios utilizando la función `normaltest` de la librería Scipy, que emplea una prueba de normalidad D'Agostino (D'agostino & Pearson, 1973). A continuación, se presenta una tabla resumen con los resultados de cada uno de los test.

| Test de normalidad D'Agostino. | | |
|--|---------|------------------------------|
| Hipótesis nula H_0 : Los tiempos de viaje promedio se distribuyen siguiendo una distribución normal. | | |
| Hipótesis alternativa H_A : No hay evidencias suficientes para aceptar que los tiempos de viaje promedios se distribuyen siguiendo una distribución normal | | |
| Algoritmo | P-valor | Rechazar / no rechazar H_0 |
| Dijkstra | 0,001 | Rechazar |
| Yen | 0,083 | Rechazar |
| IPR | 0.530 | No rechazar |
| IACO | 0,001 | Rechazar |
| SIACO | 0,004 | Rechazar |

Tabla 1. Resultados de los contrastes de normalidad por algoritmo.

En general se rechaza la hipótesis de normalidad para casi todas las distribuciones, por lo que será necesario llevar a cabo contrastes no paramétricos para estudiar las diferencias entre las distribuciones.

El contraste elegido se basa en estudiar si la media de las distribuciones difiere significativamente. Para ello se han llevado a cabo múltiples pruebas U de Mann-Whitney (Mann & Whitney, 1947). En cada prueba se ha comparado la media de cada algoritmo con la media del algoritmo propuesto. Para obtener conclusiones más exactas, se han llevado a cabo contrastes unilaterales, por lo que las hipótesis

alternativas han sido planteadas basándose en las evidencias mostradas en la ilustración

31. Nuevamente se presenta una tabla resumen con los resultados:

| Contraste U de Mann-Whitney para la diferencia de medias. | | | |
|--|------------------------------------|-----------------------|------------------------------|
| Hipótesis nula H_0 : Los tiempos de viaje promedio no varían significativamente (para cada pareja de algoritmos estudiados). | | | |
| Algoritmo | Hipótesis alternativa H_A | P-valor | Rechazar / no rechazar H_0 |
| Dijkstra | $\mu_{SIACO} - \mu_{Dijkstra} < 0$ | $3,53 \cdot 10^{-18}$ | Rechazar |
| Yen | $\mu_{SIACO} - \mu_{Yen} < 0$ | $3,34 \cdot 10^{-16}$ | Rechazar |
| IPR | $\mu_{SIACO} - \mu_{IPR} < 0$ | $1,44 \cdot 10^{-10}$ | Rechazar |
| IACO | $\mu_{SIACO} - \mu_{IACO} > 0$ | $2,05 \cdot 10^{-4}$ | Rechazar |

Tabla 2. Resultados de los contrastes sobre diferencia de medias del algoritmo SIACO contra los demás.

En vista de que todos los p-valores obtenidos son próximos a cero, se puede rechazar con rotundidad la hipótesis de que no hay diferencias significativas entre SIACO y los demás algoritmos. Así, se puede establecer que SIACO es significativamente mejor que los algoritmos básicos y que IPR. Sin embargo, los resultados de SIACO también son significativamente peores que los de IACO.

Para obtener una aproximación de la diferencia de rendimientos se utilizará la técnica de bootstrapping (Efron, 1979). Su uso permitirá obtener empíricamente intervalos de confianza evitando restricciones como la normalidad de las distribuciones estudiadas. El proceso seguido para obtener el intervalo de confianza es el siguiente:

1. Se obtienen dos muestras de longitud 50 con reemplazamiento (remuestreo) de cada uno de los algoritmos estudiados: una muestra aleatoria simple de entre las 50 simulaciones de IACO y otra muestra aleatoria simple de entre las simulaciones de SIACO.
2. Se obtienen las medias de cada una de las muestras remuestreadas.
3. Se calcula la diferencia entre las medias: $\mu_i^{IACO} - \mu_i^{SIACO}$, donde: μ_i^a representa la media de la muestra i remuestreada de la distribución de medias del algoritmo a . Por lo que una diferencia positiva indicaría que IACO tiene un mejor rendimiento que SIACO en dicha comparación i . Finalmente este valor calculado al vector de diferencias.
4. Se repiten los pasos 1,2 y 3 un total de 9999 veces. Obteniendo de esta manera un vector con 9999 observaciones sobre diferencias de medias.

