

ALBERTO DÍAZ ÁLVAREZ

MODELADO DE COMPOR- TAMIENTO DE CONDUCTO- RES CON TÉCNICAS DE INTELIGENCIA COMPU- TACIONAL

UNIVERSIDAD POLITÉCNICA DE MADRID

Alberto Díaz Álvarez

Modelado de comportamiento de conductores con técnicas de Inteligencia Computacional

Tesis doctoral, 22 de septiembre de 2016

Revisores: Rev1, Rev2 y Rev3

Supervisores: Dr. Francisco Serradilla García y Dr. Felipe Jiménez Alonso

Instituto Universitario de Investigación del Automóvil

Universidad Politécnica de Madrid

Campus Sur UPM, Carretera de Valencia (A-3), km7

28031 Madrid

Este documento esta realizado bajo licencia **Creative Commons “Reconocimiento-CompartirIgual 3.0 España”** .



Índice general

I Introducción 21

Introducción 23

II Estado de la cuestión 29

Inteligencia Computacional 31

Simulación de tráfico 55

Modelos de comportamiento 69

III Desarrollo de la tesis 83

Sistemas desarrollados 85

Estudio de modelos de comportamiento 87

Resultados 89

Conclusiones 91

Glosario

AORTA (*Approximately Orchestrated Routing and Transportation Analyzer*). Entorno de microsimulación multiagente de tráfico desarrollado en el departamento de Ciencias de la Computación de la Universidad de Austin (Texas). Url: <http://www.aorta-traffic.org/>.. 64

car-following Modelo de control de conductor que basa su comportamiento en el vehículo del mismo carril que le precede.. 15, 16, 58, 59, 71

GPL (*General Public License*). Licencia de software que garantiza las libertades del software. Asegura que cualquier versión, extensión o software derivado de éste permanecerá siendo software libre. Su última versión es la 3.0. Url: <https://www.gnu.org/licenses/gpl-3.0.html>.. 65

MatSIM (*Multi-Agent Transport Simulation*). Software de microsimulación multiagente desarrollado en la ETH Zürich. Url: <http://matsim.org>.. 64

MovSim (*Multi-model Open-source Vehicular-traffic Simulator*). Software de microsimulación que implementa tanto modelo multiagente y como modelo basado en CAs. Url: <http://www.movsim.org/>.. 16, 62

Python Lenguaje de programación OSS. Url: <https://www.python.org/>.. 10

SUMO (*Simulation of Urban MObility*). Entorno de micro y mesosimulación multiagente desarrollado por el instituto de sistemas de transporte del DLR (Centro Aeroespacial Alemán). Url: <http://www.dlr.de>.. 9, 10, 16, 17, 19, 64–66

TORCS (*The Open Racing Car Simulator*). Software de simulación creado en un principio como videojuego de carreras y que ha evolucionado hacia plataforma de simulación de técnicas de Inteligencia Artificial aplicadas a la conducción. Url: <http://torcs.sourceforge.net/>.. 81

TraaS (*TraCI as a Service*). APIs basado en SOAPs para interactuar con el simulador SUMO cuando está funcionando en modo servidor. Url: [http://traas.sourceforge.net/cms/..](http://traas.sourceforge.net/cms/) 17, 66

TraCI (*Traffic Control Interface*). Término que sirve tanto para denominar al protocolo de comunicación ofrecido por SUMO en modo servidor para la interacción remota con la simulación como para denominar a la librería desarrollada para abstraer dicho protocolo cuando se trabaja desde Python. Url: <http://www.sumo.dlr.de/wiki/TraCI..> 16, 17, 66, 85

Siglas

ACL Agent Communications Language. 53

ADAS Advanced Driver Assistance System. 80

AI Artificial Intelligence. 23, 31–34, 47, 48, 52, 72, 76

ANN Artificial Neural Network. 31, 32, 35–37, 48, 76

CA Cellular Automaton. 56, 58–62

CEP Complex Event Processing. 80

CI Computational Intelligence. 23, 24, 26, 31, 33–35, 39, 48, 70, 76

DAI Distributed Artificial Intelligence. 52

DVU Driver-Vehicle Unit. 27, 55, 62–64, 74

EC Evolutionary Computation. 39, 86

ES Evolution Strategy. 39

FCS Fuzzy Control System. 38

FIS Fuzzy Inference System. 38

FL Fuzzy Logic. 32, 48, 76

GA Genetic Algorithm. 32, 39–43, 46

GGGP Grammar Guide Genetic Programming. 46

GHR Gazis-Herman-Rothery. 76

GP Genetic Programming. 39

HC Hard Computing. 34

ITS Intelligent Transport System. 23, 69

IvCA Intra-vehicular Context Awareness. 80

KQML Knowledge Query and Manipulation Language. [53](#)

MAS Multi-Agent System. [24](#), [26](#), [52](#), [53](#), [59](#), [61–63](#), [69](#)

ML Machine Learning. [33](#)

NDS Naturalistic Driving Study. [26](#)

NLP Natural Language Processing. [33](#)

OSS Open Source Software. [63](#), [64](#)

RS Recommender System. [33](#)

SC Soft Computing. [34](#)

SI Swarm Intelligence. [39](#)

V2I Vehicle-to-Infrastructure. [62](#)

V2V Vehicle-to-Vehicle. [62](#)

Índice de figuras

1. Desde el comienzo de la revolución industrial, el uso masivo de combustibles fósiles y el crecimiento de la población propició un aumento desproporcionado de CO₂ a la atmósfera, tendencia que sigue en aumento aún con la (lenta) adopción del vehículo eléctrico. La gráfica muestra cómo ambos valores parecen estar correlacionados. Fuente: Environmental Defense Fund (edf.org). 25
2. Último censo de conductores según género segmentado por edades. Fuente: Dirección General de Tráfico (dgt.es). 27
3. Ilustración del Test de Turing. Modelo propuesto para probar si una máquina es capaz de exhibir comportamiento inteligente similar al del ser humano. Hay tres participantes, dos humanos (*A* y *C*) y una máquina (*B*), separados entre sí pero pudiendo intercambiarse mensajes de texto. *C* envía preguntas a *A* y *B* sin saber quién es humano y quién es máquina y éstos le responden. Si *C* no es capaz de identificar qué participante es la máquina, se puede concluir que la máquina es inteligente. Fuente: Hugo Férée, via Wikimedia Commons. 31
4. “La habitación china” es un experimento mental propuesto por John Searle que parte de un Test de Turing donde la máquina ha aprendido a hablar chino. Ésta es reemplazada por una persona que no sabe nada del idioma pero que va equipada con manual de correspondencias (ideograma de entrada → ideograma de salida). Cuando una persona le manda mensajes en chino, esta otra responde. ¿Podemos decir que dicha persona sabe chino? Evidentemente no. Pero entonces, ¿cómo podemos asegurar que la máquina reemplazada ha “aprendido” chino?. Autor: Jolyon Troscianko (<http://www.jolyon.co.uk/>) 31
5. Diferentes objetivos perseguidos por la Inteligencia Artificial. Las filas diferencian entre pensamiento o comportamiento mientras que las columnas separan entre inteligencia humana o el ideal de la inteligencia (racionalidad). Fuente: *Artificial Intelligence: A Modern Approach* (3rd Ed.), [Russell et al., 2003]. 33

6. Una sección del neocórtex humano, región asociada a las capacidades cognitivas y que supone alrededor de un 76% del volumen total del cerebro humano. Está distribuido en 6 capas y miles de columnas que las atraviesan, cada una con alrededor de 10,000 neuronas y un diámetro de 0,5mm. Fuente: *Blue Brain Project EPFL*, <http://bluebrain.epfl.ch/>. 35
7. Variación de la representación del modelo de neurona artificial propuesto por McCullochs y Pitts. En éste, cada una de las entradas x_i es incrementada o inhibida aplicando el producto con su peso asociado w_i . La activación vendrá determinada por la aplicación de una función (denominada “de activación”) a la suma de los valores. Esta variación en concreto incluye una entrada x_0 y un peso w_0 como bias de la neurona para la variación dinámica del umbral de activación. 36
8. Esquema general del funcionamiento de un Genetic Algorithm. Se parte de una población de individuos y en iteraciones de selección-recombinación-mutación-reemplazo se va mejorando la población hasta llegar a una solución que, si bien no tiene por qué ser necesariamente la mejor, es lo suficientemente buena como para ser válida. 40
9. No sé cómo explicarlo. uando esté más despierto mejor. 41
10. Representación del fitness como función inversamente proporcional al error de una solución. Esta forma mantiene el fitness siempre en el intervalo $(0, 1]$. Cuanto menor es el error, esto es, cuanto más tiende a 0, más se acerca el fitness a 1. 43
11. Tres ilustraciones de las técnicas de selección más comunes en Genetic Algorithm. De izquierda a derecha (a) selección proporcional al fitness, (b) selección proporcional a la posición y (c) selección por torneo. 43
12. Situaciones posibles cuando la cardinalidad del conjunto de las codificaciones de nuestro problema es (a) mayor, (b) menor o (c) igual que la cardinalidad del conjunto del espacio de soluciones posibles. 44
13. Diferencias entre una representación binaria y una representación real de un problema. 45
14. Diferencias entre un árbol que codifica un programa en GP y uno que codifica una palabra de una gramática libre de contexto en Grammar Guide Genetic Programming. 46
15. Esquema de un agente y sus propiedades. Aunque no existe una definición comúnmente aceptada de agente, sí que existe una serie de propiedades que los que los identifican. Es autónomo, opera realiza acciones sobre un entorno dependiendo de las percepciones que le llegan de éste y tiene la capacidad de comunicarse con el resto de elementos, incluidos otros agentes. 48

16. Arquitectura básica de un agente. Aunque existen múltiples arquitecturas diferentes, todas se basan en la misma estructura. El agente percibe el entorno, lo interpreta y toma la decisión de cómo actuar sobre él. 50
17. Ilustración de la diferencia entre un agente sin modelo de entorno y uno con modelo de entorno. Cada acción realizada por el agente con modelo de entorno tiene en cuenta el estado del entorno en momentos pasados. El agente sin modelo de entorno actúa tal y como interpreta el entorno en cada momento, como si sufriese de amnesia. 50
18. Distintas arquitecturas de agentes en función del comportamiento. Dependiendo de las acciones a realizar, se identifican tres tipos, los reactivos que aplican una acción sin proceso deductivo y los basados en modelo y utilidad (en algunos contextos denominados deliberativos) que basan su comportamiento en alguna forma de deducción. 51
19. Ilustración de la diferencia entre un agente sin modelo de entorno y uno con modelo de entorno. Cada acción realizada por el agente con modelo de entorno tiene en cuenta el estado del entorno en momentos pasados. El agente sin modelo de entorno actúa tal y como interpreta el entorno en cada momento, como si sufriese de amnesia. 53
20. Taxonomía clásica de simuladores en función de la granularidad (complejidad) de la simulación. En la imagen de la izquierda se muestra un ejemplo clásico de macrosimulador donde el tráfico se modela como un flujo a través de las vías. En la de la derecha, se ilustra un modelo clásico de microsimulación donde cada elemento (en este caso vehículos) circula por un carril de la vía. 57
21. Otras aproximaciones alternativas de modelos en función de la complejidad. Ejemplo de mesosimulación como ventana de microsimulación dentro de un flujo de tráfico en un macrosimulador (e.g. [Munoz et al., 2001]) y ejemplo de submicrosimulación donde se modelan componentes internos de un vehículo. 57
22. Simulador de tráfico basado en CAs. En éste, el espacio se divide en celdas que pueden, o bien estar vacías, o bien ocupadas por un vehículo a una velocidad (más oscuro implica más lento). Concretamente muestra la evolución a lo largo del tiempo del movimiento de un modelo de car-following de 5 vehículos donde en eje x representa la posición en la vía y el eje y el momento temporal (iteración) de la vía. Ilustración generada con el simulador nagel-scherckenberg-demo desarrollado en esta tesis. 58

23. Ejemplo de un modelo lineal en un espacio continuo. La posición del vehículo es un valor $x \in \mathbb{R}$. Este ejemplo muestra un modelo de car-following donde el comportamiento de la aceleración del vehículo es determinado por la distancia al coche siguiente. **TODO!** Si hago yo la ilustración, me quito de esta cita y de este dibujo tan feo. Fuente: [Tordeux et al., 2011]. 59
24. Aparición de retenciones en una autopista de 250 celdas de usando el modelo Nagel-Scherckenberg. La densidad de ocupación es de 50 coches en la vía, la velocidad máxima es de $5c/\Delta t$ y la probabilidad de frenada es de $p = 0,5$. Se puede observar en la figura cómo se desplazan las olas del atasco a lo largo de las 100 iteraciones. Ilustración generada con el simulador nagel-scherckenberg-demo desarrollado para esta tesis. 60
25. Simulación de comportamiento en intersección basada en un MAS. En ésta, cada uno de los vehículos representa a un vehículo real que posee un controlador para hacerlo autónomo. Modelar este caso de estudio con una arquitectura basada en MASs permite centrarse en el diseño del agente en concreto (i.e. el controlador de conducción del vehículo) y estudiar el comportamiento emergente surgido de la interacción de todos los agentes. Fuente: Proyecto AIM (<http://www.cs.utexas.edu/~aim/>). 61
26. Captura de pantalla del simulador MovSim. Este simulador implementa un modelo multiagente donde los vehículos incorporan sistemas de comunicación vehicular. El estudio se centra en el uso de la comunicación entre vehículos para el acoplamiento dinámico de vehículos en sus respectivos carriles. Fuente: [Gu et al., 2015]. 62
27. Captura de pantalla del simulador SUMO. Además de software de simulación propiamente dicho, SUMO provee de una interfaz gráfica que permite una visualización general, de zonas y de elementos en concreto a la vez que permite la variación de configuración de la simulación durante el desarrollo de la misma. **TODO!** Meter una imagen de nuestras simulaciones cuando estén, a poder ser sin color. Hacer el alto más pequeño 65
28. SUMO ofrece la posibilidad de interactuar con la simulación desde cualquier aplicación a través del uso del protocolo TraCI. En la figura podemos ver, de izquierda a derecha, ejemplos de comunicación a través de la interfaz como el *handshake* o inicialización, mensajes de obtención de información y modificación de la misma más una solicitud de avance de paso en la simulación y una señal de finalización de simulación y desconexión. 66

29. Concepto arquitectural de la plataforma TraaS. La plataforma se conecta como cliente a SUMO y ofrece un API basado en SOAP de mensajes que traduce en mensajes del protocolo TraCI, lo que independiza completamente la elección de lenguaje de programación a la vez que abstrae los detalles del protocolo de comunicación. 66

30. Los tres niveles jerárquicos que describen la tarea de conducción según [Michon, 1985]: *estrategia* (i.e. las decisiones generales), la *maniobra* (i.e. decisiones durante la conducción de más corto plazo) y *control* (i.e. comportamientos automáticos). (TODO!Rehacer la figura nosotros, esta no me gusta. Es del [Michon, 1985]). 69

31. Las diferentes tareas para modelar el comportamiento de un conductor al volante. Están clasificadas en dos tipos, de aceleración, encargadas de definir cómo acelera el conductor en casos con y sin vehículo delantero y de lane-changing, encargadas de las tareas relacionadas con el cambio de carril. 70

32. Estructura del modelo de comportamiento de los vehículos en [Toledo et al., 2007]. El agente comprueba constantemente se desea o no cambiar de carril, y una vez decidido comprueba la viabilidad. En cualquier caso, toda decisión finaliza con la comprobación de la aceleración. Fuente: [Toledo et al., 2007]. TODO!Quizá la imagen es muy grande y quedaría mejor si la hiciésemos más pequeña. 75

Índice de cuadros

1. Línea temporal de los principales hitos en la AI. Actualmente la AI está ofreciendo resultados muy prometedores áreas como la conducción autónoma, el procesamiento del lenguaje natural o el análisis de sentimiento entre muchos otros. 23
2. Tabla comparativa donde se contrastan las características de los simuladores seleccionados. El simulador SUMO es el que más características cumple de las originalmente planteadas. 64

Parte I

Introducción

Introducción

Es un hecho que la **Inteligencia Artificial (AI, Artificial Intelligence)** en general (y la **Inteligencia Computacional (CI, Computational Intelligence)** en particular) como área de conocimiento ha experimentado un creciente interés en los últimos años. Esto no siempre ha sido así, ya que después de un nacimiento muy esperanzador, con mucho optimismo, le siguieron unas épocas de apenas avance (ver cuadro 1). Sin embargo, en la actualidad es muy difícil encontrar un campo que no se beneficie directamente de sus técnicas.

Una de sus razones es su carácter multidisciplinar ya que, si bien se la define como área perteneciente al campo de la informática, es transversal a muy diferentes campos, como pueden ser por ejemplo la biología, neurología o la psicología, entre otros.

Dentro del área de la **Inteligencia Artificial** es común diferenciar dos tipos de aproximaciones a la hora de hablar de cómo representar el conocimiento: la **Inteligencia Artificial clásica**, que postula que el conocimiento como tal se puede reducir a un conjunto de símbolos con operadores para su manipulación, y la **Inteligencia Computacional**, que defiende que al conocimiento se llega a través del aprendizaje, y que basa sus esfuerzos en la simulación de elementos de bajo nivel que subyacen a los comportamientos inteligentes esperando que el conocimiento “emerja” de éstos.

El límite entre ambos conjuntos no está perfectamente definido, máxime si tenemos en cuenta las diferentes terminologías existentes, las sinergias entre distintas técnicas dentro del área y los diferentes puntos de vista sobre éstas por parte de los autores. Sin embargo, una de las principales diferencias de ambos paradigmas es el punto de vista a la hora de solucionar problemas, siendo la aproximación **top-down** la usada en problemas de **AI** clásica y la **bottom-up** la típica usada en la **CI**. Revisaremos las diferencias entre conceptos de diferentes autores en el capítulo II¹.

Uno de los campos de aplicación es el de los **Sistemas Inteligentes de Transporte (ITSs, Intelligent Transport Systems)**. Éstos se definen como un conjunto de aplicaciones orientadas a gestionar el transporte en todos sus aspectos y granularidades (e.g. conducción eficiente, diseño de automóviles, gestión del tráfico o señalización en redes de carreteras) para hacerlos más eficientes y seguros. El interés es tal que

Cuadro 1: Línea temporal de los principales hitos en la AI. Actualmente la AI está ofreciendo resultados muy prometedores áreas como la conducción autónoma, el procesamiento del lenguaje natural o el análisis de sentimiento entre muchos otros.

1956	Conferencia de Dartmouth. Nace el campo de la AI. Se destinan muchos recursos debido al potencial del nuevo campo..
1974	No llegan los resultados esperados. Se suspenden financiaciones y se deja de investigar en muchas áreas (AI Winter).
1980	Aparecen los sistemas expertos. Muy prometedores. Acaparan la práctica totalidad de la investigación en AI..
1987	De nuevo los resultados no son lo esperado y vuelve a disminuir el trabajo en AI..
1990	La mejora de las prestaciones, la ubicuidad de los ordenadores y nuevos conceptos (e.g. agentes) hacen que la investigación en el área vuelva a crecer. Se replantea el concepto de AI..
2000	Se retoman investigaciones relacionadas con el aprendizaje profundo. Aumenta la investigación en el área de las redes neuronales y redes bayesianas..
2006	Conferencia de Dartmouth. Se analizan los avances y se debate sobre la AI a 50 años vista. Crece la expectación y el interés en el campo.
2007	Crece el interés en el aprendizaje automático debido a los resultados obtenidos y el aumento de las fuentes de datos.

¹ Una aproximación *top-down* a los problemas funciona definiendo primero el algoritmo que resuelve el problema para posteriormente ejecutarlo y llegar así a la solución exacta. Por otro lado, una aproximación *bottom-up* el algoritmo de resolución no se programa, sino que se aprende, llegando él sólo a soluciones no necesariamente exactas pero sí lo suficientemente buenas para ser aceptadas.