5. Se obtiene la distribución asociada a este conjunto de observaciones y se calculan los puntos críticos para construir un intervalo de confianza al 95% sobre la diferencia de medias.

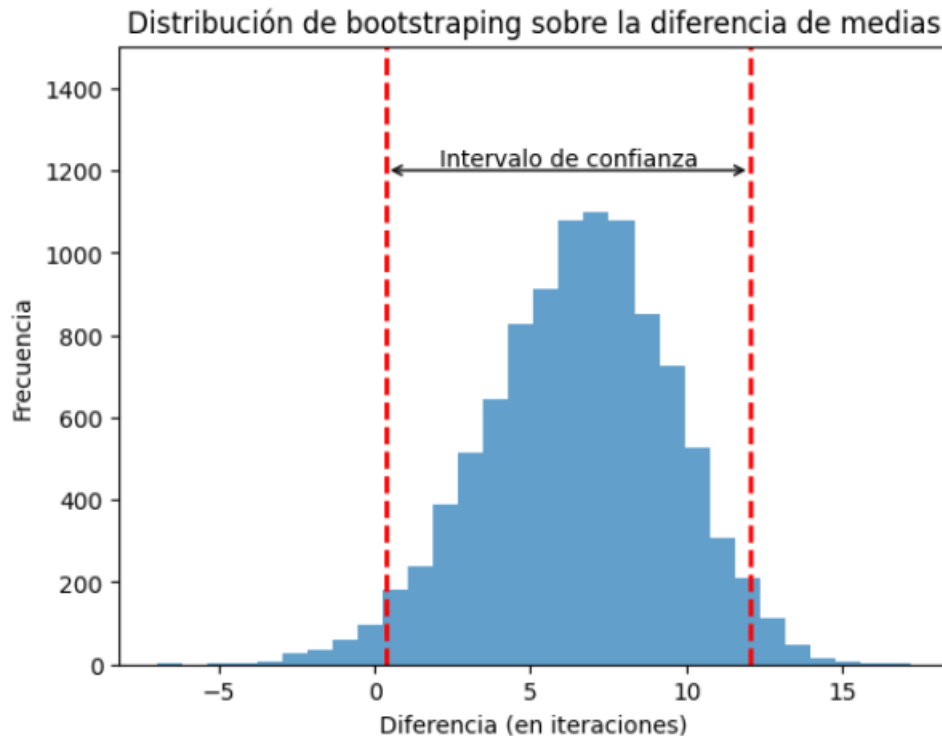


Ilustración 35. Distribución de bootstrapping sobre la diferencia de medias.

El intervalo de confianza al 95% sobre la diferencia media del tiempo promedio de viaje entre ambos algoritmos obtenido es aproximadamente (0.35, 12.05). Como era de esperar, el intervalo de confianza no contiene el valor 0, por lo que no se puede asumir que no exista una diferencia real entre los tiempos medios de viaje al 95% de confianza.

En cuanto a la cuantificación de la diferencia existente, resulta poco útil presentar cantidades absolutas, pues estas dependen de la red en cuestión. Si se repitiera el experimento para la misma topología de red, pero con mayor longitud de todas sus aristas, las diferencias absolutas podrían variar. Para conseguir una perspectiva más objetiva de los resultados obtenidos, es más conveniente estudiar las diferencias desde un punto de vista relativo. Así, el nuevo intervalo de confianza relativo es (1%, 11%), lo que se puede interpretar como que el algoritmo SIACO tiene un rendimiento entre un 1% y un 11% peor que el algoritmo IACO.