² En esta directiva, los ITS se definen como *aplicaciones avanzadas que, sin incluir la inteligencia como tal, proporcionan servicios innovadores en relación con los diferentes modos de transporte y la gestión del tráfico y permiten a los distintos usuarios estar mejor informados y hacer un uso más seguro, más coordinado y «más inteligente» de las redes de transporte.*

en el año 2010 se publicó la directiva 2010/40/UE (ver [par, 2010]) donde se estableció el marco de implantación de los ITS en la Unión Europea².

En el caso concreto del comportamiento al volante, es interesante la evaluación de los conductores para conocer su manera de actuar en determinados escenarios, y poder extraer información de éstos que nos permitan, por ejemplo, detectar qué factores pueden afectar más o menos a determinados indicadores (e.g. el consumo estimado para una ruta en concreto). Sin embargo, la evaluación en distintos escenarios puede no ser interesante debido a limitaciones existentes, como pueden ser, por ejemplo, el tiempo, el dinero o la peligrosidad del escenario.

Los simuladores de tráfico son una solución para muchas de estas limitaciones, pero suelen basar su funcionamiento en conductores y vehículos (normalmente concebidos como una única entidad) basándose en modelos de conductor que responden a funciones más o menos complejas, además con pocas o ningunas opciones de personalización. Esto provoca que dichos modelos se adapten poco al modelo de un conductor en concreto.

Esta tesis pretende explorar el tema de la generación de modelos de conductor para simuladores que respondan al comportamiento de conductores reales usando, para ello, técnicas pertenecientes al campo de la CI.

Concretamente pretende desarrollar un método para el análisis de la eficiencia de los conductores realizando, para ello, un modelo del perfil de conducción a partir de técnicas de la CI y aplicándolo a un entorno de simulación basado en *Sistemas Multiagente (MASs, Multi-Agent Systems)*. Así, una vez configurado el entorno, se podrán estudiar aspectos generales como la evolución del tráfico con determinados perfiles o particulares como el estilo de conducción o el impacto de los sistemas de asistencia.

Motivación

Los conceptos introducidos al comienzo del capítulo obedecen a una *necesidad* (aquí como eufemismo de problema) de la sociedad en la que vivimos, y que afecta tanto a nuestra generación como afectará a las venideras: la eficiencia en la conducción. Dado que es imprescindible saber que existe un problema para arreglarlo, nada mejor que puntualizar algunos hechos de sobra conocidos:

- En el año 2014, el número de vehículos a nivel mundial asciende a más de 1,200 millones, con una tendencia creciente [OICA, 2015]. Reducir en un pequeño porcentaje el consumo durante la conducción evita la emisión de toneladas de gases considerados nocivos para el medio ambiente y el ser humano³.

³ Uno puede argumentar que el parque automovilístico se recicla con nuevos vehículos eléctricos categorizados “de consumo 0”. La triste realidad es que estos vehículos consumen la electricidad generada actualmente de una mayoría de centrales de combustibles fósiles y nucleares. Además, mientras que en países desarrollados el crecimiento ha sido en torno al 4-7 %, en países subdesarrollados, donde no existe aun infraestructura para la recarga de vehículos eléctricos, dicho crecimiento ha superado el 120 %.

- Debemos abandonar los combustibles fósiles antes de que éstos nos abandonen a nosotros. Si bien es cierto que existen diferentes puntos de vista acerca de cuándo se agotarán las reservas de petróleo, también lo es que los combustibles fósiles son recursos **finitos**. Lo más probable es que no se llegue a agotar debido a la ley de la oferta y la demanda, pero hay que recordar que el petróleo se usa como base para la producción de otros muchos tipos de productos, como por ejemplo la vaselina, el asfalto o los plásticos.
- Independientemente del momento en el que se agoten los recursos, hay que hacer notar que la emisión de gases está correlacionada con el aumento de la temperatura del planeta, hecho que se ilustra en la figura 1. De seguir con el ritmo de consumo actual, se teme llegar a un punto de no retorno con consecuencias catastróficas para la vida en el planeta.
- Algo más cercano en el tiempo. La conducción eficiente afecta directamente a factores correlacionados con el número de accidentes de tráfico. Un factor de sobra conocido es el de la velocidad, factor relacionado no sólo con el número sino con la gravedad de los accidentes [Imprialou et al., 2016]. Otros indicadores son las aceleraciones, deceleraciones y maniobras de cambio de dirección, cuya frecuencia es inversamente proporcional a la eficiencia en la conducción y directamente proporcional a la agresividad, falta de seguridad y accidentes ([Dingus et al., 2006] y [Lerner et al., 2010]).

Estos hechos son solo algunos que ponen de manifiesto la necesidad de centrarse en el problema de cómo hacer de la conducción una actividad más eficiente y segura.

La **conducción eficiente** o *eco-driving* es definida como la aplicación de una serie de reglas de conducción con el objetivo de reducir el consumo de combustible, independientemente del tipo (e.g. electricidad, gasolina, gas natural, ...).

Ser capaces de identificar o al menos estimar qué conductores son considerados como no eficientes es importante debido a que, de esta manera, se pueden identificar los hábitos recurrentes en este tipo de perfil y adecuar la formación para eliminar dichos hábitos. Más aún teniendo en cuenta la relación existente entre la peligrosidad y algunas conductas agresivas. Un ejemplo donde la identificación de perfiles no eficientes pueden tener impacto claro económico y social es el de las empresas cuya actividad se basa en el transporte de mercancías o de personas.

Sin embargo, identificar la conducta de un conductor no es fácil, dado que su comportamiento se ve condicionado por numerosos factores como el estado de la ruta, el del tráfico o el estado físico o anímico. Además, la ambigüedad de las situaciones dificulta todavía más la identificación. Por ejemplo, un conductor puede ser clasificado en un momento como agresivo o no eficiente en una situación

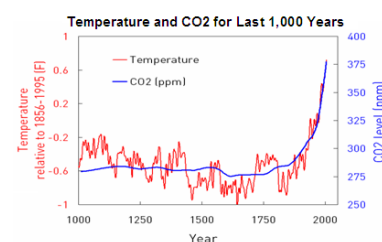


Figura 1: Desde el comienzo de la revolución industrial, el uso masivo de combustibles fósiles y el crecimiento de la población propició un aumento desproporcionado de CO₂ a la atmósfera, tendencia que sigue en aumento aún con la (lenta) adopción del vehículo eléctrico. La gráfica muestra cómo ambos valores parecen estar correlacionados. Fuente: Environmental Defense Fund (edf.org).

únicamente porque su comportamiento ha sido condicionado por las malas reacciones por parte de los demás conductores.

El análisis de todos los posibles casos es una tarea prácticamente imposible. Por ello, las simulaciones pueden dar una estimación de los posibles resultados de un estudio en el mundo real. Las simulaciones con **MASs** representan a los conductores como agentes permitiendo la evaluación del comportamiento tanto individual como general del sistema en base a sus individuos a través de iteraciones discretas de tiempo.

Si dichos agentes son obtenidos mediante la modelización de conductores a partir de sus datos reales, su comportamiento en la simulación podría ser considerado como fuente de datos para condiciones de tráfico y/o rutas no contempladas en el mundo real. De esta forma, se dispondría de un marco de trabajo para la comparación de diferentes conductores sin necesidad de exponerlos a todos y cada uno de los posibles eventos posibles. También sería factible evaluar sistemas de asistencia evitando los problemas de no comparabilidad de condiciones del entorno entre pruebas.

Demostrar que la evaluación de un modelo del conductor en entornos simulados es equivalente a la evaluación de conductores en entornos reales implica que se pueden comparar dos conductores usando un criterio objetivo, es decir, sin depender del estado del resto de factores a la hora de realizar la prueba de campo. Dicho de otro modo, implicaría que es posible comparar la eficiencia de dos conductores independientemente del estado del tráfico e, incluso, sobre rutas diferentes.

Objetivos

El objetivo de esta tesis doctoral es la de demostrar la hipótesis 1, quedando dicha demostración dentro de los límites impuestos por los supuestos y restricciones indicados más adelante.

Hipótesis 1 (H1): *La aplicación de técnicas pertenecientes al campo de la CI con datos extraídos de un entorno de microsimulación de espacio continuo y tiempo discreto basado en sistemas multiagentes permitirá modelar, de manera fiel a la realidad, el comportamiento de conductores reales.*

Por tanto, el objetivo de la tesis es el de simular el comportamiento de conductores en entornos de micro-simulación a partir de su comportamiento en entornos reales usando técnicas de **CI**. Para ello se consideran los siguientes objetivos específicos:

- Estudiar y aplicar técnicas de la **CI** sobre el área de la conducción.
- Realizar un **Estudio Naturalista de Conducción (NDS, Naturalistic Driving Study)**⁴ sobre conductores reales para:

⁴ Los **NDS** basan su funcionamiento en la captura masiva de datos de conducción, normalmente involucrando una gran cantidad de sensores, para analizar el comportamiento del conductor, las características del vehículo, la vía, etcétera. La cantidad de sensores y la velocidad de captura hacen que la tarea de analizar y extraer conclusiones sea una tarea prácticamente imposible para un humano, por lo que es necesario el uso de técnicas de análisis de datos que suelen recaer en los campos de la estadística y del aprendizaje automático.

1. Generar modelos personalizados de conductor a partir de los datos de conducción obtenidos.
 2. Aplicar modelos de conductores a entornos de simulación multiagente.
 3. Validar los modelos de conductor contra conductores reales.
- Estudiar la efectividad de sistemas de asistencia encaminados a mejorar la eficiencia y analizar el comportamiento de conductor.

Supuestos

- Se supone que el comportamiento de un conductor es el comportamiento en línea y el comportamiento de cambio de carril⁵.
- Los datos de los que extraer el comportamiento se corresponderán con lecturas realizadas durante el día, con buena visibilidad y sin lluvia.
- El tipo de vehículo sobre el que modelar el comportamiento será el de un utilitario.
- El conductor a modelar pertenecerá al grupo más representativo de conductores. Esto se corresponde con varón de 35 a 39 años (ver figura 2).

⁵ Son conocidos en la literatura como *car-following* y *lane-changing* respectivamente. Entraremos en detalle sobre ambos conceptos en el capítulo [Modelos de comportamiento](#)

Restricciones

- El sistema multiagente hará uso de [Driver-Vehicle Units \(DVUs\)](#) como agentes, es decir, usando la tupla (conductor, vehículo) como un todo.
- La resolución máxima del modelo creado es de 1Hz.
- En el caso de los modelos que hacen uso de redes neuronales artificiales, no se pueden explicar las razones del comportamiento inferido.

Estructura de la tesis

La tesis está estructurada de la siguiente manera:

En los capítulos [I](#), [II](#) y [III](#) se expone la revisión realizada del estado de la cuestión donde se explica en qué punto se encuentra la literatura de los temas en los que se apoya la presente tesis.

En el capítulo [XXX](#) se explica el método seguido para la confirmación de la hipótesis describiendo además las instrumentaciones, los conjuntos de datos obtenidos, las técnicas utilizadas y las aplicaciones desarrolladas.

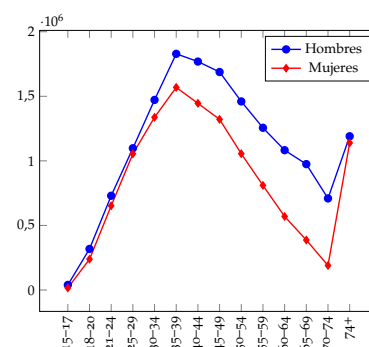


Figura 2: Último censo de conductores según género segmentado por edades. Fuente: Dirección General de Tráfico ([dgt.es](#)).

Por último en el capítulo III se exponen los resultados y las conclusiones extraídas de la tesis. Además, tras las conclusiones se indican una serie de posibles líneas futuras de trabajo consideradas interesantes tras la realización de la tesis.

Parte II

Estado de la cuestión

Inteligencia Computacional

El comportamiento de un individuo en un entorno se ve influenciado por una infinidad de variables. Identificar las relaciones entre éstas es en la mayoría de las ocasiones una tarea que va de lo muy difícil a lo imposible, más aún si añadimos que éstas son muy numerosas y pueden llegar a ser imposibles de cuantificar o incluso de detectar.

La CI engloba un conjunto de técnicas que facilitan enormemente estas tareas. En este capítulo se ofrece una perspectiva de la literatura actual sobre las técnicas de la CI que son de interés para esta tesis. Introduciremos el concepto y las nociones de “agente” y de “aprendizaje” para posteriormente introducir algunas de las técnicas utilizadas dentro del área. Por último, desarrollaremos las tres técnicas principales sobre las que reposa el trabajo teórico de esta tesis: Redes Neuronales Artificiales, Logica Difusa y Computación Evolutiva.

Inteligencia Artificial vs. Inteligencia Computacional

¿Qué es la CI? Para entender el significado de éste término tenemos que entender cómo ha evolucionado el término AI a lo largo de los años.

El primer concepto a introducir es el de “conexionismo”. Se puede considerar a Santiago Ramón y Cajal como principal precursor de esta idea por sus trabajos acerca de la estructura de las neuronas y sus conexiones (e.g. [y Cajal, 1888] y [Ramón and Cajal, 1904]). Otros prefieren citar el trabajo “*A logical calculus of the ideas immanent in nervous activity*” ([McCulloch and Pitts, 1943]) sobre **Redes Neuronales Artificiales (ANNs, Artificial Neural Networks)** o “*The organization of behavior*” ([Hebb, 1968]) acerca de la teoría del aprendizaje como primeros trabajos en este tema. Independientemente de su origen, el conexionismo postula que la mente y el conocimiento surgen de redes formadas por unidades sencillas interconectadas (i.e. neuronas).

Por otro lado, en 1950, Alan Turing publicó un artículo que comenzaba con la frase “*Can machines think?*” [Turing, 1950], introduciendo el famoso Test de Turing para determinar si una máquina es

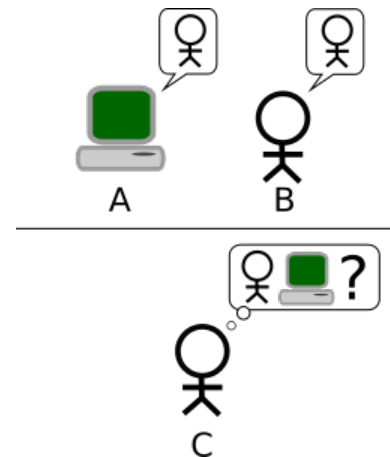


Figura 3: Ilustración del Test de Turing. Modelo propuesto para probar si una máquina es capaz de exhibir comportamiento inteligente similar al del ser humano. Hay tres participantes, dos humanos (A y C) y una máquina (B), separados entre sí pero pudiendo intercambiarse mensajes de texto. C envía preguntas a A y B sin saber quién es humano y quién es máquina y éstos le responden. Si C no es capaz de identificar qué participante es la máquina, se puede concluir que la máquina es inteligente. Fuente: Hugo Féréé, via Wikimedia Commons.

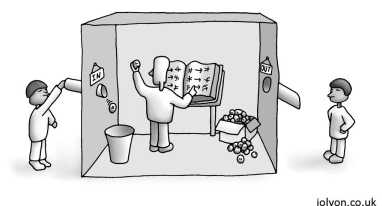


Figura 4: “La habitación china” es un experimento mental propuesto por John Searle que parte de un Test de Turing donde la máquina ha aprendido a hablar chino. Ésta es reemplazada por una persona que no sabe nada del idioma pero que va equipada con manual de correspondencias (ideograma de entrada → ideograma de salida). Cuando una persona le manda mensajes en chino, esta otra responde. ¿Podemos decir que dicha persona sabe chino? Evidentemente no. Pero entonces, ¿cómo podemos asegurar que la máquina reemplazada ha “aprendido” chino?. Autor: Jolyon Troscianko (<http://www.jolyon.co.uk/>)

⁶ El propio concepto de “pensar” es en sí un tema controvertido en el propio ser humano: ¿pensar es algo inherentemente biológico? ¿surge de la mente? Tanto si sí como si no, ¿de qué forma lo hace? Por ello existen detractores de la validez del Test de Turing. Por ejemplo, el experimento de la habitación china (figura 4) nace precisamente como refutación de dicho test, aunque puede llevar a cuestiones quizá más intrigantes. Por ejemplo, si la máquina es capaz de realizar una acción sin entender lo que hace y por qué lo hace, ¿qué garantías tenemos de que el humano sí es capaz? Si los ordenadores operan sobre símbolos sin comprender el verdadero contenido de éstos, ¿hasta qué punto los humanos lo hacen de forma diferente?.

⁷ Es injusto achacar el *AI Winter* sólo al libro *Perceptrons*. El “efecto gurú” del libro fue sólo la gota que colmó el vaso. A la emoción inicial por los avances le siguieron muchos años de promesas incumplidas, investigación sin resultados significativos, limitaciones de hardware, aumento de la complejidad del software (los comienzos de la crisis del software. Ver [Dijkstra, 1972]). Todo ello provocó un desinterés y una disminución de la financiación que se retroalimentaron la una a la otra.

⁸ A esta década se la conoce como segundo *AI Winter* dado que la investigación sobre Sistemas Expertos disminuye. Sin embargo no fue un abandono tan acusado como el del primer *AI Winter*.

⁹ Es comprensible ya que el método clásico produce modelos fáciles de interpretar mientras que el enfoque conexionista produce modelos cuyo funcionamiento en general no es del todo deducible. Sin embargo, existen problemas con alto grado de complejidad muy difíciles (o imposibles) de modelar. Más aún cuando éstos son de naturaleza estocástica [Siddique and Adeli, 2013].

o no inteligente (figura 3). Se puede considerar este momento como el punto donde se estableció el objetivo a largo plazo del campo de la *AI*, ya que en el artículo Turing propuso un método para determinar si una máquina era capaz de exhibir comportamiento inteligente. Sin embargo, no fue hasta 1956 en la Conferencia de Dartmouth [McCarthy et al., 1956] donde John McCarthy acuñó el término *AI* a la vez que presentó el tema de la conferencia como la pregunta realizada por Turing en dicho artículo.

A partir de este punto la investigación en *AI* recibió muchísima atención por parte de investigadores y gobiernos, lo que se tradujo en financiación. Los estudios estaban dominados por aquellos relacionados con las ideas del conexionismo hasta que en 1969, se publicó el libro *Perceptrons* [Minsky and Papert, 1969] de Marvin Minsky y Seymour Papert, donde se expusieron las limitaciones de los modelos de *ANNs* desarrollados hasta la fecha. El impacto fue tal que la investigación en *AI* se abandonó casi por completo. Concretamente el conexionismo dejó de estar presente en la literatura científica durante dos décadas. Es lo que se conoce como *AI Winter*⁷.

El interés por el campo volvió de nuevo a principios de 1980 con la aparición en escena de los primeros Sistemas Expertos, los cuales se consideran como el primer caso de éxito en la *AI* ([Russell et al., 2003]). A finales de la década, sin embargo, empezaron a resurgir los enfoques conexionistas, en gran parte por la aparición de nuevas técnicas de entrenamiento en perceptrones multicapa o por conceptos como activación no lineal en neuronas (e.g. [Rumelhart et al., 1985] o [Cybenko, 1989]). En este momento los sistemas expertos empezaron a perder interés frente al nuevo avance del conexionismo⁸.

Mientras que el enfoque clásico de la *AI* postulaba que la mente operaba de la misma manera que una máquina de Turing, es decir, mediante operaciones sobre un lenguaje de símbolos, el enfoque del conexionismo postulaba que la mente, el comportamiento inteligente, emergía de modelos a más bajo nivel. Esto provocó que algunas voces se alzaran contra lo que se consideraba el “enfoque incorrecto” de la *AI*. Sin embargo, otras técnicas alineadas con el conexionismo (debido a su enfoque de comportamiento emergente y aproximación como lo son la *Logica Difusa* (FL, Fuzzy Logic) o los *Algoritmos Genéticos* (GAs, Genetic Algorithms)) ganaban popularidad y alimentaban el éxito cosechado por este “enfoque incorrecto”⁹.

Esto provocó una explosión de terminologías para diferenciar las investigaciones de la propia *AI* clásica. Por un lado se evitaba el conflicto, nombrando las áreas de trabajo con un término más acorde con el comportamiento o técnica utilizada. Por otro, se separaba de las connotaciones negativas que fue cosechando la *AI* con el paso de los años, “promesas, pero no resultados”).

Lo verdaderamente interesante es ver la evolución de la literatura, y por tanto de los objetivos de la *AI* durante estos años. En el

nacimiento del campo, se buscan literalmente máquinas que piensen como humanos, o al menos seres racionales, con mente. Con el paso de los años (y los continuos choques contra la realidad), la literatura va tendiendo hacia la búsqueda de conductas y comportamientos inteligentes cada vez más específicos. Este hecho se hace más patente en este momento, donde cada investigación se nombra de cualquier forma menos con el término **AI** (e.g. **Aprendizaje Automático (ML, Machine Learning)**, **Sistemas de Recomendación (RSs, Recommender Systems)**, o **Procesamiento de Lenguaje Natural (NLP, Natural Language Processing)**). Es evidente que la **AI** se puede observar desde diferentes puntos de vista, todos perfectamente válidos. En [Russell et al., 2003], tras un análisis de las definiciones existentes en la literatura por parte de diferentes autores, se hace énfasis en este hecho mostrando los diferentes puntos de vista a la hora de hablar de lo que es la **AI**. El resumen se puede observar en la figura 5.

Thinking Humanly "The exciting new effort to make computers think ... <i>machines with minds</i> , in the full and literal sense." (Haugeland, 1985) "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning .. ." (Hellman, 1978)	Thinking Rationally "The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985) "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Acting Humanly "The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990) "The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	Acting Rationally "Computational Intelligence is the study of the design of intelligent agents." (Poole et al, 1998) "AI ... is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

Figura 5: Diferentes objetivos perseguidos por la Inteligencia Artificial. Las filas diferencian entre pensamiento o comportamiento mientras que las columnas separan entre inteligencia humana o el ideal de la inteligencia (racionalidad). Fuente: *Artificial Intelligence: A Modern Approach* (3rd Ed.), [Russell et al., 2003].

Volviendo al tema de la terminología, muchas de las diferentes técnicas se fueron agrupando dentro de diferentes áreas. Una de ellas es la conocida como Computational Intelligence. Dado que persigue el mismo objetivo a largo plazo y que surge de la propia **AI** parece lógico mantenerla como un subconjunto y no como un nuevo campo del conocimiento humano. Sin embargo, algunos autores abogan por que la **CI** es un campo diferenciado de la **AI**.

Podemos definir la **CI** como la "rama de la **AI** que aporta soluciones a tareas específicas de forma inteligente a partir del aprendizaje mediante el uso de *datos experimentales*". A diferencia de la aproximación clásica de la **AI**, se buscan aproximaciones a las soluciones y no las soluciones exactas. Esto es debido a que muchos problemas son de naturaleza compleja, ya sea por la relación entre sus múltiples variables, a la falta de información o a la imposibilidad de traducirlos a lenguaje binario.

Se puede fijar el año 1994 como el que el término **CI** nace como área, coincidiendo con el cambio de nombre del *IEEE Neural Networks*

Council a IEEE Computational Intelligence Society (<http://cis.ieee.org/>). Poco antes, en 1993, Bob Marks en su trabajo [Bezdek, 1993] presentó las que él consideraba diferencias fundamentales entre la AI y la CI resumiéndolas en la siguiente frase.

“Neural networks, genetic algorithms, fuzzy systems, evolutionary programming, and artificial life are the building blocks of CI.”

Durante estos años ganaba popularidad también el concepto del **Soft Computing (SC)**. Éste engloba las técnicas que buscan resolver problemas con información incompleta o con ruido. Debido a que el conjunto de técnicas definidas como consituyentes del SC son las mismas que las de la CI algunos autores consideran ambos términos equivalentes. Nosotros consideramos que el SC es un punto de vista de la computación más que de la AI en contraposición con el **Hard Computing (HC)**, y que la CI hace uso de métodos del SC¹⁰.

¹⁰ El HC es como se define la computación convencional frente al SC. El HC basa sus técnicas en aquellas basadas en modelos analíticos definidos de forma precisa y que en ocasiones requieren mucho tiempo de cómputo. Están basados en lógica binaria, análisis numérico, algoritmos y respuestas exactas. El SC por otro lado es tolerante a la imprecisión y al ruido y tiende a llegar a soluciones aproximadas de manera más rápida. Se basa en modelos aproximados, emergencia de algoritmos y modelos estocásticos.

Aprendizaje

La resolución clásica a un problema suele ser la aplicación de una secuencia de instrucciones basadas en un conjunto de símbolos (e.g. una función escrita en el lenguaje de programación C). Esta forma de solucionar un problema no *aprende* a solucionarlo. Se puede interpretar como que la solución está grabada en su memoria.

En la aproximación de la CI, existen modelos y existen técnicas para hacer aprender esos modelos. La aplicación de dichas técnicas es lo que se conoce como **aprendizaje**. Las técnicas de aprendizaje en CI se suelen clasificar en 2 paradigmas:

- **Supervisado.** El entorno presentado al modelo consiste en un conjunto de la forma $D = (I_i, O_i) | \forall i \in \mathbb{N}$, donde cada O_i es la salida esperada del modelo a la entrada I_i . Los algoritmos tratarán de ajustar el modelo todo lo posible para que las salidas obtenidas sean lo más parecidas a las salidas originales. Este paradigma de entrenamiento suele estar relacionado con problemas de *regresión*.
- **No supervisado.** Al modelo se le ofrece un conjunto de la forma $D = I_i | \forall i \in \mathbb{N}$, donde cada I_i es una entrada al problema, pero del que no conocemos la salida. Los algoritmos dentro de esta categoría harán uso de estos datos para ir reajustando el modelo tratando de encontrar las estructuras ocultas entre dichos datos (e.g. patrones, correlaciones o clases). Es un paradigma de entrenamiento íntimamente relacionado con problemas de *clasificación*.

Algunos autores hacen uso de técnicas pertenecientes a ambos paradigmas en forma de aproximación híbrida para suplir deficiencias u optimizar/acelerar el aprendizaje. Un claro ejemplo lo podemos ver en [Hinton, 2006], donde los autores hacen uso de *autoencoders* como técnica no supervisada para la inicialización de los pesos de

una red neuronal y posteriormente realizan un entrenamiento supervisado para la optimización es éstos.

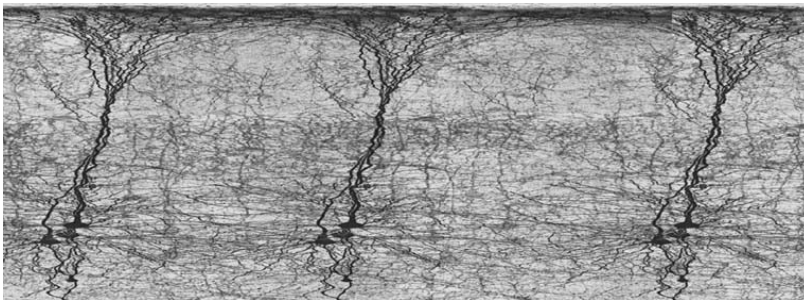
Otros, añaden un paradigma más a estos dos existentes, el denominado aprendizaje **por refuerzo**. Sin embargo, nosotros preferimos considerarlo como un tipo de aprendizaje supervisado, ya que la única característica que lo diferencia es que es un tipo de aprendizaje que se usa en entornos de aprendizaje *on line* ajustando el modelo en función de los estímulos percibidos del entorno por sus acciones sobre el mismo, y no en un entorno aislado previo de entrenamiento (*off line*), como la práctica totalidad de técnicas supervisadas y no supervisadas.

Técnicas en la Inteligencia Computacional

Bajo el paraguas de la **CI** se incluyen muchas técnicas diferentes, entre las cuales están las usadas en esta tesis. El resto de la sección describe el funcionamiento de cada una de estas técnicas.

Redes Neuronales Artificiales

Una **Red Neuronal Artificial** puede considerarse como una herramienta que trata de replicar las funciones cerebrales de un ser vivo de una manera muy fundamental (esto es, desde sus componentes más básicos, las neuronas) basándose para ello en estudios de neurobiología y de ciencia cognitiva moderna del cerebro humano¹¹.



Una **ANN** es independiente del modelo del problema a solucionar. Se la puede considerar como una caja negra que aprende las relaciones que subyacen en los datos del problema para abstraer el modelo a partir de éstas. Estas características de aprendizaje y abstracción son los factores determinantes por los que son usadas en prácticamente todas las áreas de la ciencia y de la ingeniería ([Du and Swamy, 2006]).

EL MODELO DE NEURONA ARTIFICIAL de McCulloch-Pitts introducido en el trabajo "A logical calculus of the ideas immanent in nervous

¹¹ Aún apoyándose en la topología y funcionamiento del cerebro humano para realizar el símil, lo cierto es que dichos modelos distan aún de considerarse *cerebros artificiales*. La red neuronal más compleja hasta la fecha es la propuesta en [Pakkenberg and Gundersen, 1997]. Figura 6: Una sección del neocórtex humano, región asociada a las capacidades cognitivas y que supone alrededor de 160,000 parámetros a ser ajustados (podemos abstraernos y pensar en ellos como conexiones entre neuronas). Como dato anecdótico, se estima que solo en el neocórtex (ver figura 6) del ser humano existen alrededor de 10,000 neuronas y un diámetro de 0.5mm. Fuente: *Blue Brain Project*, EPFL, <http://bluebrain.epfl.ch/>, entre 100 y 100,000 neuronas vecinas ([Pakkenberg and Gundersen, 1997]). Esto supone entre $2 \cdot 10^{12}$ y $2 \cdot 10^{15}$ conexiones. Tecnológicamente hablando, la sensación es que estamos aún a años luz de aproximarnos siquiera a la complejidad de un cerebro humano.

activity” ([McCulloch and Pitts, 1943], ilustración en la figura 7) es el considerado primer trabajo en la disciplina de las ANNs, y aunque existen muchas y muy diferentes tipologías y formas de operar con redes, todas funcionan de la misma manera: unidades (i.e. neuronas) interconectados mediante enlaces por los que fluye la información de manera unidireccional, donde algunas de dichas unidades sirven de entrada al sistema (i.e. entradas o sensores), otras sirven de salida del sistema (i.e. salidas y actuadores) y otras como elementos internos (i.e. ocultas), y donde las conexiones se ajustan mediante un proceso denominado *entrenamiento*.

Figura 7: Variación de la representación del modelo de neurona artificial propuesto por McCullochs y Pitts. En éste, cada una de las entradas x_i es incrementada o inhibida aplicando el producto con su peso asociado w_i . La activación vendrá determinada por la aplicación de una función (denominada “de activación”) a la suma de los valores. Esta variación en concreto incluye una entrada x_0 y un peso w_0 como bias de la neurona para la variación dinámica del umbral de activación.



Este primer modelo de neurona proponía una función escalón para determinar si la neurona se activaba o no (analogía al funcionamiento de la neurona artificial). Posteriormente aparecieron nuevos modelos de neuronas diferentes funciones de activación. De éstas, las más comunes son las de tipo sigmoide¹².

¹² Concretamente la función logística de pendiente 1 definida como:

$$\frac{1}{1 + e^{-x}} \quad (1)$$

Parrafito sobre la limitación de la neurona singular para dar hilo a las topologías

EXISTEN DIFERENTES TOPOLOGÍAS DE REDES NEURONALES O arquitecturas dependiendo de qué forma toma el grafo que modela las neuronas y sus conexiones. En este caso, las redes neuronales pueden ser de dos tipos:

- **Feed-Forward.** Sus grafos no contienen ningún ciclo (figura XXX). Es la topología más usada en aplicaciones prácticas debido a su sencillez y su efectividad. En ellas el flujo de información sigue un camino desde las entradas hasta las salidas, sin ninguna retroalimentación. No es requisito que las neuronas se agrupen en capas, aunque suele ser la estructura común. A las redes de más de dos capas ocultas (i.e. las capas que se encuentran entre la capa de neuronas de entrada y la capa de neuronas de salida) se las denomina “profundas” o *deep*. Algunos tipos pertenecientes a esta categoría pueden ser el Perceptrón [Rosemblat, 1957], el Perceptrón multicapa [Rumelhart et al., 1986], el algoritmo LVQ y su sucesor los Mapas Auto-Organizados [Kohonen, 1998].
- **Recurrentes.** Sus grafos contienen uno o más ciclos, de tal manera

que el flujo de información de salida de una neurona puede llegar a afectar a su propio estado. Estas topologías representan de una forma más fiel las bases biológicas de las ANN, pero son más complejas a la hora de operar y entrenar. Algunos casos particulares de este tipo de arquitectura son las Redes de Hopfield [Hopfield, 1982] o las memorias LSTM (del inglés Long-Short Term Memory) [Hochreiter & Schmidhuber, 1997].

Aprendizaje en Redes Neuronales Artificiales

Logica Difusa

La lógica matemática (y por extensión la teoría de conjuntos)¹³ tiene como misión servir de fundamento del razonamiento matemático. Se basa en la definición precisa y con rigor de un razonamiento evitando cualquier tipo de ambigüedad y de contradicción. Es por ello que la lógica tradicional no suele servir como fundamento de razonamientos del mundo real.

Los conceptos que se manejan en el mundo real suelen ser vagos, llenos de imprecisiones. Además tienden a ser nombrados cualitativamente, no cuantitativamente, y cuando existe una correspondencia, ésta suele estar marcada por la subjetividad de los términos.

Explicar lógica difusa y control difuso. Indicar los controladores difusos de segundo, tercer y sucesivos niveles.

Teoría de conjuntos difusos

A diferencia de los conjuntos tradicionales, los conjuntos difusos expresan el grado de pertenencia de un elemento a la categoría representada por el conjunto. La definición podría escribirse de la siguiente manera:

TODO!Creo que habría que definir antes qué es un dominio

Definición 1: Sea X una colección de elementos. Se define al **conjunto difuso** F como un conjunto ordenado de pares de la forma $F = (x, \mu_F(x)) | x \in X$, siendo $\mu_F(x) \in [0, 1] \forall x \in X$.

La función de la definición 1 se denomina **función de pertenencia**, y caracteriza unívocamente a un conjunto difuso del dominio de X .

TODO!Quizá aquí habría que decir qué es una partición de un dominio

¹³ Se puede establecer el siglo IV a.C. como el momento del nacimiento de la lógica dentro de la física Aristotélica, que permaneció inalterada hasta los trabajos de Galileo (cita?) y Newton (cita, seguramente el Principia Mathematica) en el siglo XVI. d.C., momento en que se separó y permaneció como disciplina paralela perteneciente más al campo de la filosofía que de la física y la matemática. Empezó a relacionarse de nuevo con la matemática a principios del siglo XIX y a principios del siglo XX la lógica y la teoría de conjuntos pasaron a convertirse en partes indispensables la una de la otra. Por ello suelen ir de la mano cada vez que se habla de la una y de la otra. La evolución de la teoría de conjuntos (Cantor, finales del siglo XIX, buscar referencia) y su unión con la lógica es una época bastante convulsa dentro de la historia de la matemática pero esta tesis no es lugar para su desarrollo. Por ello se ofrece la referencia al libro *La historia de la matemática* de Miguel de Guzmán (referencia. Poner la página. Lo mismo no es de aquí y es del libro de "Los lógicos", pero estoy casi seguro de que es aquí.) en caso de que el lector tenga interés en el tema.

Operaciones entre conjuntos

La unión, intersección y el complemento son operaciones básicas en la teoría de conjuntos. **TODO!** hablar aquí de t_{norm} , t_{conorm} y complemento, pero someramente. No hay que enrollarse demasiado.

Razonamiento

En [?] hay un capítulo de razonamiento que parece que está guay. Revisarlo un poco a fondo a ver si merece la pena tirar or ahí.

Sistemas de inferencia difusa

Los **Sistemas de Inferencia Difusa (FISs, Fuzzy Inference systems)** o **Sistemas de Control Difuso (FCSs, Fuzzy Control systems)** son sistemas que utilizan el razonamiento difuso para inferir una respuesta a partir de un conjunto de entradas. Habitualmente son divididos en tres bloques conceptuales, los cuales se ilustran en la figura XXX:

- **Fuzzificación.** Traducir los valores de entrada en crudo del dominio sobre el que está definida cada variable lingüística a sus respectivos grados de pertenencia a conjuntos difusos a través de sus funciones de pertenencia. **TODO!** Ojo, algunos controladores toman como valores de entrada conjuntos difusos según [?]. Habrá que buscar sobre ello.
- **Inferencia.** Realiza todo el proceso de razonamiento difuso a partir del conjunto de reglas que dan significado a este controlador difuso.
- **Defuzzificación.** Traduce los conjunto difuso resultado del proceso de inferencia a valores de los dominios sobre los que están definidos dichos conjuntos difusos. **TODO!** En un Sugeno, la salida es una función directamente así que se podría especificar que en un tipo Sugeno, se puede ver como que la salida son sólo singletons, manteniendo la generalización del proceso de funcionamiento de un FIS.

Hablar someramente de los tres tipos clásicos que se usan, e indicar que al final los más usados son el Mamdani y el Sugeno. Añadir también quizá una tabla comparativa entre los tres o al menos entre los dos principales:

El consecuente de un FIS de tipo Mamdani siempre es un conjunto difuso. Por tanto, el proceso de sacar un valor crisp es costoso. Lo bueno, se mantiene significado semántico de las salidas. El consecuente en un Sugeno es un valor, y se puede decir que no necesita

proceso de defuzzificación. Si embargo, la respuesta pierde significado semántico si la suma de la fuerza de salida no es 1 (no entiendo qué quiero decir con esto).

Computación Evolutiva

Otra de las técnicas inspiradas en principios biológicos es la de la **Computación Evolutiva (EC, Evolutionary Computation)**. Esta área de la **CI** trabaja sobre algoritmos iterativos inspirados en la o .

Existe un conjunto de algoritmos denominados **Inteligencia de enjambre (SI, Swarm Intelligence)** que también son algoritmos iterativos estocásticos inspirados en cómo se comportan sistemas colectivos donde el conocimiento emerge de la interacción de individuos, ya se trate de sistemas naturales como por ejemplo el de las bandadas de murciélagos ([Yang, 2010]) o de sistemas artificiales como sistemas de partículas abstractas ([Artyukhin and Avery, 2014]). Algunos autores consideran este conjunto de algoritmos como pertenecientes al subcampo de la **EC**, pero nosotros consideramos a la **SI** como un subcampo independiente dado que el adjetivo *evolutivo* no es característico de este tipo de algoritmos¹⁴.

Algoritmos Genéticos: bases y funcionamiento

Un **GA** es un algoritmo iterativo estocástico donde se aplica una serie de pasos inspirados en la para la resolución de un problema. Éstos fueron introducidos por John Henry Holland en su libro “*Adaptation in Natural and Artificial Systems*” ([Holland, 1975]) y constituyen una familia de soluciones con mucho éxito en problemas de aprendizaje, búsqueda y optimización. Desde nuestro punto de vista, podemos decir que la **EC** es el área que trabaja con **GA**¹⁵.