No obstante, es importante recalcar nuevamente que los resultados obtenidos solo se pueden afirmar con rotundidad para el caso concreto de estudio propuesto. Los rendimientos reales y sus diferencias podrían llegar a variar en un contexto real, puesto

que, para empezar, los hiperparámetros de ambos modelos han sido optimizados para una red de estas características. Por ello cabe la posibilidad de que su valor haya sido sesgado en el proceso y no se comporte adecuadamente en otros contextos.

Ello no desmerece en absoluto el estudio realizado, puesto que con este se han conseguido dos propósitos. El primero es obtener una consideración general sobre la diferencia de rendimientos entre ambos algoritmos. Con esta podemos establecer que, al menos mediante este método, limitar la información obtenida a un entorno local conlleva una ligera pero existente reducción del rendimiento. La segunda meta alcanzada es la de haber diseñado un protocolo de entrenamiento y análisis de modelos de inteligencia de enjambres capaces de solucionar problemas TAP. Esta resulta especialmente relevante porque es directamente extrapolable a cualquier otro algoritmo o red. Esto facilita enormemente obtener conclusiones relevantes para un despliegue a gran escala, pues el proceso sería equivalente y tan solo haría falta una mayor capacidad de computación que la disponible a la hora de realizar este Trabajo de Fin de Grado.

5. CONCLUSIONES

Debido al constante crecimiento de la densidad poblacional en las áreas urbanas y al consecuente aumento del número de vehículos en circulación, el problema del tráfico en las ciudades se ha agravado en las últimas décadas. Este problema de movilidad urbana requiere de manera apremiante la propuesta de soluciones que mejoren la calidad de vida de sus ciudadanos.

En vista de que la viabilidad de implementación a corto plazo de los algoritmos existentes es limitada, este Trabajo de Fin de Grado propone la creación del algoritmo *Sonic Inverted Ant Colony Optimization* (SIACO). Este algoritmo de optimización por enjambres incorpora elementos y características de algunas de las soluciones existentes, pero prestando especial atención a aspectos relacionados con la adopción de la solución en un contexto real.

5.1 Resultados

Con este objetivo en mente, se ha modificado la interfaz de comunicación entre vehículos y entorno para cambiar su habitual comportamiento inspirado en procesos químicos, que requería el acceso a información global, y, por consiguiente, numerosas

comunicaciones entre todos los vehículos y todos los dispositivos inteligentes de la ciudad; para pasar a un comportamiento inspirado en la propagación de ondas sonoras. Mediante esta modificación se consigue limitar la cantidad de comunicaciones para que cada vehículo solo tenga que interactuar con un único vehículo o dispositivo inteligente.

Pese a esta nueva limitación, la pérdida de información es mínima, pues el rendimiento del algoritmo desarrollado no difiere en gran medida del de los algoritmos existentes previos. Ello se puede verificar gracias al análisis de los resultados de múltiples simulaciones conducidas en la fase de pruebas. Estos revelan que, de media, la diferencia en los tiempos de viaje de los vehículos que utilizan SIACO es solo entre un 1% y un 11% menos efectivo (al 95% de confianza) que el algoritmo en el que se basó su desarrollo: IACO.

A pesar de su bajada de rendimiento, el algoritmo desarrollado tiene una combinación propia de ventajas únicas:

- Propone un modelo de comunicación entre vehículos y entorno eficiente, unidireccional y sencillo de implementar. Esto se debe a que cada vehículo debe recibir un único dato numérico del vehículo que le precede, sin necesidad si quiera de que el predecesor reciba confirmación.
- Su comunicación uno a uno garantiza el anonimato de la información sensible transmitida, abogando por la privacidad del usuario. Además, a medida que los vehículos comunican información, esta se va agregando y modificando, siendo imposible recuperar la información original emitida.
- La transmisión local de información y su característica principal de atenuación añaden robustez al modelo. Esto se debe a que, en caso de que un vehículo o dispositivo dañado, defectuoso o malicioso emita información, el impacto o la confusión generada se limitaría a su entorno local. Esto ocurre porque el valor de la información errónea transmitida disminuiría progresivamente a medida que uno se aleja del agente emisor, hasta desaparecer por completo.
- El carácter reactivo del algoritmo permite a los vehículos tomar decisiones en tiempo real teniendo en cuenta incluso factores para los que no fue diseñado. Entre ellos se podrían encontrar por ejemplo los retrasos ocasionados por los semáforos, accidentes o las inclemencias del tiempo. Esto es gracias a que el modelo propuesto solo reacciona a la densidad vehicular en cada momento, no a su causa.