La o es una integración de teorías que complementan la teoría de la evolución propuesta por Charles Darwin. En el momento de enunciar la teoría, no se conocía de qué manera los organismos vivos transmitían sus rasgos característicos a su progenie. Las características en las que se inspiran los **GAs** de la síntesis evolutiva moderna incluyen, entre otras:

- El principio de selección natural. Introducido en la teoría de la evolución, el principio de selección natural indica que las características de un individuo en un entorno son las que facilitan (o dificultan) sus posibilidades de reproducción, y por tanto de posibilidad de transmisión de dichas características.
- La teoría genética, por Gregor Mendel. Concretamente los conceptos de gen (unidad de información genética que tiene posición y valor dentro de un genotipo), alelo (variación de valor de un

¹⁴ El punto de vista de los autores que lo incluyen es que la evolución no sólo ha sido la razón de la adaptación de los individuos, sino que también es la causante de comportamientos inteligentes grupales. Y, aunque es una apreciación cierta, el adjetivo *evolutivo* se refiere a cómo trabajan los algoritmos, no a lo que ha propiciado la evolución en la naturaleza.

¹⁵ Dentro de la **EC** existen familias de algoritmos consideradas por muchos autores como entidades diferentes a los **GAs**:

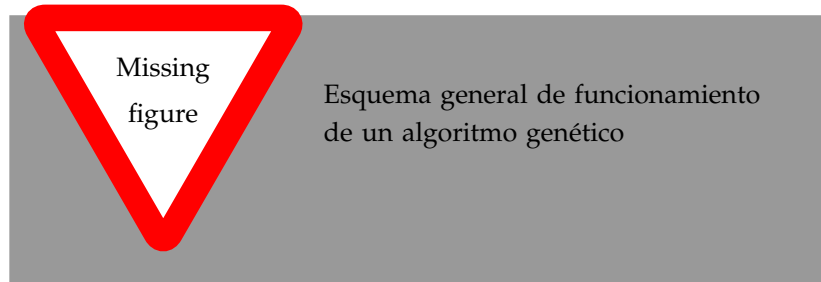
- **Estrategias Evolutivas (ESs, Evolution Strategies)**. Surgen casi a la par con los **GA** por parte de Ingo Rechenberg ([Rechenberg, 1973]). Se caracterizan principalmente por ser algoritmos iterativos inspirados en la evolución pero que basan su funcionamiento en los pasos de selección y mutación y donde el reemplazo es total.
- **Programación Genética (GP, Genetic Programming)**. Sus comienzos se pueden trazar hasta Koza 13, donde el autor hace uso de algoritmos genéticos para la generación de programas. La idea de esta aproximación es la de representar los genotipos de los individuos no como cadenas de símbolos (la representación clásica de genotipos) sino como árboles sintácticos.

Quizá en el momento de su nacimiento pudieran estar más diferenciadas, pero en la actualidad se puede pensar en ellas como diferentes especializaciones de un **GA** genérico. En el caso de las **ESs**, es evidente que se pueden considerar como una especialización de algoritmo genético donde la recombinación y el reemplazo parcial pasan a un segundo plano, y en el caso de la **GP**, son un **GA** una estructura de datos diferentes para el genotipo (la cual conlleva una serie de modificaciones a los algoritmos de recombinación y mutación, así como una serie de desventajas inherentes a su representación).

gen que da lugar a variaciones en el individuo, como por ejemplo color de ojos), genotipo (conjunto de genes que caracterizan a un individuo de una especie), fenotipo (las características físicas del individuo, determinadas por el genotipo y el entorno) y la herencia genética.

- La mutación como fuente de variación dentro de una población.
- El principio de diversidad alélica de la genética de poblaciones.

Figura 8: Esquema general del funcionamiento de un Genetic Algorithm. Se parte de una población de individuos y en iteraciones de selección-recombinación-mutación-reemplazo se va mejorando la población hasta llegar a una solución que, si bien no tiene por qué ser necesariamente la mejor, es lo suficientemente buena como para ser válida.



Basándose en estos conceptos, la estructura de un GA se asemeja al proceso de evolución de poblaciones finitas de individuos. Un ejemplo de esta estructura lo podemos ver en la figura 8. Los principales pasos dentro del algoritmo son los siguientes:

- **Inicialización.** El algoritmo crea una *población* de individuos donde cada uno representa una posible aproximación a la solución del problema a resolver.
- **Selección.** Al igual que en la naturaleza, donde el individuo más apto tiene la mayor probabilidad de sobrevivir y tener descendencia con sus características diferenciadoras, en un GA el individuo más apto (que represente una solución más cercana a la buscada) es el que más probabilidades tiene de ser seleccionado para transmitir sus características a las sucesivas poblaciones.
- **Recombinación.** Tras seleccionar a uno o más individuos, existe una probabilidad (generalmente alta) de que éstos generen uno o más individuos con una composición genética resultado de la combinación de éstos. El algoritmo general habla de selección de 2 individuos y del operador “cruce”, pero es posible generalizarlo para que el cruce vaya de 1 individuo (i.e. no hay cruce, es traspaso de una versión de los genes) a n (i.e. todos los individuos tienen probabilidad de transmitir su genotipo, ya sea directamente o aplicando una función a combinaciones de éstos). La recombinación consigue que la composición genética de los individuos (que probablemente sea buena porque han sido seleccionados frente al resto de individuos) se perpetúe en sucesivas poblaciones.
- **Mutación.** Una vez generada una descendencia en la recombinación (o no, pueden ser los individuos originales), los individuos

tiene cierta probabilidad (por lo general baja) de que su genotipo sea modificado aleatoriamente. Esto puede hacer del individuo resultante una mala solución, pero también puede hacer que la solución sea nueva o si no, al menos ofrecer más variedad genética a la población.

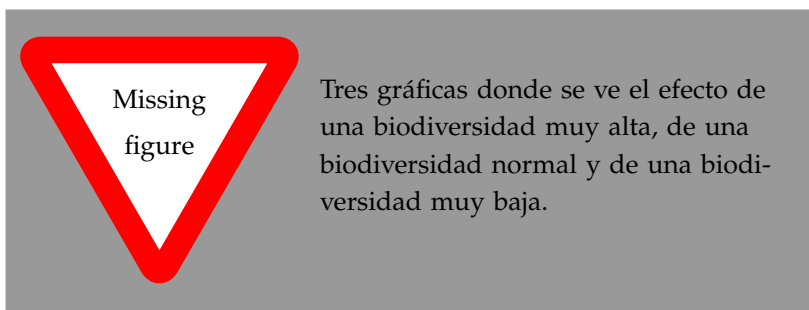
- **Reemplazo.** Tras uno o más pasos de selección, recombinación y mutación, los nuevos individuos generados reemplazarán a individuos de la población inicial. En general la población en un GA contiene el mismo número de individuos según pasan las generaciones, pero no es un requisito fundamental.

El balance de la biodiversidad: presión selectiva y convergencia

Las características de todo individuo vienen determinadas por su genotipo. Éste contiene conjuntos de uno o más genes que se manifiestan en el fenotipo como rasgos característicos del individuo. Algunos rasgos son más favorables que otros¹⁶ y hacen que el individuo tenga más probabilidad de sobrevivir y, por tanto, de recombinarse y transmitir sus genes.

LA PRESIÓN SELECTIVA es el mecanismo por el cual se favorece o penaliza la biodiversidad. Cuando es fácil sobrevivir en un entorno, todas las variaciones de una especie son capaces de recombinarse, tener descendencia y seguir perpetuando rasgos característicos. Cuando es difícil, sólo los más aptos son los que sobreviven y por lo tanto los que se reproducen.

LA BIODIVERSIDAD se refiere a la riqueza genética que existe entre los individuos de la población. Si la biodiversidad es alta, tenemos más posibilidades de combinaciones de genes. Visto de otro modo, cuando tenemos más variedad de genes en la población, la cantidad de posibles soluciones a explorar es mayor.



Mantener una biodiversidad justa en un GA es esencial. Cuando la variedad genética es demasiado baja, los individuos tienden a ser muy parecidos y se hace imposible que aparezcan rasgos genéticos nuevos¹⁷, haciendo que el algoritmo se estanque. Por otro lado,

¹⁶ Lo bueno o malo que es un individuo viene determinado también por el entorno y no sólo por el individuo. Un ejemplo claro es el de la mariposa del abedul. A principios del siglo XIX, en Inglaterra, la mariposa se confundía con los líquenes de la corteza del abedul debido a que tenían colores y patrones similares (ambas eran blancas). Sin embargo, un cambio en un único gen las convertía en negras, y por tanto muy apetecibles para los depredadores, por lo que se puede concluir que las mejores eran las de color blanco. Sin embargo, para mediados de siglo, el hollín del humo del carbón de las fábricas había teñido de negro la práctica totalidad de los líquenes convirtiendo la ventaja de la mariposa blanca en una total desventaja. Por tanto, los rasgos son favorables dependiendo del entorno.

Figura 9: No sé cómo explicarlo. uando esté más despierto mejor.

¹⁷ En realidad sí es posible, con la mutación, pero generalmente es un operador que entra en juego con probabilidad muy baja y por tanto no se puede depender únicamente en él para mantener la biodiversidad.

cuando la variedad genética es muy alta quiere decir que hay muchos individuos todos con características muy distintas. Esto provoca tal cantidad de posibles variaciones que el algoritmo acaba avanzando muy lentamente hacia la solución. Una ilustración de este concepto se puede ver en la figura 9.

SEGÚN EL ALGORITMO CONVERGE, LA BIODIVERSIDAD DISMINUYE. Esto es debido a que cuando según vamos acercando nuestra población a una región del espacio, todos los valores que se toman tienden a ser cercanos a esta posición. Esta característica se denomina *explotación*, y es el caso contrario a la *exploración*, es decir, cuando los individuos están distribuidos por todo el espacio de búsqueda buscando posibles soluciones.

Es importante que nuestro algoritmo tienda a explorar en los estadios iniciales de su funcionamiento, y que tienda a explotar cuando se encuentre en momentos cercanos a la localización de una solución. Ésto se puede lograr mediante un buen ajuste de la presión selectiva, aunque el operador de recombinación es el más indicado para gestionar este equilibrio exploración/explotación.

Aptitud y selección de individuos

Cuando hablamos del individuo “más apto”, nos referimos a la mejor solución encontrada hasta el momento. Para ello, la manera de decidir cuándo una solución es mejor que otra es la aptitud o *fitness* de una solución.

¹⁸ En realidad, para la mayoría de los algoritmos de los diferentes pasos de un GA basta con que la función de fitness defina una función de orden entre soluciones (de la forma “la solución S_1 es mejor que la solución S_2 ”). Sin embargo, asociar un valor real a cada una de las soluciones nos da esta relación de orden además de otras ventajas, como por ejemplo un valor del que podemos observar la evolución a lo largo de los pasos y otros algoritmos como la selección proporcional al fitness.

EL FITNESS de una solución es una función que nos da una idea de la calidad de ésta como número real¹⁸ en su entorno.

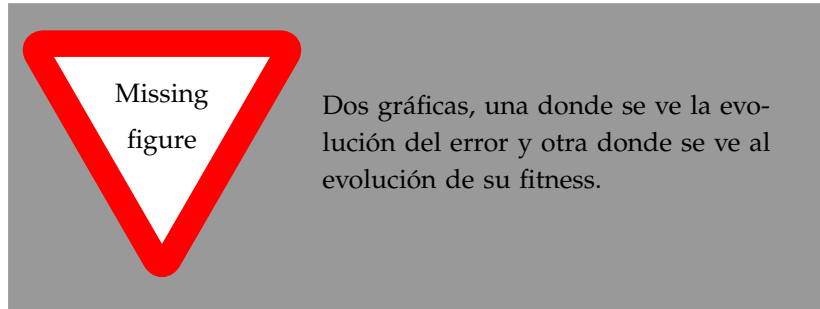
El rango del depende de la representación de las soluciones y del problema en sí. Una forma muy útil que mantiene la relación de orden y permite su fácil representación gráfica es la mostrada en la ecuación 2 (ver figura 10). No obstante esta representación es válida únicamente cuando es posible conocer el error de una determinada solución. Otros casos requieran de funciones alternativas de fitness.

$$\frac{1}{1 - error} \quad (2)$$

TODO!No sé si se podría hablar del gaussian kernel aquí como medida del error. De hecho no sé si se usa en algún lado, pero yo creo que sería válido y que da representaciones más fieles en una gráfica.

LA SELECCIÓN se apoya en el fitness para aumentar o disminuir la probabilidad de un individuo para ser seleccionado. Este operador

Figura 10: Representación del fitness como función inversamente proporcional al error de una solución. Esta forma mantiene el fitness siempre en el intervalo $(0, 1]$. Cuanto menor es el error, esto es, cuanto más tiende a 0, más se acerca el fitness a 1.



es uno de los más influyentes en un GA para mantener el nivel de presión selectiva.

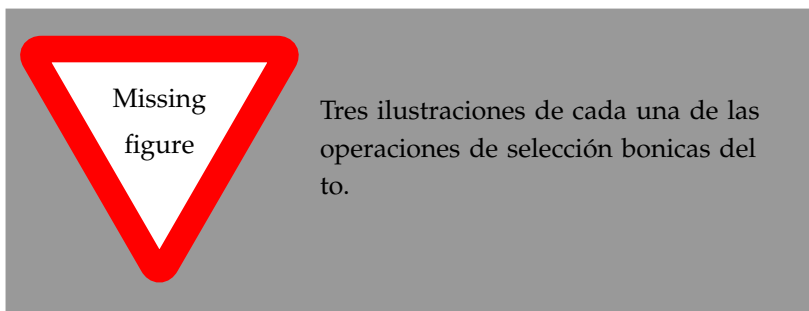


Figura 11: Tres ilustraciones de las técnicas de selección más comunes en Genetic Algorithm. De izquierda a derecha (a) selección proporcional al fitness, (b) selección proporcional a la posición y (c) selección por torneo.

El operador de selección se ejecuta cada vez que se quiere seleccionar un individuo para la recombinación por lo que, si la recombinación requiere de dos individuos, el operador de selección será llamado dos veces. Las técnicas más comunes, ilustradas en la figura 11 son las siguientes:

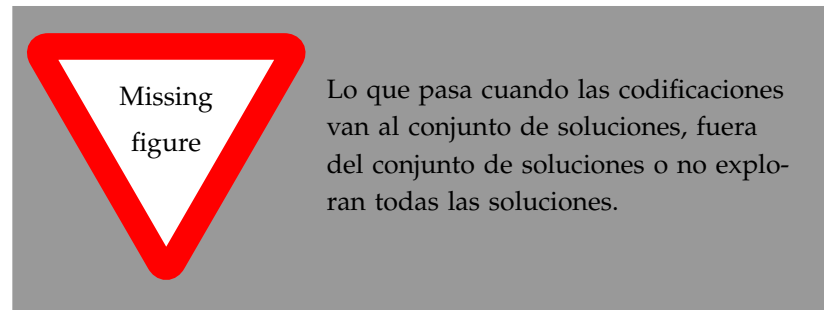
- **Selección proporcional** al fitness (también conocido como *roulette wheel*). En esta selección a los individuos se ordenan por fitness y se les asigna una probabilidad proporcional a su fitness. Una vez hecho esto, se genera un valor aleatorio y se comprueba qué individuo ha sido seleccionado. Este operador tiene la desventaja de que, si en un momento unos pocos individuos consiguen un fitness muy alto respecto al resto, acapararán la práctica totalidad del rango de probabilidades, haciendo que disminuya drásticamente la variedad genética.
- **Selección proporcional a la posición.** Varía con el anterior en que la probabilidad asociada a cada individuo es proporcional a la posición y no al fitness. Aunque supone una ventaja respecto al anterior en materia de mantener la biodiversidad, si el tamaño de la población es medio o grande, la probabilidad queda muy repartida, eliminando casi por completo la presión selectiva sobre la población.
- **Selección por torneo.** Esta selección se apoya en dos pasos. En el primero, se extraen aleatoriamente un número n de individuos, in-

dependientemente de su . Una vez extraídos, se seleccionará aquel que tenga el mayor. Este operador es rápido, dado que no es necesario calcular el fitness de todos los individuos, sino de sólo los seleccionados y, con el parámetro n correctamente seleccionado, mantiene la presión selectiva.

Representación de individuos

Un individuo es una solución a nuestro problema. Sin embargo, no existe una única representación para un problema y elegir la mejor representación posible es crucial para un buen desempeño del algoritmo. Para comprender la importancia de una correcta codificación, vamos a formalizar un poco el concepto del problema.

Figura 12: Situaciones posibles cuando la cardinalidad del conjunto de las codificaciones de nuestro problema es (a) mayor, (b) menor o (c) igual que la cardinalidad del conjunto del espacio de soluciones posibles.



Supongamos que tenemos $P = p_1, \dots, p_n$ el dominio del problema (determinado por la codificación del problema, es decir, todos los posibles genotipos), $I = i_1, \dots, i_m$ con $m > n$ el dominio de los posibles individuos, $I_v = i_1, \dots, i_j \subset I$ con $m > j$ el subconjunto de I con todos los individuos que son solución y $f : P \rightarrow I$ una aplicación inyectiva (para simplificar nuestra formalización). Dependiendo de la cardinalidad de los conjuntos se nos pueden dar tres casos (figura 12):

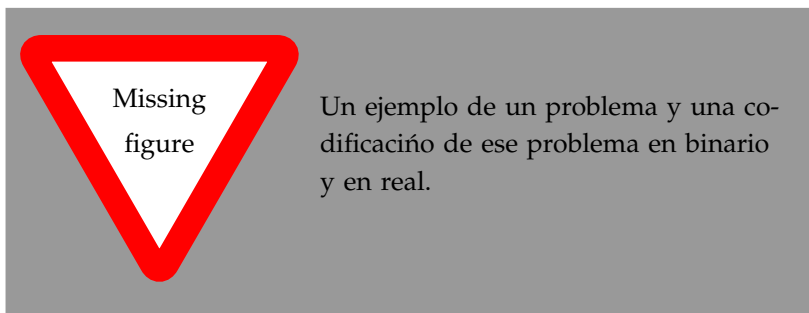
1. $n > j$. Cuando la codificación de nuestro problema genera individuos fuera del conjunto de soluciones, quiere decir que habrá codificaciones que nos generen individuos no válidos. Nuestro algoritmo tendrá que tener mecanismos para que estos individuos no sean seleccionados (e.g. asignando un 0) con la consiguiente penalización en tiempo.
2. $n < j$. En este caso, todas las codificaciones de nuestras soluciones tendrán una correspondencia con un individuo válido. Sin embargo, existirán individuos válidos que no serán alcanzados nunca por nuestras codificaciones, y por tanto habrá soluciones no exploradas.
3. $n = j$. Este caso es el óptimo. Al aplicar f a cualquier codificación de solución para nuestro problema, alcanzaremos un individuo

válido. Por tanto no tendremos ningún problema de individuos inválidos ni de soluciones fuera de nuestro alcance.

Por ello es importante dedicar especial atención a la codificación de los genotipos de nuestro individuo. Y aunque la codificación es dependiente del problema, existen codificaciones estándar que se usan en la mayoría de los problemas.

LA CODIFICACIÓN **en lista** es quizá la más usada en algoritmos genéticos. En ésta, el genotipo de un individuo se representa como una lista de longitud fija de valores, todos ellos pertenecientes al mismo alfabeto.

Existen dos representantes típicas de esta codificación, dependiendo de si el alfabeto es finito o infinito¹⁹: La representación binaria, donde los posibles valores que puede tomar cada gen del genotipo es 0 o 1 y la representación real, donde los valores pueden tomar cualquier valor $x \in [0, 1] \subset \mathbb{R}$ (figura 13).