5.2 Posibles líneas de trabajo futuras

No obstante, el algoritmo está aún lejos de considerarse completo, pues, como se ha visto a lo largo del trabajo, muchos aspectos son aún mejorables o se podrían llevar a cabo estudios más exhaustivos. Entre estos estudios se encontrarían:

- Replicar el algoritmo cambiando la función de atenuación, para la que se propone tomar inspiración de otros fenómenos físicos como la difusión del calor.
- Encontrar un algoritmo de cálculo de rutas más cortas que el algoritmo de Dijkstra empleado. Se propone por ejemplo utilizar algoritmos clásicos de la familia de la optimización por colonias de hormigas.
- Analizar la sostenibilidad del algoritmo a largo plazo. Pues, como se menciona en el texto, es posible que el algoritmo pierda flexibilidad a medida que pasa el tiempo. Se sugiere entonces la definición de un límite de información que es capaz de recordar el sistema.
- Cuantificar el impacto marginal de cada variable que caracteriza una vía y proponer una formulación para la obtención de un coeficiente de absorción acústica característico de cada vía.

Se considera, aun así, que todas las anteriores, modificarían el rendimiento ligeramente, siendo más propias de un proceso de refinado final. Sin embargo, a continuación se proponen dos nuevas líneas de investigación de especial interés que podrían modificar sustancialmente el atractivo del modelo:

- Feromonas características de cada vía: Hasta ahora, todas las feromonas emitidas son indistinguibles. Ello se traduce en que la información que recibe un vehículo en un nodo desde una arista pueda interpretarse como: el retraso esperado generado por los atascos en esta arista **o alguno de los posibles caminos que la contiene** es proporcional a la feromona percibida. Aunque ya se han tomado medidas para tratar de eliminar esta ambigüedad, se propone también caracterizar cada feromona para que se conozca la arista en la que fue emitida. De esta manera, los vehículos comenzarían a emitir un vector de feromonas con aquellas cuyo valor superase el umbral audible.
- Despejar rutas para los vehículos de emergencias: A menudo, los vehículos de servicios de emergencia como ambulancias, policías o bomberos deben

desempeñar sus funciones en entornos urbanos altamente congestionados. Dada la importancia de su trabajo, avisan, mediante la sirena de emergencia, a los conductores que tienen delante para que despejen el camino. Aun así, es habitual que se vean obligadas a reducir su velocidad formándose situaciones como esta:



Ilustración 36. Ambulancia entre el tráfico (Vanguardia, 2022).

Para ello se propone invertir el sentido de propagación de las feromonas, de tal manera que un vehículo de emergencia pueda emitir feromonas a lo largo de la ruta más corta a su destino. Esto permite que a medida que el vehículo de emergencia llegue a cada arista, los vehículos ya se hayan sentido repelidos con antelación y este pueda a máxima velocidad.

Este enfoque no resulta trivial porque es necesario ajustar la cantidad de feromona emitida. Si por ejemplo una ambulancia pudiera emitir a lo largo de su ruta una cantidad, digamos infinita, de feromonas, sin duda despejaría inmediatamente el camino a seguir. No obstante, estas feromonas seguirían propagándose por la red y causarían el efecto de feromona residual apreciable en la ilustración 27, pudiendo confundir a vehículos en un área considerable. Por otro lado, tampoco necesitarían cuatro carriles vacíos para llegar a su destino. Sería necesario entonces cuantificar la feromona mínima que permitiera a estos vehículos circular a la máxima velocidad sin que ello afectara significativamente a otros usuarios.

5.3 Consideraciones finales

Resolver el problema de movilidad urbana se trata sin duda de un tema poco explorado hasta la fecha, por lo que la cantidad de posibles mejoras o nuevas propuestas podría extenderse indefinidamente. Y es que, con este Trabajo de Fin de Grado se ha buscado dar solución a un problema que podría ser tan solo la punta del iceberg. No obstante, un problema que será determinante en un futuro no muy lejano. Desde mi perspectiva, y de manera completamente subjetiva, creo que queda un largo recorrido hasta alcanzar la visión actual de la "ciudad del futuro", para el que la inteligencia de enjambres podría suponer una herramienta indispensable, especialmente teniendo en cuenta la creciente cantidad de redes de dispositivos conectados a internet. Este Trabajo de Fin de Grado me ha demostrado que el poder de las cosas simples puede equipararse al de las complejas redes neuronales con millones de parámetros. Ha cambiado sin duda mi perspectiva sobre la utilidad de las matemáticas añadiendo una nueva dimensión y me ha permitido dar el primer paso en ese "camino hacia las ciudades del futuro". Por tanto, tras haber concluido este proyecto, no puedo sino esperar ansioso el momento de dar el siguiente paso.

6. REFERENCIAS

6.1 Bibliografía.

- Abdellah, A. R., Artem, V., Muthanna, A., Gallyamov, D., & Koucheryavy, A. (2020). Deep Learning for IoT Traffic Prediction Based on Edge Computing. En *Communications in computer and information science* (pp. 18-29). https://doi.org/10.1007/978-3-030-66242-4_2
- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna. *KDD '19: Proceedings Of The 25th ACM SIGKDD International Conference On Knowledge Discovery & Data Mining*. <https://doi.org/10.1145/3292500.3330701>
- Asimov, I. (1950). Runaround. En *I, robot*. Roc.
- Bektas, T. (2006). The multiple traveling salesman problem: an overview of formulations and solution procedures. *Omega*, 34(3), 209-219. <https://doi.org/10.1016/j.omega.2004.10.004>
- Chakraborty, A., & Kar, A. K. (2017). Swarm Intelligence: A Review of Algorithms. En *Nature-Inspired Computing and Optimization* (pp. 475-494). https://doi.org/10.1007/978-3-319-50920-4_19
- D'agostino, R., & Pearson, E. S. (1973). Tests for departure from normality. Empirical results for the distributions of b_2 and $\sqrt{b_1}$. *Biometrika*, 60(3), 613-622. <https://doi.org/10.1093/biomet/60.3.613>
- Dias, J. C., Machado, P., Silva, D. C., & Abreu, P. H. (2014). An Inverted Ant Colony Optimization approach to traffic. *Engineering Applications Of Artificial Intelligence*, 36, 122-133. <https://doi.org/10.1016/j.engappai.2014.07.005>
- Dirección General de Tráfico [DGT]. (2013, 22 octubre). 2 segundos, una vida. *Revista Tráfico y Seguridad Vial*. Recuperado 3 de febrero de 2024, de <https://revista.dgt.es/es/educacion-formacion-old/conducir-mejor/2013/distancia-seguridad.shtml>
- Dirección General de Tráfico [DGT]. (2015, julio). Velocidad, distancia de reacción y distancia de frenado. *Revista Tráfico y Seguridad Vial*. Recuperado 3 de febrero

de 2024, de <https://revista.dgt.es/es/multimedia/infografia/2015/0706-Distancia-reaccion-y-frenado-campo-futbol.shtml>

Dirección General de Tráfico [DGT]. (2023, diciembre). *Consejos y normas de Seguridad Vial*. https://www.dgt.es/export/sites/web-DGT/galleries/downloads/conoce-el-estado-del-traffic/operaciones-especiales/4_Operacion-de-Trafico-Navidad-2023-2024_Consejos-y-normas-de-Seguridad-Vial_V-I.pdf