¹⁹ Esto no quiere decir que sean las únicas, sino que son casi equivalentes. Si, por ejemplo, nuestro problema necesita una codificación donde el alfabeto necesita 16 caracteres, es más eficiente trabajar con un alfabeto hexadecimal. Los alfabetos finitos además tienen una desventaja en función de la cardinalidad del alfabeto: si la cardinalidad es c y la longitud del genotipo es l , pueden representar hasta c^l puntos en el espacio por lo que si nuestro problema no requiere de tantos puntos estaremos generando individuos no válidos, con los problemas que ello conlleva (ver Figurafig:why-we-need-a-correct-codification)

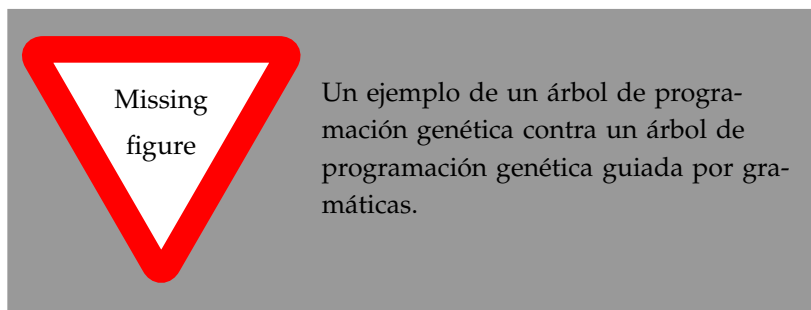
No existe una codificación superior a otra, todo depende del problema. El sentido común suele dictar que cuando el dominio del problema es continuo (e.g. aproximación de funciones continuas), se debe usar una codificación infinita mientras que cuando es discreto (e.g. búsqueda de ruta óptima en un grafo), una finita.

LA CODIFICACIÓN EN ÁRBOL es otra representación clásica en algoritmos genéticos usada, principalmente, para programación genética. En ella, cada uno de los individuos representa un posible programa. Las ventajas de esta representación son claras: facilidad de definir un problema y longitud variable de genotipos.

Sin embargo, posee dos desventajas principales: la primera es relativa a la operación de recombinación. Debido a que los genes del genotipo son de tipos diferentes (i.e. unos representan a variables, otros representan a constantes, otros a funciones unarias, otros a binarias, ...), es necesario implementar controles para garantizar que un individuo (a) recién creado, (b) resultado de una recombinación o (c) mutado sea válido. La segunda está relacionada con la ventaja

de la longitud. Los individuos a lo largo del tiempo en esta representación tienden a crecer (después de todo, el espacio de exploración en un árbol de longitud variable es infinito) por lo que es necesario instalar mecanismos para evitar este comportamiento.

Figura 14: Diferencias entre un árbol que codifica un programa en GP y uno que codifica una palabra de una gramática libre de contexto en Grammar Guide Genetic Programming.



Una línea muy prometedora con codificaciones en árbol es la de la [Programación Genética Guiada por Gramáticas \(GGGP, Grammar Guide Genetic Programming\)](#). En ésta los árboles de los individuos no representan programas en sí, sino una palabra perteneciente a una gramática libre de contexto (figura 14). De esta manera, se garantiza que la generación de individuos es válida (se generan a partir de unas reglas de producción) y la recombinación y mutación son más sencillas de implementar.

Manipulación de individuos: recombinación y mutación

La población debe evolucionar para llegar a una solución. El principal paso en la evolución de una población en un [GA](#) es el de la recombinación, y aunque la mutación nos da probabilidad de saltar a regiones del espacio de búsqueda no exploradas con anterioridad, la recombinación guía todo el proceso de búsqueda hacia la solución.

Ambos operadores dependen más de la representación del problema que del problema en sí. Por ello existen operadores genéricos para codificaciones en lista de alfabeto finito o infinito así como para codificaciones de árboles de programa o gramaticales. Esto no quiere decir que estos operadores no se puedan personalizar, sino que son formas de operar con ventajas y desventajas ya conocidas. Por tanto, si es posible codificar un problema de una forma clásica, no es necesario reimplementar los operadores de recombinación y mutación.

LA RECOMBINACIÓN es la principal responsable de la búsqueda dirigida en el espacio de estados, es decir, de explorar el espacio y explotar las potenciales soluciones. y esta toma dos formas: exploración y explotación. en el espacio guía al algoritmo hacia mejores soluciones

Recombinaciones clásicas para los individuos de antes. Matizar que la recombinación puede depender mucho del problema.

Recombinaciones con árboles y sus problemas principales.

Reemplazo de individuos

Tasa generacional, diferencia entre generacional vs. steady-state y variación de elitismo es en realidad una tasa generacional donde deja un individuo.

Algoritmos de reemplazo clásicos.

Variaciones y operadores adicionales

Hablar de los catastróficos y los que me falten. Hay que encontrar más, porque me parece ridículo un apartado para sólo uno.

Genetic Algorithms distribuidos

Optimización de Fuzzy Inference systems mediante Genetic Algorithms

Agentes inteligentes

Si echamos un poco la vista atrás, en la figura 5 se mostraban los cuatro objetivos perseguidos por la AI. En uno de ellos en particular se la entiende como el estudio del conseguir que las entidades (e.g. sistemas, software, ...) actúen de la manera más inteligente posible. A dichas entidades se las conoce como *agentes*, concretamente en este contexto como *agentes inteligentes*²⁰. Sin embargo, si es difícil encontrar un consenso en la definición de agente más lo es a la hora de definir cuándo la conducta de éstos es inteligente.

Lo que sí existe es una serie de características comunes que se repiten a lo largo de la literatura (figura 15):

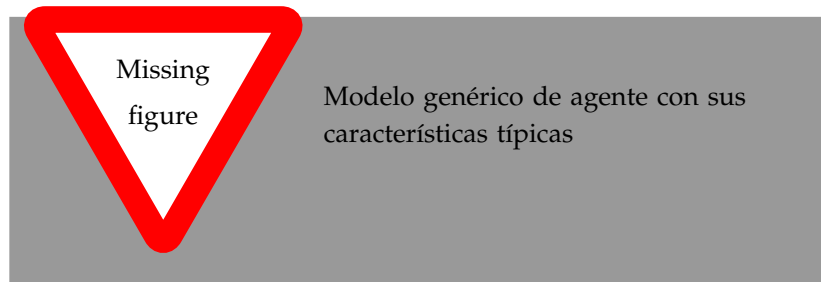
- Operan siempre en un **entorno**, ya sea éste físico (e.g. una red de carreteras para un vehículo autónomo) o virtual (e.g. un cliente de correo electrónico para un clasificador de spam).
- Tienen la capacidad de **percibir** el entorno por medio de *sensores* y de **actuar** sobre él por medio de *actuadores*.
- Son **autónomos** en el sentido de que pueden actuar sin intervención externa (e.g. humana u otros agentes) teniendo control sobre su estado interno y su comportamiento. Algunos autores les presuponen una autonomía absoluta mientras que otros hablan de que sólo es necesaria cierta autonomía parcial.
- Tienen **objetivos** a cumplir, actuando para ello sobre el entorno de la manera que les indique su comportamiento.

²⁰ En realidad los autores prefieren denominarlo *agente racional*, dado que captura la esencia de lo que es un comportamiento inteligente. Sin embargo, según esta definición, hasta un elemento tan rudimentario como un termostato puede ser considerado como elemento inteligente, ya que realiza siempre la mejor acción para cumplir sus objetivos, por simples que puedan parecer. Dónde está el límite entre qué es y que no es un agente inteligente cae dentro de los dominios de la filosofía.

- Pueden ser **sociales**, es decir, tienen la capacidad de comunicarse con otras entidades (e.g. otros agentes) para llevar a cabo sus objetivos.

²¹ En [Russell et al., 2003] se define como “...just something that acts” alegando que la palabra *agent* proviene del latín *agere*. Para clarificar esto, *agere* es la forma verbal para *hacer*, pero imprime un significado de movimiento/acción. **Figura 1.1: Esquema de un agente y sus propiedades.** Aunque existe la posibilidad de definir un agente (de la categoría de agente) como algo que realiza acciones sobre un entorno dependiendo de las percepciones que le llegan de éste y tiene la capacidad de comunicarse con el resto de elementos, incluidos otros agentes.

Por tanto nosotros usaremos la siguiente definición: Un agente es una entidad física o virtual que realiza una acción²¹ de manera total o parcialmente autónoma dada una secuencia de percepciones del entorno en el que se ubica.



Pero, ¿qué hace a un agente inteligente? Según algunos autores, el hecho de que posea unos objetivos y autonomía suficiente para cumplirlos ya denota inteligencia (**TODO!** encontrar el trabajo y citar). Según otros, es necesario que el comportamiento sea flexible, esto es, que sea reactivo (reacciona ante el entorno que percibe), proactivo (iniciativa para tratar de cumplir sus objetivos) y social (capaz de interactuar con otros agentes para cumplir sus objetivos) [Wooldridge et al., 1995]. Y otros directamente exigen, además, un comportamiento racional a la hora de cumplir los objetivos para calificarlo de inteligente (**TODO!** encontrar el trabajo y citar).

Por tanto, asumiremos la definición ofrecida por [Russell et al., 2003] donde, se indica que un agente es considerado **agente inteligente** cuando éste realiza la mejor acción posible (según un criterio de medida). En este contexto, “la mejor acción posible” se refiere en términos de objetivos y comprensión del entorno, que puede ser o no correcta²².

Las nociones de agentes inteligentes y la de **CI** van de la mano. Esto es debido a que su definición funciona a la perfección para las técnicas de la **CI**, esto es, agentes autónomos que perciben el entorno (problema) y actúan de la mejor manera posible sobre él (resuelven) de acuerdo a su conocimiento del medio y su estado interno (en base a algoritmos como **ANN**, **FL**, ...). Por ello desde mediados de los años 1990 el concepto de agente inteligente ha ganado tanta popularidad²³.

Tipos de entorno

La tupla (*entorno, agente*) es esencialmente una metáfora para referirse a la tupla (*problema, solución*) por lo que existen casi tantos

²² Que la comprensión del entorno no sea total es un factor clave que diferencia la racionalidad de la omnisciencia. La omnisciencia significa conocer el resultado de toda acción antes de realizarla y por tanto implica el conocimiento de absolutamente todos los detalles del entorno. La racionalidad existe dentro de un contexto de conocimiento limitado.

²³ Tanto es así que en algunos trabajos se define el objetivo de la **AI** como la implementación de la función agente, esto es, la función que realiza la correspondencia de una percepción a una acción, para un problema dado.

entornos diferentes como problemas.

Afortunadamente es posible caracterizar los entornos de acuerdo a un conjunto de propiedades o dimensiones. Este conjunto es usado por la totalidad de la literatura a la hora de caracterizar entornos:

- **Observable.** Un entorno es **totalmente observable** cuando el agente es capaz de captar toda la información relevante para la toma de una decisión y no necesita mantener ningún modelo interno del entorno, **parcialmente observable** cuando la información obtenida es incompleta o tiene ruido y **no observable** cuando el agente no posee sensores.
- **Multiagente o. monoagente.** Un entorno es **multiagente** cuando requiere de múltiples agentes interactuando para llegar a una solución mientras que es **monoagente** cuando sólo requiere de uno para ello.
- **Determinista o. no determinista.** Si el estado del entorno actual depende totalmente del estado anterior, se dice que el entorno es **determinista**. Si no es así, se considera **no determinista o estocástico**²⁴.
- **Episódico o. secuencial.** Un entorno en el que las acciones se dividen atómicamente donde cada una de ellas conlleva un ciclo de (percepción, decisión, acción) y sin relación una con otra se denomina episódico. Si en lugar de ello la acción del agente puede afectar a las decisiones futuras se dice que el entorno es **no episódico o secuencial**.
- **Estático o. dinámico.** Si durante la toma de decisión en entorno no cambia, se dice que el entorno es **estático**. En caso contrario, se dice que es **dinámico**.
- **Discreto o. continuo.** Esta dimensión en realidad se divide en cuatro, estado del entorno, tiempo en el entorno, percepciones y acciones. La dimensión es **discreta** cuando ésta se divide en una partición discretizada, y **continua** cuando no. Por ejemplo, en el Juego de la Vida de Conway, si se modela en un sistema multiagente, tanto el estado (i.e. tablero) como el tiempo (i.e. turnos) como las percepciones y acciones están discretizadas. Sin embargo, en un entorno de conducción automática se puede determinar que las cuatro dimensiones son continuas.
- **Conocido o. desconocido.** Un entorno es **conocido** cuando es posible determinar cuál va a ser el resultado de una acción. Si por el contrario no es posible, entonces se dice que es **desconocido**.

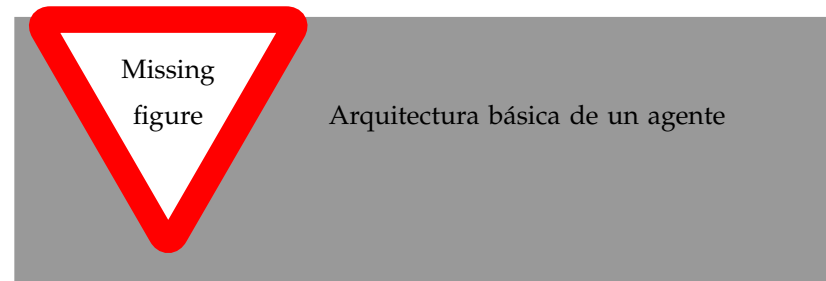
²⁴ En general, los entornos del mundo real tienden a ser tan complejos que es imposible para un agente abarcar todos los aspectos medibles de éste. Por lo tanto, sea o no la naturaleza del entorno determinista, en general se suele suponer éste como no determinista.

Arquitecturas

Existe una serie de arquitecturas básicas o tipos de agentes que dependen principalmente de cómo perciben el entorno y de qué forma

se comportan aunque, dependiendo de los autores, las nomenclaturas, tipologías y esquemas pueden variar. Por ello, hemos decidido ofrecer una abstracción donde poner de manifiesto las partes comunes y no comunes entre arquitecturas.

Figura 16: Arquitectura básica de un agente. Aunque existen múltiples arquitecturas diferentes, todas se basan en la misma estructura. El agente percibe el entorno, lo interpreta y toma la decisión de cómo actuar sobre él.

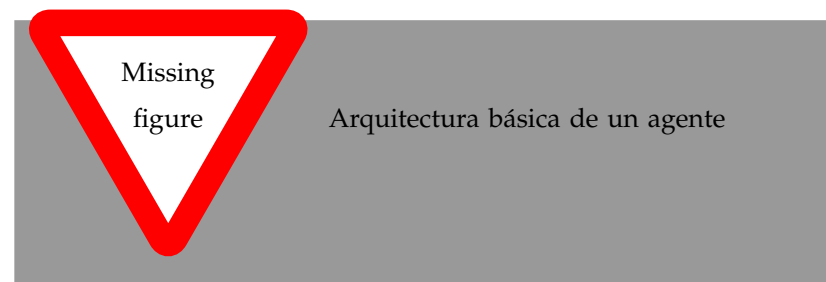


La figura 16 muestra el esquema de las partes principales de un agente. En general, toda arquitectura de agente inteligente está cortada por el mismo patrón y obedece al siguiente funcionamiento:

1. El agente, a través de sus **sensores**, percibe el entorno en el que éste se mueve.
2. De acuerdo a cómo recordamos el entorno (llamémoslo **modelo del entorno**), el agente genera una **interpretación del entorno** tal y como supone el agente que es. Esto es, percibe el entorno y, de acuerdo a sus sensaciones, lo entiende de una determinada forma.
3. Esta interpretación del entorno es pasada a un proceso de **inferencia** el cual, en función la implementación para la consecución de sus objetivos, generará una serie de acciones a realizar sobre el entorno.
4. Estas acciones serán ejecutadas sobre el entorno a través de una serie de **actuadores**, provocando probablemente una modificación en éste que será percibida de nuevo en momentos sucesivos.

La primera diferencia clave surge en la manera que se ofrece al bloque de inferencia la interpretación del entorno y genera la primera clasificación (figura 17):

Figura 17: Ilustración de la diferencia entre un agente sin modelo de entorno y uno con modelo de entorno. Cada acción realizada por el agente con modelo de entorno tiene en cuenta el estado del entorno en momentos pasados. El agente sin modelo de entorno actúa tal y como interpreta el entorno en cada momento, como si sufriese de amnesia.



- **Sin modelo de entorno.** Si el agente ofrece su interpretación del entorno directamente, sin hacer uso de información histórica sobre el entorno que se ha movido. Otras formas de denominar a estos agentes es como *agentes reactivos* o *simple-reflex agents* ([Russell et al., 2003]). Sin embargo, los términos *reactivo* o *reflex* para algunos autores se refieren a la forma de inducción de acciones a partir de percepciones, y por ello preferimos la denominación *sin modelo de entorno*.
- **Con modelo de entorno.** El agente genera su interpretación más detallada del entorno a partir de las percepciones que llegan desde los sensores y de el histórico del entorno que mantiene. Otras formas de llamarlo es *agentes con estado* o *Model-based*, pero lo hemos denominado de esta manera para diferenciar que el modelo que se mantiene en este punto pertenece únicamente al entorno.

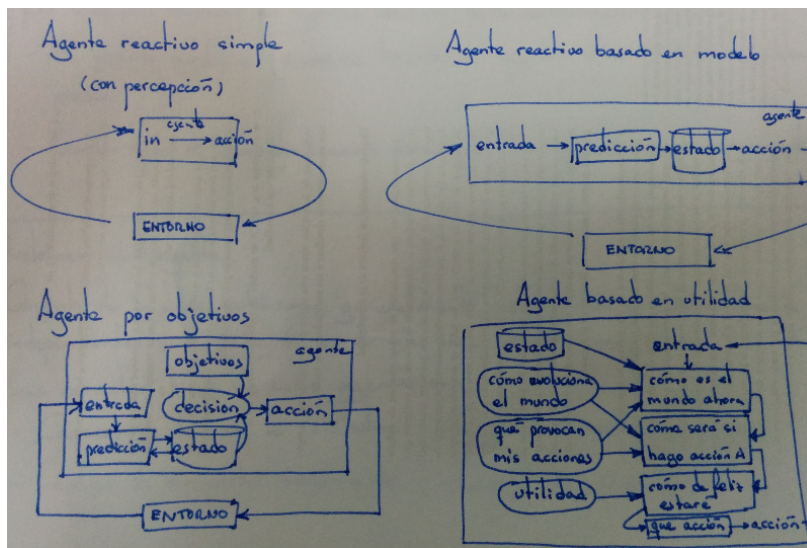


Figura 18: Distintas arquitecturas de agentes en función del comportamiento. Dependiendo de las acciones a realizar, se identifican tres tipos, los reactivos que aplican una acción sin proceso deductivo y los basados en modelo y utilidad (en algunos contextos denominados deliberativos) que basan su comportamiento en alguna forma de deducción.

La siguiente clasificación viene motivada por la forma de deducir el conjunto de acciones a ser aplicadas por parte de los sensores. En este sentido podemos identificar tres tipos distintos de agentes (figura 18):

- **Reactivos.** Son aquellos donde el uso de un proceso de razonamiento explícito es demasiado costoso para producir una conducta en un tiempo aceptable. Se suelen implementar como correspondencias (percepción → acción) sin ningún razonamiento adicional.
- **Basados en objetivos.** Plantean una deducción de forma que determinan cuál sería el estado del entorno tras aplicar varias o todas las acciones que puede realizar. En base a los resultados, selecciona la acción que se corresponde con sus propios objetivos.
- **Basados en utilidad.** Éstos plantean una deducción similar a los basados en objetivos con la diferencia de que, mientras los primeros sólo diferencian entre entorno objetivo o no objetivo, éstos

asignan un valor (i.e. *utilidad*) a cada uno de los escenarios de entorno posibles para seleccionar el mejor (e.g. el que mayor utilidad tiene).