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions On Systems, Man And Cybernetics. Part B. Cybernetics*, 26(1), 29-41. <https://doi.org/10.1109/3477.484436>

Dorokhin, S., Artemov, A., Likhachev, D., Novikov, A., & Starkov, E. (2020). Traffic simulation: an analytical review. *IOP Conference Series. Materials Science And Engineering*, 918(1), 012058. <https://doi.org/10.1088/1757-899x/918/1/012058>

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *Annals Of Statistics*, 7(1). <https://doi.org/10.1214/aos/1176344552>

Efron, B., & Tibshirani, R. J. (1993). An Introduction to the Bootstrap. En *Springer eBooks*. <https://doi.org/10.1007/978-1-4899-4541-9>

Encuesta de Características Esenciales de la Población y las Viviendas. (2021). En *Instituto Nacional de Estadística (INE)*.

Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12(10), 2451-2471. <https://doi.org/10.1162/089976600300015015>

Grasse, P. P. (1959). Insectes Sociaux. En *La reconstruction du nid et les coordinations interindividuelles chezBellicositermes natalensis etCubitermes sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs* (pp. 41-81).

Hennessy, D. A., & Wiesenhal, D. L. (1997). The relationship between traffic congestion, driver stress and direct versus indirect coping behaviours. *Ergonomics*, 40(3), 348-361. <https://doi.org/10.1080/001401397188198>

- Ho, M. C., Lim, J. M., Soon, K. L., & Chong, C. Y. (2019). An improved pheromone-based vehicle rerouting system to reduce traffic congestion. *Applied Soft Computing*, 84, 105702. <https://doi.org/10.1016/j.asoc.2019.105702>
- Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term memory. *Neural Computation*, 9(8), 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hunter, J. D. (2007). Matplotlib: a 2D Graphics environment. *Computing In Science & Engineering*, 9(3), 90-95. <https://doi.org/10.1109/mcse.2007.55>
- Jakubiak-Lasocka, J., Lasocki, J., Siekmeier, R., & Chłopek, Z. (2014). Impact of Traffic-Related Air Pollution on Health. En *Advances in experimental medicine and biology* (pp. 21-29). https://doi.org/10.1007/5584_2014_14
- Jing, X., Liu, Y., & Cao, W. (2012). A Hybrid Genetic Algorithm for Route Optimization in Multimodal Transport. *Fifth International Symposium On Computational Intelligence And Design*. <https://doi.org/10.1109/iscid.2012.73>
- Karaboga, D. (2005). AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION. En *Erciyes University, Engineering Faculty, Computer Engineering Department* (Technical report-tr06; p. 0). Recuperado 15 de noviembre de 2023, de https://abc.erciyes.edu.tr/pub/tr06_2005.pdf
- Kennedy, J., & Eberhart, R. (2002). Particle swarm optimization. *Proceedings Of ICNN'95 - International Conference On Neural Networks*. <https://doi.org/10.1109/icnn.1995.488968>
- Khamis, A. (2024). SWARM INTELLIGENCE ALGORITHMS. En *Optimization Algorithms* (10.a ed., pp. 431-535).
- Kinsler, L. E., Frey, A. R., Coppens, A. B., & Sanders, J. V. (2000). ABSORPTION AND ATTENUATION OF SOUND. En *Fundamentals of Acoustics* (4.a ed., pp. 210-245). John Wiley & Sons.
- Lopez, P. A., Wiessner, E., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flotterod, Y., Hilbrich, R., Lucken, L., Rummel, J., & Wagner, P. (2018). Microscopic Traffic Simulation using SUMO. *2018 21st International Conference On Intelligent Transportation Systems (ITSC)*. <https://doi.org/10.1109/itsc.2018.8569938>
- Mallik, S. (2014). Intelligent Transportation System. En *International Journal of Civil Engineering Research* (Vol. 5, pp. 367-372).