En la literatura se describen muchos tipos de agente, como por ejemplo los agentes BDI (Believe-Desire-Intention) o los agentes lógicos (i.e. el entorno se representa con reglas lógicas y se infiere mediante métodos como por ejemplo deducción lógica o prueba de teoremas). Sin embargo, éstos pueden definirse en los términos aquí expuestos (figuras 16, 17 y 18).

Sistema Multiagente

Son aquellos sistemas compuestos de dos o más agentes que interactúan de alguna manera para llegar a una solución.

Cuando los agentes son inteligentes y el problema cae dentro del dominio de la AI, el ámbito de estudio es el de la **Inteligencia Artificial Distribuida (DAI, Distributed Artificial Intelligence)**, la rama dedicada a la resolución de problemas mediante procesamiento descentralizado.

Desde el punto de vista de la ingeniería de sistemas, y a pesar del aumento de complejidad, los MAS, al ser sistemas inherentemente descentralizados, ofrecen múltiples ventajas frente a los sistemas centralizados tradicionales:

- Los sistemas son más robustos y fiables frente a fallos, ya que los agentes son autónomos e independientes del resto.
- La modificación del sistema se puede realizar sobre la marcha, agente a agente sin necesidad de parar el sistema al completo.
- Su diseño fuerza a desacoplar las dependencias entre agentes.
- Son inherentemente paralelizables y por tanto pueden llegar a ser más eficientes que sus homólogos centralizados. Este punto es quizá el más controvertido, ya que esta ganancia en eficiencia se puede perder rápidamente en función de la cantidad de comunicación existente entre agentes.
- Debido al nivel de complejidad alcanzado en los sistemas existentes en la actualidad, la computación se distribuye a través de múltiples sistemas, normalmente heterogéneos. La tendencia además es a la alza. La definición de los MAS hace natural su implementación en este tipo de arquitecturas.

Desde el punto de vista de la AI podemos añadirles la ventaja de que permiten el estudio de conductas complejas de poblaciones a partir del comportamiento de sus elementos básicos, facilitando el estudio de modelos de teorías sobre éstos.

LA COMUNICACIÓN ENTRE AGENTES, se trata de una característica clave en un MAS, ya que para denominarse de esta manera dos o más agentes deben interactuar (i.e. comunicarse) entre si. Esta interacción puede implementarse de diversas maneras²⁵ y siempre toman una o las dos formas siguientes (figura 19):

- **Cooperación.** Los agentes intercambian información entre sí para llegar a una solución. Esta solución puede ser fragmentada (i.e. cada agente posee parte de la solución y se comunican para ir avanzando de forma común hacia la solución global) o poseerla uno o varios agentes que hacen uso de más agentes para ir avanzando la solución.
- **Competición.** Los agentes compiten dentro de un entorno, generalmente mediante la adquisición de recursos limitados. Un ejemplo de este tipo de sistemas multiagente puede ser aquellos sistemas de vida artificial.

²⁵ Las formas clásicas de comunicación son el de paso de mensajes, los sistemas de pizarra y la estigmergia. Para los dos primeros existen dos propuestas para estándar de lenguaje de comunicación, [Knowledge Query and Manipulation Language \(KQML\)](#) ([Finin et al., 1994]) y [Agent Communications Language \(ACL\)](#) ([Poslad, 2007]). La tercera forma de comunicación suele ser muy dependiente del problema y no se apoya en lenguajes estándares. Se trata de una forma de comunicación basada en la modificación del entorno, como la efectuada por las hormigas en la búsqueda de alimento, donde éstas dejan rastros de feromonas modificando el entorno para modificar el comportamiento del resto de la colonia.

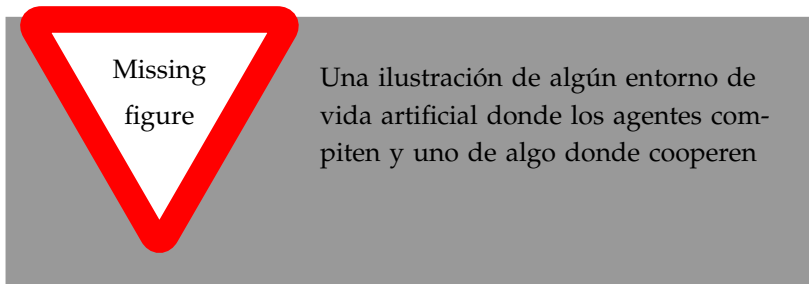


Figura 19: Ilustración de la diferencia entre un agente sin modelo de entorno y uno con modelo de entorno. Cada acción realizada por el agente con modelo de entorno tiene en cuenta el estado del entorno en momentos pasados. El agente sin modelo de entorno actúa tal y como interpreta el entorno en cada momento, como si sufriese de amnesia.

TODO! Aquí una figura de un entorno de vida artificial y un entorno multiagente donde cooperen

Simulación de tráfico

En un sistema de comportamiento tan caótico como lo es el del tráfico, obtener modelos de su funcionamiento es una tarea prácticamente imposible. Por un lado, el número de variables existentes es innumerable y en muchos casos con relaciones no detectables a primera vista. Por otro, es un sistema que funciona en el mundo real, es decir, donde las mediciones en muchos casos afectan a los resultados y donde muchas mediciones no se pueden realizar, ya sea por regulaciones vigentes o por imposibilidad física.

Los simuladores de tráfico son herramientas de software que, usando diferentes modelos para representar sus componentes, describen el tráfico como sistema, permitiendo, entre otros:

- Extracción de resultados y conclusiones en escenarios de tráfico determinados.
- Implementación de técnicas determinadas en tráfico simulado para su evaluación sin necesidad de alterar el tráfico real.
- Introducción de modificaciones en puntos determinados (e.g. espaciales o temporales) de un escenario conocido para estudiar la divergencia en la evolución del tráfico.

El objetivo principal de un simulador de tráfico es el de hacer que sus modelos se parezcan lo máximo posible a la realidad. En este capítulo vamos a ver cuál es la realidad actual de este tipo de simuladores, cuáles son sus diferentes tipologías y formas de modelar los diferentes aspectos del tráfico y, posteriormente, qué simulador de los disponibles en el mercado es el idóneo para nuestro trabajo.

Limitaremos nuestro estudio a los simuladores de DVUs²⁶, obviando otros tipos de simulación de tráfico que nada tienen que ver con esta temática, como por ejemplo los orientados a la evaluación de sistemas de señalización inteligentes (e.g. [Jin et al., 2016]) o a la estimación de emisiones (e.g. [Quaassdorff et al., 2016]).

²⁶ A lo largo del capítulo, se usaran indistintamente los términos DVU, conductor y vehículo para referirse al mismo concepto. En caso de no ser así, se indicará de manera explícita.

Clasificación de simuladores de tráfico

Los aspectos simulables y medibles del problema del tráfico son muy diversos, dependiendo sobre todo:

- Del nivel de **complejidad** del tráfico (e.g. modelar una vía por la que circula un centenar de coches no es lo mismo que modelar una ciudad por la que circulan millones).
- De **qué** queremos medir (e.g. evaluar a un conductor en una situación determinada o evaluar la evolución del flujo de tráfico en un cuello de botella causado por un accidente).
- De **cómo** (e.g. un [Autómata Celular \(CA, Cellular Automaton\)](#) se modela de forma diferente a un modelo lineal de vías o carriles).

El resto de la sección ofrece una visión de las principales categorías existentes para clasificar a los simuladores de tráfico.

Tipos de simulador en función de la complejidad

La complejidad en una simulación se refiere al nivel de detalle que queremos alcanzar durante la ejecución de la misma y/o en sus resultados. Es evidente que según aumentamos el detalle en la simulación aumenta la cantidad de cálculo. Por ejemplo, si queremos modelar el comportamiento de 10 billones de canicas cayendo por un tubo es considerablemente más eficiente modelarlas como un fluido con una serie de parámetros que como una colección de elementos individuales, cada uno con sus propiedades (e.g. masa, aceleración, ...) e interaccionando entre sí.

El caso de los simuladores de tráfico es similar. En éstos existe un amplio intervalo de granularidades, desde por ejemplo el flujo de entrada en una autovía hasta el consumo de carburante de un vehículo en ciudad. Lo más común es clasificar los simuladores dentro de dos grandes grupos, los cuales se ilustran en la figura 20:

- **Microsimulación** o simulación de tipo **micro**. Su objetivo es estudiar, desde un punto de vista de granularidad fina (e.g. vehículos o peatones), las micropropiedades del flujo de tráfico como, por ejemplo, los cambios de carril, las aproximaciones a vehículos de lanteros o los adelantamientos, para evaluar su comportamiento. Tiene dos principales ventajas, la posibilidad de estudiar el tráfico como un todo a partir de sus elementos más simples (ofreciendo una representación más fiel de éste) y la posibilidad de estudiar cada elemento por separado. Sin embargo, la principal desventaja de este tipo de modelos es que cada elemento de la simulación requiere de cómputo independiente y por tanto simulaciones con alto contenido de elementos pueden llegar a ser inviables²⁷.

²⁷ Existen técnicas de computación distribuida que superan ampliamente los límites impuestos por la computación en un único nodo. Un ejemplo relativamente reciente es el simulador de IBM *Megaffic*. Éste implementa un modelo de granularidad micro donde cada elemento es un agente independiente (i.e. sistema multiagente) usando para ello entornos con cientos de núcleos de proceso que proveen de capacidad suficiente para modelar ciudades enteras como Tokio (ver [Osogami et al., 2012] y [Suzumura and Kanezashi, 2012]).

- **Macrosimulación** o simulación de tipo **macro**. Este tipo de modelos centran su esfuerzo en estudiar el flujo de tráfico como un todo (generalmente como fluido), explorando sus macropropiedades (e.g. evolución del tráfico, efectos onda, velocidad media o flujo en vías). Su ventaja principal es que a nivel macroscópico permiten estudiar propiedades que a nivel microscópico requeriría una cantidad ingente de recursos. Sin embargo, con este modelo es imposible obtener información precisa de un elemento en particular del tráfico.



Figura 20: Taxonomía clásica de simuladores en función de la granularidad (complejidad) de la simulación. En la imagen de la izquierda se muestra un ejemplo clásico de macrosimulador donde el tráfico se modela como un flujo a través de las vías. En la de la derecha, se ilustra un modelo clásico de microsimulación donde cada elemento (en este caso vehículos) circula por un carril de la vía.

Aunque ésta es la categorización típica de modelos, en la literatura aparecen otros tipos de modelo con granularidades que pueden considerarse no pertenecientes a ninguno de estos dos conjuntos. Éste es el caso de los simuladores de tipo **sub-micro** y de tipo **meso**, de los cuales se muestra un ejemplo en la figura 21.

Los **sub-micromodelos** especifican granularidades por debajo del nivel de “vehículo” o “peatón”. Por ejemplo, en ([Minderhoud, 1999]) trabaja a nivel de funcionamiento del control de cruce inteligente de un vehículo en función del entorno del vehículo.

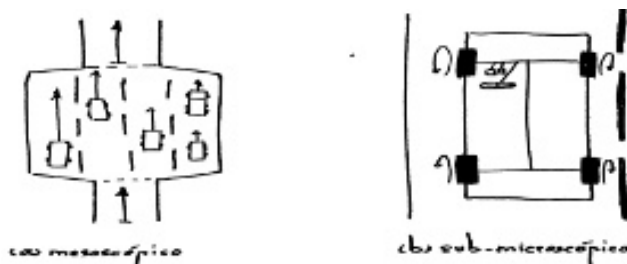


Figura 21: Otras aproximaciones alternativas de modelos en función de la complejidad. Ejemplo de mesosimulación como ventana de microsimulación dentro de un flujo de tráfico en un macrosimulador (e.g. [Munoz et al., 2001]) y ejemplo de submicrosimulación donde se modelan componentes internos de un vehículo.

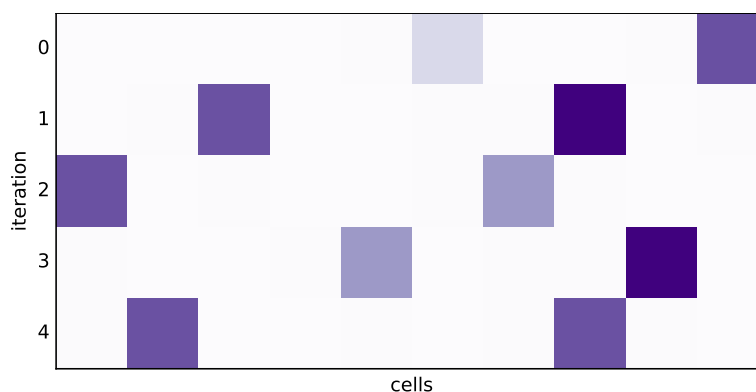
Por otro lado los **mesosimuladores** (e.g. [Munoz et al., 2001] o [Casas et al., 2011]) nacen para amortiguar los problemas inherentes a la complejidad en los micromodelos y a la falta de resolución en los macromodelos.

Dado que el objetivo de la tesis la evaluación de modelos de comportamiento de conductores concretos, nos ceñiremos al uso de simuladores que modelen un nivel de granularidad **micro**.

Tipos de simulador en función del espacio y el tiempo

Existen otras dos formas de clasificar los simuladores en función de cómo evolucionan en la simulación las dimensiones **espacio** y **tiempo**. Sin embargo, aunque *complejidad*, *espacio* y *tiempo* son dimensiones diferentes a la hora de clasificar simuladores, el tipo de simulador según una de ellas tiende a determinar en gran medida los tipos en las demás.

Figura 22: Simulador de tráfico basado en CAs. En éste, el espacio se divide en celdas que pueden, o bien estar vacías, o bien ocupadas por un vehículo a una velocidad (más oscuro implica más lento). Concretamente muestra la evolución a lo largo del tiempo del movimiento de un modelo de *car-following* de 5 vehículos donde en eje x representa la posición en la vía y el eje y el momento temporal (iteración) de la vía. Ilustración generada con el simulador *nagel-scherckenberg-demo* desarrollado en esta tesis.



En el caso de la dimensión **espacio**, la clasificación diferencia las simulaciones que se mueven por un espacio discreto o por uno continuo:

- **Espacio discreto.** Simulación donde el espacio está dividido en celdas que normalmente sólo pueden estar ocupadas por un elemento en un momento determinado. Este es el caso, por ejemplo, de los simuladores basados en CAs. La figura 22 ilustra el comportamiento de uno de estos tipos de simulador.
- **Espacio continuo.** Simulación que transcurre en una secuencia infinita de puntos en el espacio. Es el caso por ejemplo de los simuladores basados en modelos lineales, como podemos ver en la figura 23.

En el caso de la dimensión **tiempo**, la división se realiza en los mismos términos que en los del espacio:

- **Tiempo discreto.** También denominada *simulación de eventos discretos*, divide el tiempo en intervalos discretos, generalmente (aunque existen excepciones) de longitud fija durante toda la simulación. Los simuladores basados en CAs son también simuladores típicos discretos, ya que cada posición en el espacio se va calculando para cada intervalo discreto de tiempo (ver figuras 22 y 24).
- **Tiempo continuo.** En estos simuladores el tiempo es un factor más para un modelo de ecuaciones diferenciales. La figura 23 ilustra un modelo de *car-following* que puede implementarse en una simulación de tiempo continuo si la aceleración viene determinada por un modelo que entre otros factores incluye el tiempo.

Dado que en nuestro caso queremos conocer la situación exacta del vehículo y no una situación aproximada en una separación discreta del espacio. Esto nos dirige hacia simuladores de espacio continuo. Por otro lado, realizamos la recolección de datos en intervalos cuantificables de tiempo, los cuales serán usados para modelar los comportamientos de los conductores y para contrastar los resultados; por tanto, la elección en la dimensión tiempo ha de ser de eventos discretos.

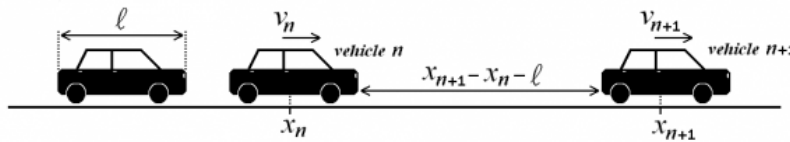


Figura 23: Ejemplo de un modelo lineal en un espacio continuo. La posición del vehículo es un valor $x \in \mathbb{R}$. Este ejemplo muestra un modelo de *car-following* donde el comportamiento de la aceleración del vehículo es determinado por la distancia al coche siguiente. **TODO!** Si hago yo la ilustración, me quito de esta cita y de este dibujo tan feo. Fuente: [Tordeux et al., 2011].

Modelos de microsimulación

Los simuladores que se basan en un modelo de granularidad micro están en su mayoría implementados en dos tipos de paradigma: *Autómatas Celulares* y *Sistemas Multiagente*.

Existe un tercer punto de vista a la hora de implementar este tipo de modelos, que es el de los sistemas de partículas. Sin embargo, su ámbito de aplicación es el mismo que el del punto de vista macroscópico, esto es, usar sistemas de partículas para el análisis del tráfico como fluido. Por tanto, el resto de la sección describirá los dos tipos principales sin tener en cuenta éste último.

Microsimulación basada en *Autómatas Celulares*

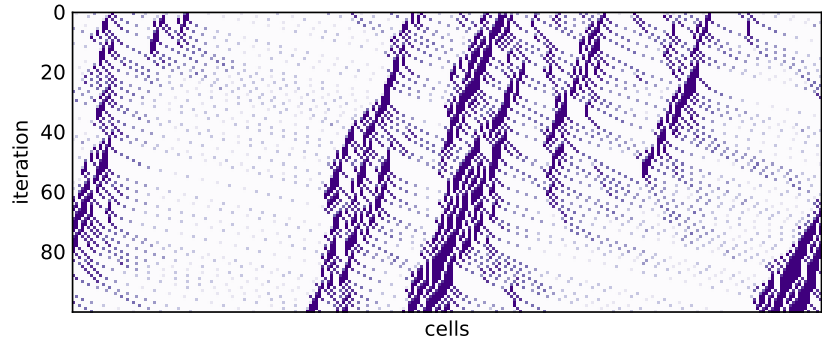
Un *CA* es una colección ordenada de celdas o *células* ordenadas en un espacio n -dimensional que parcelan el universo en estudio. Cada una de ellas se encuentra en un estado (e.g. contiene un valor numérico), y el estado de toda la malla se actualiza de manera síncrona (esto es, todas a la vez) en intervalos regulares de tiempo denominados *ciclos*. El cambio de estado de cada célula depende de los valores de las células vecinas y del mismo algoritmo de modificación al que responden todas y cada una de las células.²⁸

Estos modelos de microsimulación, debido a la propia naturaleza de los *CAs*, se encuentran clasificados como simuladores de tiempo y espacio discreto, y se usa debido a su facilidad de implementación y a su eficiencia, ya que es fácilmente paralelizable.

El modelo clásico de esta aproximación es el propuesto por Nagel-Scherckenberg en su artículo *A cellular automaton model for freeway traffic* [Nagel and Schreckenberg, 1992], un modelo teórico creado para la simulación de tráfico en autopistas. La figura 24 muestra la

²⁸ Existen arquitecturas diseñadas para operar de esta manera, esto es, arquitecturas basadas en *CAs* (e.g. [Margolus, 1993]). En ellas, cada ciclo de reloj actualiza todas las celdas de memoria del autómata. Éstas arquitecturas se suelen usar para la implementación de modelos físicos superando en varios órdenes de magnitud la capacidad computacional de las arquitecturas tradicionales.

Figura 24: Aparición de retenciones en una autopista de 250 celdas de usando el modelo Nagel-Scherckenberg. La densidad de ocupación es de 50 coches en la vía, la velocidad máxima es de $5c/\Delta t$ y la probabilidad de frenada es de $p = 0,5$. Se puede observar en la figura cómo se desplazan las olas del atasco a lo largo de las 100 iteraciones. Ilustración generada con el simulador *nagel-scherckenberg-demo* desarrollado para esta tesis.



- La vía está dividida en celdas de longitud $7,5m$. La razón de este valor es debida a que esta es la distancia media entre los parachoques traseros de dos coches consecutivos en un atasco.
- La celda puede tener dos estados, vacía o con un vehículo a velocidad $v = \{0, \dots, v_{max}\} \in \mathbb{N}$, esto es, discreta. La unidad de medida es $c/\Delta t$ siendo c un número de celdas.
- Δt queda establecido en $1s$, considerado el tiempo medio de reacción de un conductor ante una eventualidad. Esto hace, por ejemplo, que una velocidad de $6c/\Delta t$ sea $45m/s$ ($162km/h$).
- En cada ciclo, se realizan cuatro acciones de manera simultánea: acelerar (una unidad si no están a la máxima velocidad), frenar (si se ven obligados en función de la velocidad y la distancia del siguiente vehículo), freno aleatorio (la velocidad se reduce en una unidad hasta un mínimo de $v = 1c/\Delta t$ con una probabilidad de $p = 0,5$) y reposicionamiento (se avanzan tantas celdas como indica la velocidad.)

En general los modelos de la literatura suelen ser una variación de éste con ligeras modificaciones para estudiar aspectos concretos de modelos de tráfico, como la modificación del paso de *aleatorización* (e.g. [Barlovic et al., 1998]) o celdas más pequeñas (e.g. [Krauss et al., 1997]) para comprobar la metaestabilidad del flujo de tráfico, o modelos y reglas para vías de dos carriles (e.g. [Brilon and Wu, 1999] en la figura 22 y [Nagel et al., 1998]).

Microsimulación basada en sistemas multiagentes

Los modelos basados en CA, aunque interesantes, no son suficientemente realistas desde un punto de vista microscópico. Por poner un ejemplo, en una situación típica de un modelo Nagel-Scherckenberg, las posiciones de los vehículos se dan en múltiplos de $7,5$ metros.

Además, un vehículo realiza aleatoriamente aceleraciones y deceleraciones de 27km/h . Incluso en una situación favorable, cualquier vehículo puede realizar una aceleración de 0 a 162km/h en sólo 6 segundos. Por tanto, no ofrecen una visión demasiado realista ni fiable en caso de querer realizar estudios muy detallados de tráfico a nivel micro.

Por otro lado, en un MAS cada uno de los individuos o “agentes” tiene su propia entidad dentro del sistema. Esto es, perciben tanto el entorno como al resto de agentes y actúan de acuerdo a lo percibido y a su comportamiento preestablecido. Basarse no sólo en las magnitudes físicas del resto de vehículos (e.g. distancia, aceleración, ...) sino también en un comportamiento de conducción ofrece un interesante campo de estudio a nivel cognitivo. El paradigma de los sistemas multiagentes se desarrolla en detalle en el capítulo ?? y los comportamientos concretos de los agentes de interés para esta tesis en el capítulo II. Por ello, este apartado únicamente hará una pequeña introducción a estudios existentes y aplicaciones de simuladores basados en este modelo.

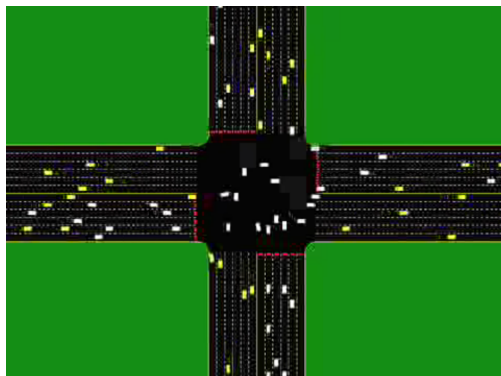


Figura 25: Simulación de comportamiento en intersección basada en un MAS. En ésta, cada uno de los vehículos representa a un vehículo real que posee un controlador para hacerlo autónomo. Modelar este caso de estudio con una arquitectura basada en MASs permite centrarse en el diseño del agente en concreto (i.e. el controlador de conducción del vehículo) y estudiar el comportamiento emergente surgido de la interacción de todos los agentes. Fuente: Proyecto AIM (<http://www.cs.utexas.edu/~aim/>).

A diferencia de los CAs, los MASs pueden emplazarse en un entorno virtual que represente un espacio continuo y no discreto. Esto permite modelar con mayor fidelidad magnitudes físicas asociadas a cada agente (e.g. posición y velocidades actuales, dimensiones del vehículo, masa, velocidad máxima permitida, ...). Sin embargo, aun así no es una propiedad inherente de éstos. No existe ninguna limitación en cuanto a la representación del espacio y es perfectamente posible representar un modelo basado en un CA usando para ello un MAS.

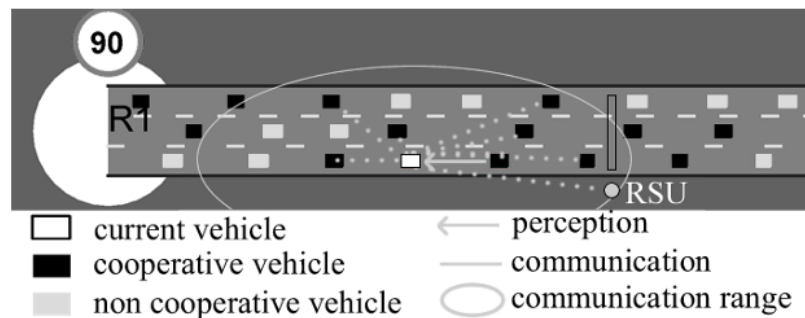
Cada uno de los agentes es independiente del resto, y una consecuencia directa es que el comportamiento de cada individuo permite evaluar comportamientos grupales complejos, como el descrito en la figura 25²⁹. Esta independencia da la posibilidad de tener todos los agentes diferentes entre sí y, por tanto, ofrece la ventaja de permitir experimentar con diferentes perfiles de conducción (e.g. un perfil agresivo en un flujo de tráfico dominado por conductores tranquilos). Esto es debido a que en un MAS cada agente es una parte del sistema y las decisiones de cómo se ha de comportar las toma él mis-

²⁹ Existe también una aproximación de modelar agentes como zonas de la vía y donde los vehículos, aunque poseen comportamiento, lo reciben del agente que les guía de acuerdo a la zona en la que se encuentran. Esto tiene la ventaja de que el paso de información a vehículos dentro de la misma zona se realiza mucho más rápido en un entorno distribuido. Para más información, en [Galis and Rao, 2000] se implementa un simulador siguiendo este concepto.

mo. Desde el punto de vista de un CA, el comportamiento existe en cada celda, sin dar control al contenido o estado de cada celda.

En general los estudios basados en este modelo suelen seguir el patrón 1 DVU, 1 agente, dando así una enorme cantidad de posibilidades a experimentar. Por ejemplo en [Das et al.,] se hace uso de sistemas difusos para decidir cómo comportarse en la vía mientras que en [Ehlert and Rothkrantz, 2001a] se hace uso de un patrón reactivo. Otros, como [Dia, 2002] o [Balmer et al., 2004] hacen uso de encuestas o censos para establecer las propiedades y calibrar los parámetros de diferentes tipos de agentes.

Figura 26: Captura de pantalla del simulador MovSim. Este simulador implementa un modelo multiagente donde los vehículos incorporan sistemas de comunicación vehicular. El estudio se centra en el uso de la comunicación entre vehículos para el acoplamiento dinámico de vehículos en sus respectivos carriles. Fuente: [Gu et al., 2015].



Los estudios en materia de simuladores de tráfico con MAS no se limitan a vehículos, sino que se usan también en otras áreas como el control de luces de tráfico o agentes para peatones entre otros. Un ejemplo es el estudio presentado en [Clymer, 2002] donde los agentes del sistema son las señales de tráfico luminosas y no los vehículos, y donde el objetivo es adaptar la señalización en una red de carreteras para minimizar al máximo el tiempo de espera por parte de los vehículos en las intersecciones gestionadas por las señales.

En los últimos años, otro concepto que está en auge es el de las llamadas redes o comunicaciones intervehiculares, o *Vehicle-to-Vehicle* (V2V) y extravehiculares, o *Vehicle-to-Infrastructure* (V2I). El modelo de MAS permite la implementación rápida de diferentes políticas y protocolos de comunicación via sensores y actuadores para estudiar estos tipos de redes de comunicación (ver figura 26). Estudios como por ejemplo [Shiose et al., 2001] o [Galís and Rao, 2000] hacen uso de un MAS para implementar diferentes formas de V2V con el objetivo de aliviar congestiones de tráfico (en el primer caso) y por el propio estudio de las comunicaciones en si (en el segundo caso). En el caso de redes V2I, un buen ejemplo es [Dresner and Stone, 2004], donde se representan como agentes tanto los vehículos como las intersecciones de la vía. Éstas gestionan un sistema de reservas de tokens que los vehículos solicitan cuando van a entrar en la intersección y devuelven cuando salen, gestionando comunicando en todo momento mediante eventos los cambios en dicho sistema. El estudio concluye que una comunicación de este tipo es más eficiente que una intersección clásica basada en señales de tráfico luminosas.

Software de simulación

Para la realización de esta tesis es necesario contar con un paquete de simulación que permita modelar un MAS en el que poder ejecutar los modelos de comportamiento desarrollados.

Aunque en un principio se ha valorado el desarrollo de una solución propia, la oferta de simuladores en el mercado es muy amplia, cada uno de ellos implementando uno o varios modelos diferentes y bajo muy distintas licencias. Por ello se ha optado por la elección de un paquete de simulación ya desarrollado.

Para elegir el mejor simulador que se adapte a nuestras necesidades se ha realizado un listado de características obligatorias y, de este modo, realizar una primera criba eliminando simuladores no aptos:

1. **Tipo de simulador.** Para nuestras necesidades es necesario un simulador que implemente **microsimulación**, ya que es el único tipo de granularidad que permite evaluar el comportamiento de un conductor independientemente del resto de la simulación. Además, debido a la forma en la que se recolectan los datos, es necesario que represente un **espacio continuo** y una dimensión de **tiempo discreto** con una resolución de al menos 1 segundo.
2. **Modelo de simulación.** Debe ofrecer un entorno basado en un MAS donde cada DVU se comporte como agente individual.
3. **Entorno de simulación.** Debe ofrecer un entorno de **simulación de tráfico general**, permitiendo la creación de escenarios. Quedan excluidos los simuladores de propósito específico o de casos particulares como simuladores de autopistas, congestiones o colisiones.
4. **Extensibilidad.** El simulador debe permitir extender de alguna la ejecución de los modelos desarrollados en los agentes (DVUs). Aunque se puede considerar que si es simulador **Software Abierto (OSS, Open Source Software)**, es posible modificar su comportamiento para adecuarlo a los modelos desarrollados, es mejor que el propio software ofrezca los mecanismos necesarios para la integración sin necesidad de tocar los fuentes del sistema.
5. **Sistema operativo.** Es imprescindible que el software se ejecute sobre sistemas operativos GNU/Linux por la configuración de los sistemas sobre los que se trabaja.

Posteriormente se ha desarrollado un listado de características deseables. No son determinantes para descartar simuladores pero sí favorecen la elección de unos sobre otros.

1. **Activo.** Es preferible que el sistema esté activamente desarrollado porque eso favorece la aparición de parches y mejoras sobre el software. En caso contrario, se trata de un proyecto con poca actividad por parte de sus autores.

2. **Lenguaje de programación.** Es favorable la implementación de los modelos en código Python.
3. **Licencia.** Es preferible una licencia de tipo [OSS](#) ya que, en caso de error o falta de funcionalidad, es posible acceder a los fuentes para modificarlos.
4. **Sistema operativo.** Es favorable que el sistema se ejecute en entornos tipo OS-X.

Entornos de simulación a estudiar

El primer listado de características deja atrás la mayoría de simuladores (una gran cantidad de ellos son o bien de propósito específico, están desarrollados para sistemas operativos Windows o no permiten extender su modelo). Tras la selección, nos quedamos con tres simuladores: [AORTA](#), [MatSIM](#) y [SUMO](#).

Cuadro 2: Tabla comparativa donde se contrastan las características de los simuladores seleccionados. El simulador [SUMO](#) es el que más características cumple de las originalmente planteadas.

	AORTA	MatSIM	SUMO
Activo	✗	✓	✓
Lenguaje de programación	Scala	Java	C++/Python
Licencia			
Propietaria	✗	✗	✗
Open Software	✓	✓	✓
Compatible GPL	✓	✓	✓
Extensibilidad DVUs			
Código fuente	✓	✓	✓
API	✗	✗	✓
Sistema operativo			
GNU/Linux	✓	✓	✓
OS X	✓	✓	✓
Windows	✓	✓	✓

Dichos entornos están prácticamente igualados en las características presentadas, tal y como se puede observar en el cuadro 2. in embargo, en materia de extensibilidad, [SUMO](#) es el único que permite el desarrollo de [DVUs](#) de manera externa. Tanto [AORTA](#) como [MatSIM](#) requieren la modificación del código fuente del simulador para varias los comportamientos de los conductores. [AORTA](#) además no es un proyecto que se mantenga activo en la actualidad (las últimas modificaciones del repositorio datan de principios del año 2014).

No obstante se ha tratado de modificar los comportamientos de los conductores en los tres simuladores para validar este hecho y ha quedado patente que para nuestro cometido es mucho más eficaz usar [SUMO](#) como simulador para nuestro estudio.

Entorno seleccionado: SUMO

En definitiva, el simulador que más se adapta a nuestras necesidades y el que se usará como simulador base en el desarrollo de esta tesis será SUMO. Existen muchas publicaciones acerca de aspectos de este simulador, pero las principales son [Krajzewicz et al., 2002], [Behrisch et al., 2011] y [Krajzewicz et al., 2012].

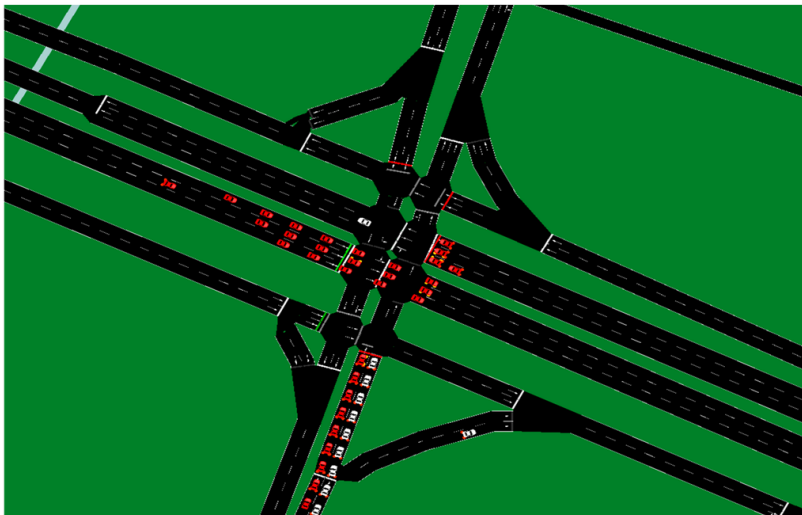


Figura 27: Captura de pantalla del simulador SUMO. Además de software de simulación propiamente dicho, SUMO provee de una interfaz gráfica que permite una visualización general, de zonas y de elementos en concreto a la vez que permite la variación de configuración de la simulación durante el desarrollo de la misma. **TODO!** Meter una imagen de nuestras simulaciones cuando estén, a poder ser sin color. Hacer el alto más pequeño

SUMO es un entorno de microsimulación de código abierto³⁰ desarrollado por el instituto de sistemas de transporte del Centro Aeroespacial Alemán. Implementa un modelo discreto en el tiempo y continuo en el espacio.

Además de simulación clásica, incorpora de una interfaz gráfica (se puede ver una captura de la vista gráfica en la figura 27) donde se puede ver el comportamiento de cada vehículo durante la simulación. Es interesante para obtener de un vistazo información acerca del funcionamiento del modelo en concreto a controlar. Otras de las características que el simulador ofrece son las siguientes:

- Granularidad micro y meso.
- Multimodalidad permitiendo modelar no sólo tráfico de vehículos sino de peatones, bicicletas, trenes e incluso de barcos.
- Simulación con y sin colisiones de vehículos.
- Diferentes tipos de vehículos y de carreteras, cada una con diferentes carriles y éstas con diferentes subdivisiones de subcarriles (diseño conceptual para permitir modelar comportamientos en vehículos como motocicletas y similares).

Al estar licenciado bajo la licencia GPL, su distribución implica a su vez la distribución de su código fuente. Esto permite la modificación de su comportamiento y el desarrollo de nuevos modelos

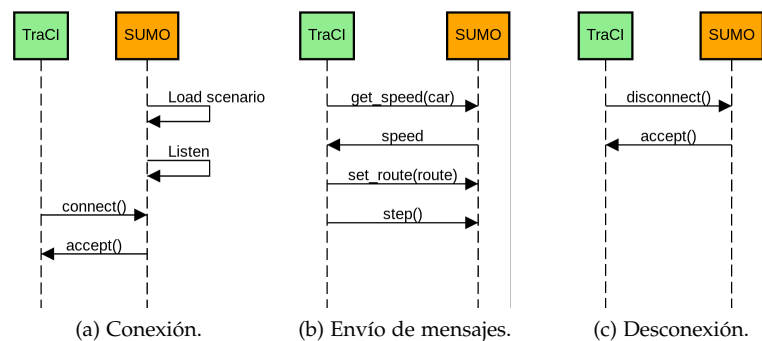
³⁰ Licenciado bajo la GPL, concretamente la versión 3.0.

integrados dentro del simulador. Sin embargo nosotros no haremos uso de esta característica, sino que usaremos SUMO como aplicación servidor y el módulo TraCI como aplicación cliente desde donde gestionar todos los aspectos de la simulación.

La interfaz TraCI

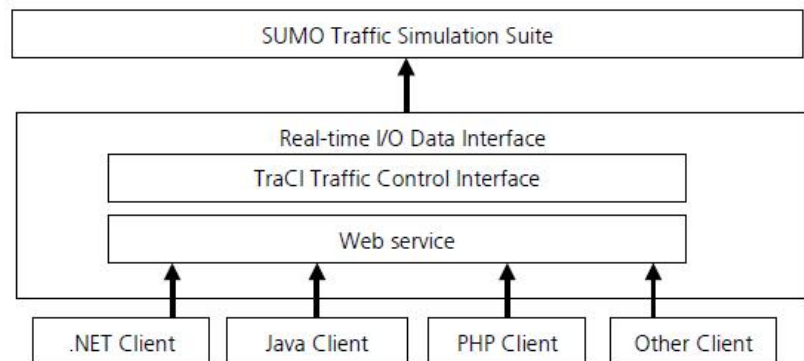
TraCI ([Wegener et al., 2008]) es tanto el nombre del protocolo de comunicación expuesto por SUMO en su versión servidor como el nombre de la librería escrita en Python para interactuar con el mismo.

Figura 28: SUMO ofrece la posibilidad de interactuar con la simulación desde cualquier aplicación a través del uso del protocolo TraCI. En la figura podemos ver, de izquierda a derecha, ejemplos de comunicación a través de la interfaz como el *handshake* o inicialización, mensajes de obtención de información y modificación de la misma más una solicitud de avance de paso en la simulación y una señal de finalización de simulación y desconexión.



Como protocolo, la interacción a través de cliente/servidor comienza especificando a SUMO que se desea trabajar de este modo. En ese momento, SUMO se inicializa en modo servidor dejando abierto un puerto TCP para la conexión del cliente (figura 28 (a)).

Figura 29: Concepto arquitectural de la plataforma TraaS. La plataforma se conecta como cliente a SUMO y ofrece un API basado en SOAP de mensajes que traduce en mensajes del protocolo TraCI, lo que independiza completamente la elección de lenguaje de programación a la vez que abstrae los detalles del protocolo de comunicación.