- Mann, H. B., & Whitney, D. R. (1947). On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *Annals Of Mathematical Statistics*, 18(1), 50-60. <https://doi.org/10.1214/aoms/1177730491>
- Martínez Parpolowicz, Á. (2024). *Desarrollo de un entorno de simulación de tráfico para la implementación de soluciones basadas en sistemas multiagente* [Trabajo de Fin de Grado]. U-tad | Centro Universitario de Tecnología y Arte Digital.
- Nguyen, T., & Jung, J. J. (2021). Inverse pheromone-based decentralized route guidance for connected vehicles. En *Proceedings of the 36th Annual ACM Symposium on Applied Computing* (pp. 459-463). <https://doi.org/10.1145/3412841.3441925>
- Patriksson, M. (1994). *The Traffic Assignment Problem: Models and Methods*. <https://ci.nii.ac.jp/ncid/BA2532277X>
- Python Software Foundation. (2023). *Python Language Reference, Version 3.10*. <https://docs.python.org/3.10/>
- Reynolds, C. W. (1987). Flocks, herds and schools: A distributed behavioral model. *Computer Graphics*, 21(4), 25-34. <https://doi.org/10.1145/37402.37406>
- Simulife Hub. (2023, 5 mayo). *Swarm intelligence simulation. Project «Screaming Insects»* [Vídeo]. YouTube. <https://www.youtube.com/watch?v=Yu7sF9rcVJY>
- Soon, K. L., Lim, J. M., Parthiban, R., & Ho, M. C. (2019). Proactive eco-friendly pheromone-based green vehicle routing for multi-agent systems. *Expert Systems With Applications*, 121, 324-337. <https://doi.org/10.1016/j.eswa.2018.12.026>
- Tang, J., Liu, G., & Pan, Q. (2021). A Review on Representative Swarm Intelligence Algorithms for Solving Optimization Problems: Applications and Trends. *IEEE/CAA Journal Of Automatica Sinica*, 8(10), 1627-1643. <https://doi.org/10.1109/jas.2021.1004129>
- Vanguardia, L. (2022, 21 marzo). ¿Qué hay que hacer si viene una ambulancia por la carretera? *La Vanguardia*. <https://www.lavanguardia.com/motor/20220320/8139206/que-hay-hacer-viene-ambulancia-carretera-pmv.html>

- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., Van Der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., . . . Vázquez-Baeza, Y. (2020). SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261-272. <https://doi.org/10.1038/s41592-019-0686-2>
- Von Frisch, K. (1950). *Bees: their vision, chemical senses, and language*. <https://books.google.es/books?hl=es&lr=&id=5AfxAwAAQBAJ&oi=fnd&pg=PT4&dq=waggle+dance++Karl+Ritter+von+Frisch&ots=O6ipNNnt-7&sig=VInX5O0mvdwYs3YzzZVkJmm2mK8#v=onepage&q&f=false>
- Wahab, M. N. A., Nefti-Meziani, S., & Atyabi, A. (2015). A Comprehensive Review of Swarm Optimization Algorithms. *PloS One*, 10(5), e0122827. <https://doi.org/10.1371/journal.pone.0122827>
- Wooldridge, M. (2009). *An Introduction to MultiAgent Systems*. John Wiley & Sons.
- Yen, J. Y. (1971). Finding the K Shortest Loopless Paths in a Network. *Management Science*, 17(11), 712-716. <https://doi.org/10.1287/mnsc.17.11.712>

6.2 Índice de imágenes.

| | |
|---|----|
| Ilustración 1. Principal medio de transporte usado por los españoles para ir a su lugar habitual de trabajo (Nota: Elaboración propia a partir de datos del Instituto Nacional de Estadística (INE), 2021)..... | 10 |
| Ilustración 2. Número de publicaciones por algoritmo de inteligencia de enjambres 2000-2020. (Wahab et al., 2015) | 13 |
| Ilustración 3. Reglas de las bandadas de pájaros: Separación, alineamiento y coherencia. (Khamis, 2024) | 14 |
| Ilustración 4. Proceso de optimización por colonias de hormigas. (Wahab et al., 2015) | 20 |
| Ilustración 5. Algoritmo Artificial Bee Colony. (Khamis, 2024) | 21 |
| Ilustración 6. Velocidad, distancia de reacción y distancia de frenado (Dirección General de Tráfico [DGT], 2015) | 23 |

| | |
|--|----|
| Ilustración 7. Hierarchical Multi-Agent Pheromone-based System (HMAPS) (Soon et al., 2019). | 30 |
| Ilustración 8. Ejemplo simple de tráfico discretizado..... | 35 |
| Ilustración 9. Ejemplo simple de tráfico discretizado con pesos dependientes. | 38 |
| Ilustración 10. Representación visual de aristas. | 38 |
| Ilustración 11. Tipos de aristas. | 39 |
| Ilustración 12. Escenario básico general discretizado. | 39 |
| Ilustración 13. Ejemplo de incorporación discretizada..... | 40 |
| Ilustración 14. Ejemplo de incorporación discretizada con densidad vehicular máxima. | 41 |
| Ilustración 15. Ejemplo de incorporación discretizada con densidad vehicular no máxima..... | 42 |
| Ilustración 16.Caso1: Otro vehículo accedió a B desde la arista con prioridad en t+1. | 43 |
| Ilustración 17. Caso 2: Nadie accedió a B desde la arista con prioridad en t+1..... | 43 |
| Ilustración 18. Propagación de la información en el ejemplo discretizado. | 45 |
| Ilustración 19. Función de atenuación y umbral perceptible. | 50 |
| Ilustración 20. Nodo con grado de salida mayor que uno. ¿Quién es el predecesor del vehículo verde? | 51 |
| Ilustración 21. Nodo con grado de entrada mayor que uno. ¿Quién es el sucesor del vehículo verde? | 51 |
| Ilustración 22. Nodo con grados de entrada y salida mayor que uno. ¿En qué se diferencia el tráfico que percibe el vehículo verde del que percibe el morado?..... | 51 |
| Ilustración 23. Caso de estudio con densidades vehiculares desbalanceadas | 52 |
| Ilustración 24. Comparación de la función de atenuación con distintos valores de k. | 54 |
| Ilustración 25. Efecto de la feromona residual. | 55 |
| Ilustración 26. Red sintética creada usando la herramienta Netedit del paquete de SUMO..... | 59 |

| | |
|--|----|
| Ilustración 27. Digrafo asociado a la red creada..... | 59 |
| Ilustración 28. Red irregular creada a partir de modificaciones aleatorias en la red sintética de la ilustración 29..... | 60 |
| Ilustración 29. Comparación empírica mediante diagramas de violín de los resultados obtenidos por algoritmo. | 65 |
| Ilustración 30. Estado de una simulación en la iteración 500 con el algoritmo de Dijkstra..... | 65 |
| Ilustración 31. Estado de una simulación en la iteración 500 con el algoritmo de Yen. | 66 |
| Ilustración 32. Estado de una simulación en la iteración 500 con el algoritmo IACO. | 66 |
| Ilustración 33. Estado de una simulación en la iteración 500 con el algoritmo IPR. .. | 67 |
| Ilustración 34. Estado de una simulación en la iteración 500 con el algoritmo SIACO. | 68 |
| Ilustración 35. Distribución de bootstrapping sobre la diferencia de medias. | 71 |
| Ilustración 36. Ambulancia entre el tráfico (Vanguardia, 2022). | 75 |

6.3 Índice de tablas

| | |
|--|----|
| Tabla 1. Resultados de los contrastes de normalidad por algoritmo. | 69 |
| Tabla 2. Resultados de los contrastes sobre diferencia de medias del algoritmo SIACO contra los demás..... | 70 |