Una vez el servidor se encuentra en ese estado, el cliente se conecta enviando una señal de conexión indicando que él se encargará de controlar la simulación. Desde ese momento y hasta que el cliente no envíe una señal de desconexión (figura 28 (c)), el cliente podrá enviar y recibir todos los mensajes que desee para capturar información y modificar los detalles de la simulación, incluido el mensaje *step*, que es el encargado de avanzar un paso en la simulación (figura 28 (b)).

Como librería, TraCI es un módulo desarrollado en Python 2.7. Si bien es cierto que es posible trabajar directamente con el protocolo

de comunicación a través de sockets, una librería abstrae todos los detalles dando una interfaz de trabajo más clara y sencilla. Por ello, aunque no se usarán en la tesis, existen otras dos implementaciones que merece la pena mencionar:

- **TraCI4J**³¹. El homólogo de la librería de abstracción de Python pero para el lenguaje Java. Está desarrollada por un tercero.
- **TrasS**³². Una plataforma ofrecida como SaaS que proporciona una interfaz de servicios web bajo protocolo SOAP para abstraer el protocolo en mensajes HTTP (figura 29).

³¹ <https://github.com/egueli/TraCI4J>.

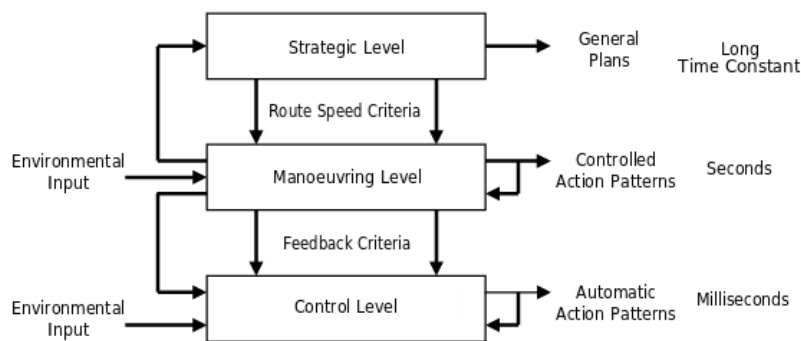
³² <http://traas.sourceforge.net/cms/>.

Modelos de comportamiento

El objetivo que perigue la simulación de tráfico es hacer cada vez más realistas los modelos generados. En un modelo basado en MAS³³, el realismo aumenta cuanto más se asemeja el comportamiento de un agente al de un conductor real.

¿A qué nos referimos cuando hablamos del comportamiento? Según el diccionario de la real academia, “comportar” se define como *actuar de una manera determinada*. Sin embargo, no queda claro qué factores constituyen el comportamiento al volante, es decir.

Conducir implica la ejecución de múltiples tareas en paralelo, cada una de diferente nivel cognitivo. Además, las acciones no está limitadas a la interacción con el vehículo; el conductor ha de tener en cuenta otros factores como, por ejemplo, la situación de tráfico en la que se encuentra inmerso, las señales, los peatones o los dispositivos ITS.



³³ Donde cada uno de los agentes trabaja de manera independiente sobre el mismo problema, esto es, el tráfico.

Figura 30: Los tres niveles jerárquicos que describen la tarea de conducción según [Michon, 1985]: *estrategia* (i.e. las decisiones generales), la *maniobra* (i.e. decisiones durante la conducción de más corto plazo) y *control* (i.e. comportamientos automáticos). (TODO! Rehacer la figura nosotros, esta no me gusta. Es del [Michon, 1985]).

Michon en [Michon, 1985] divide en tres los niveles de abstracción de las tareas: el de **control**, que se ocupa de las tareas de más bajo nivel destinadas éstas a mantener la conducción como son la aceleración o los cambios de marcha, el de **maniobra** o táctico, donde sus tareas son las encargadas de mantener la interacción con el entorno como los cambios de carril o el control de las señales y demás estímulos externos, y el **estratégico**, que engloba las tareas de más alto nivel como el razonamiento y la planificación de rutas (ver figura 30).

El comportamiento de un conductor al volante tiene una relación directa con el nivel de abstracción de maniobra o táctico. Ésta se puede concebir como la encargada de planificar acciones a corto pla-

³⁴ No obstante algunos autores (e.g. **TODO!** Observation-based lane-vehicle-assignment hierarchy for microscopic simulation on an urban street network o [Toledo et al., 2003]) han demostrado que en ocasiones, la planificación de la ruta sí afecta a la preferencia de un conductor por un carril determinado.

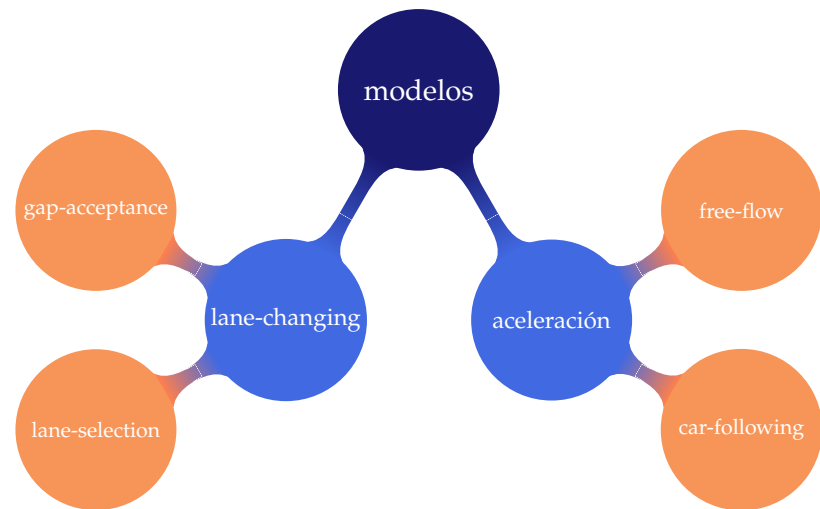
zo para conseguir objetivos a corto plazo. Las tareas de control son automáticas, y influyen poco o nada en las tomas de decisión relacionadas con tareas como cuánto acelerar en esta situación y cuándo cambiar de tráfico. Las tareas estratégicas estan a un nivel más alto de abstracción (e.g. la ruta a seguir hasta mi destino) y tampoco afectan demasiado al comportamiento en situaciones concretas³⁴.

En este capítulo se explorará qué modelos de interés dentro de la CI se han desarrollado hasta el momento y cuáles son sus limitaciones.

Tipos de maniobra

En el nivel táctico del comportamiento, las tareas que se realizan durante la conducción son las relacionadas a circular con el vehículo dentro del flujo de tráfico, sin bajar demasiado de detalle. En la literatura, estas tareas se centran en dos clases diferentes (figura 31): las de **aceleración**, encargadas de modelar el comportamiento de un conductor en el carril en el que se encuentra y las de **cambio de carril**, encargadas de decidir y ejecutar los cambios de carril.

Figura 31: Las diferentes tareas para modelar el comportamiento de un conductor al volante. Están clasificadas en dos tipos, de aceleración, encargadas de definir cómo acelera el conductor en casos con y sin vehículo delantero y de lane-changing, encargadas de las tareas relacionadas con el cambio de carril.



Modelos de aceleración

Se ocupan de gestionar el comportamiento del conductor sobre la aceleración (positiva o negativa) en un entorno lineal como lo es un carril de tráfico.

Se identifican dos comportamientos diferenciados dentro de este: el de circulación sin impedimento, donde el comportamiento del conductor viene motivado por la velocidad que quiere alcanzar sin impedimentos (más allá de los impuestos por el tipo y condición de la vía) y el de circulación con impedimento, donde el comportamiento se ve influenciado por el co-

che delantero en el mismo carril, obligando a disminuir la velocidad deseada en el conductor en cuestión.

LOS MODELOS DE ...

SI LA VELOCIDAD ESTÁ CONDICIONADA a la de un vehículo delante-ro hablamos de una situación . Se dice que el vehículo está siguiendo (*following*) a otro vehículo si, en el caso de ir a la velocidad deseada, éste acabaría colisionando con el vehículo seguido³⁵.

Los diferentes modelos de existentes basan su funcionamiento en relacionar la velocidad del vehículo con algún factor medible. Se identifican cuatro factores diferentes en los que están basados los modelos de ([Aghabayk et al., 2015] y [Toledo, 2007]:

- **Espacio.** Basan su funcionamiento en mantener una distancia segura con el vehículo delantero, normalmente variable en función de la velocidad ([Pipes, 1953], **TODO!**Nonlinear effects on the dynamics of car following). El problema por el que el comportamiento exhibido en estas relaciones no se corresponde con el comportamiento real se atribuye a que un conductor reacciones a muchos más estímulos.
- **Estímulo-respuesta.** La respuesta está directamente relacionada con el estímulo creado por el comportamiento del coche de delante, generalmente una respuesta de tipo aceleración o deceleración aplicada con un pequeño retardo τ ([Chandler et al., 1958]). Algunas modificaciones sobre el algoritmo original son la asimetría en la tasa de cambio de aceleración y deceleración (**TODO!**Car following and steady-state flow) o la inclusión de segundos coches delanteros (**TODO!**Car following theory of steady state traffic flow, **TODO!**An extended model for car following). Los métodos de esta clase de modelo suelen ser sencillos, pero tiene dos problemas principales, (i) que los modelos no capturan los comportamientos de diferentes tipos de conductores o vehículos y (ii) supone que el conductor es capaz de observar los más mínimos cambios en el coche delantero cuando esto en realidad no es así.
- **Psicofísicos.** Éstos asumen que los cambios se perciben a partir de un determinado umbral, a diferencia del resto de modelos vistos hasta ahora que varían en función del cambio de otras medidas, independientemente de los mínimas que sean. Introducen el concepto de *umbral perceptual* de una medida como el límite a partir del cual se observa un cambio en dicha medida.

³⁵ Más formalmente, un modelo de este tipo es una serie de ecuaciones y reglas que actualizan las propiedades de un vehículo n a lo largo del tiempo, siendo éste representado por una tupla de la forma (x_n, v_n, a_n) donde x_n es la localización espacial, v_n la velocidad y a_n la aceleración.

Modelos de cambio de carril

El tráfico real no está compuesto por un sólo carril, sino por varios, donde los adelantamientos son necesarios para evitar congestiones y

donde se necesitan cambios de carril para acceder a posiciones más ventajosas para llegar a un destino concreto.

Los modelos de cambio de carril o se ocupan de determinar cuándo un vehículo quiere desplazarse de un carril a otro (denominado **lane-selection**) y de la ejecución del mismo (denominado **merging**). Además, en la literatura, se identifican dos tipos diferentes de cambios de carril ([Gipps, 1986], [Yang and Koutsopoulos, 1996], [Toledo et al., 2003]):

- **Mandatory.** Ejecutados cuando es obligatorio abandonar el carril actual o acceder al carril objetivo.
- **Discretionary.** cuando el cambio se ve motivado para mejorar la situación actual de conducción

Tiene pinta de que estos, por separado, complicado. Seguramente se queden en “lane-change”. El lane-execution lo he encontrado por ahí como “merging”, y parece que lo mismo hay cosa para hablar por ahí.

LOS MODELOS DE ...

LOS MODELOS DE ...

Se suele apoyar en gap-acceptance models, donde los vehículos calculan si caben o no en un hueco (critical gap) en función de parámetros como, por ejemplo, la velocidad relativa con los vehículos delantero y trasero del carril al que cambiar (Modeling Integrated Lane-Changing Behavior, Toledo et al., 2003).

(Gipps, 1986, A Model for the Structure of Lane-Changing Decisions) propone un framework para el problema de cambio de carril que incluye numerosos factores, entre ellos señales de tráfico, tipos de vehículo (e.g. camiones) o “urgencia” en el cambio de carril (e.g. proximidad a una salida o giro). El principal problema de las reglas de este modelo es que asumen que el cambio de carril ocurre sin forzar a los vehículos del carril de destino a modificar su comportamiento como disminuir la velocidad o parar.

En (Fritzsche, 1994, A model for traffic simulation) se describe un modelo de microsimulación para analizar cuellos de botella (e.g. un accidente donde se bloquea uno de los carriles). Es un caso típico donde los vehículos no pueden cambiar de carril sin la participación activa del resto de vehículos (colaboración). El modelo lo describe de una manera muy sucinta y no considera comportamiento colaborativo en el cambio de carril.

(Modelling lane utilisation on British dual-carriageway roads: effects on lane-changing) developed a microscopic simulation model

for the investigation of lane changing behaviour on multi-lane uni-directional roadways. The rules pertaining to the desire and the possibility to change lane are based on similar logic to that described by Gipps (1986). Again, the assumption of the model is that if the available gap in the target lane is smaller than a given acceptable limit, no lane changing will take place. The main concern of the study is the relationship between lane utilisation and traffic flow on dual-carriageway roads under normal flow conditions (i.e. without incidents) and the model is adequate for this purpose. However, it could not produce realistic results when incidents or lane closures affect the flow conditions.

(Barcelo et al., 1996, PETRI: A parallel environment for a real-time traffic management and information system) describen el simulador AIMSUN. El comportamiento de cada vehículo en la simulación es modelado a través de múltiples modelos de comportamiento (e.g. car following, lane changing, gap acceptance). El modelo de cambio de carril es el usado en el modelo de Gipps, aunque el propio simulador permite la modelización de incidentes por lo que debe existir alguna variación del modelo original o un nuevo modelo para ese caso concreto.

(Wagner et al., 1997, Realistic multi-lane traffic rules for cellular automata) describe "un modelo de microsimulación mínimo para reproducir características macroscópicas en el flujo de tráfico". El objetivo: definir reglas realistas para modelar el cambio de carril usando carreteras con múltiples carriles. Para el cambio de carril, describen una serie de reglas para describir "cuándo" cambiar de carril, y una "regla restrictiva de seguridad" que especifica que un coche que quiere cambiar de carril no moleste al coche de detrás en el carril objetivo. El modelo fue capaz de reproducir de forma satisfactoria las características de uso de los carriles en carreteras de múltiples carriles con diferentes flujos de tráfico sobre condiciones normales (i.e. sin incidentes) de tráfico.

En (Hunt and Lyons, 1994, Modelling of dual-carriageway lane-changing using neural networks) desarrollan un modelo de decisión usando redes neuronales. El modelo funciona a partir de presentarle una entrada visual del entorno alrededor del vehículo que quiere cambiar de carril. Sin embargo, no considera la cooperación entre vehículos.

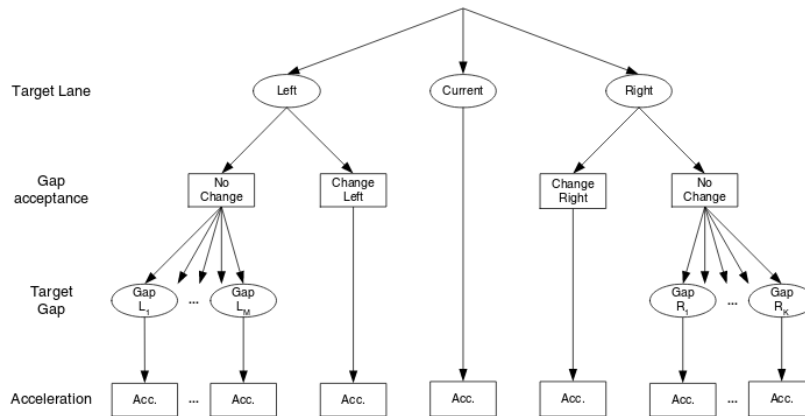
En (Yang and Koutsopoulos, 1996, A Microscopic Traffic Simulator for Evaluation of Dynamic Traffic Management Systems) presentan el simulador MITSIM, desarrollado por el MIT (creo) en el que se habla específicamente de comportamiento colaborativo en cambio de carriles haciendo uso de lo que denominan *courtesy yielding function* (algo así como función de cesión de paso de cortesía), la cual se usa para hacer espacio al vehículo que va a incorporarse al carril. Sin embargo, los detalles de dicho proceso no están especificados en el paper.

En (Modelling lane changing and merging in microscopic traffic simulation, 2002) (simulador SITRAS) dicen que según el comportamiento de (Gipps, 1986), el cambio de carril no ocurre nunca en una situación de congestión, y por tanto en este tipo de situaciones el vehículo tiene que forzar su movimiento hacia el carril. Como la interacción entre conductores en ese tipo de maniobras requiere comportamiento complejo de toma de decisiones, éstos pueden ser modelados con técnicas de agentes autónomos. También aquí hablan de los DVOs (driver-vehicle objects) (¿esto es un clon de DVUs?. Tienen características individuales como (i) tipo de vehículo, (ii) magnitudes físicas (tamaño, velocidad máxima, ...), (iii) tipo de conductor y (iv) nivel de conocimiento de la red (porque afecta en la elección de ruta). Tienen un objetivo, llegar del origen al destino tan rápido como puedan. Esto implica un conjunto de decisiones a hacer en intervalos periódicos durante su funcionamiento (i) selección de ruta cada vez que se entra en un nuevo tramo y (ii) cálculo de la aceleración en cada intervalo).

En (Chaib-draa and Levesque, 1996,) proponen un framework para trabajar con tres tipos diferentes de situaciones (rutina, familiar y no familiar) en sistemas multiagente, demostrando la aplicación en escenarios de microsimulación urbana). Su modelo se basa en una estructura jerárquica definida por los niveles de comportamiento humano y de técnicas de razonamiento propuestas por (Rasmussen, 1986,) (skill-rule-knowledge). El comportamiento basado en habilidades (skills) se refiere a las actividades completamente automatizadas (percepción-ejecución) usadas típicamente en situaciones rutinarias. comportamiento basado en reglas se refiere a situaciones estereotipadas (percepción-reconocimiento de la situación-planificación-ejecución) aplicable en su mayoría en situaciones familiares. El comportamiento basado en el conocimiento se refiere a actividades conscientes que implican trabajo de resolución de problemas y toma de decisiones (perception-reconocimiento de la situación-toma e decisión-planificación de la ejecución) que suelen ser necesarias en situaciones poco familiares. En SITRAS se ven estas diferencias claramente en el cálculo de la aceleración. (i) si no hay ninguna otra restricción, llegar a la máxima velocidad es acelerar hasta la máxima velocidad (skill), (ii) si hay una luz roja más adelante, se va frenando hasta para el coche (rule) y (iii) si se recibe información de coches alrededor (por ejemplo se está incorporando un nuevo coche a nuestro carril) se requiere un conocimiento más complejo (knowledge). Los DVO aquí tienen las siguientes debilidades: no tienen memoria (sólo planean el segundo siguiente de acuerdo a la información actual) y tienen poco contacto directo con los demás DVOs de alrededor (saben del de delante y del de detrás, pero no de los lados).

Modelos mixtos

Los modelos de aceleración y de cambio de carril han sido tratados tradicionalmente como modelos independientes, siendo estos primeros mucho más estudiados que los segundos³⁶.



³⁶ Este hecho es motivado por la dificultad en la captura de datos en los cambios de carril y, por tanto, por su escasez. Fuente: [Toledo et al., 2007]. El agente comprueba constantemente si se desea o no cambiar de carril, y una vez decidido comprueba la viabilidad. En cualquier caso, toda decisión finaliza con la comprobación de la aceleración. Fuente: [Toledo et al., 2007]. **TODO!** Quizá la imagen es muy grande y quedaría mejor si la hiciésemos más pequeña.

Sin embargo, a partir de los años 90 se ha tendido al desarrollo de modelos de comportamiento que combinan los modelos de aceleración y de cambio de carril ([Ma, 2004]). Un ejemplo actual de este tipo de modelos es el descrito en [Toledo et al., 2007], el cual se basa en el concepto de “objetivo a corto plazo” para elaborar un “plan a corto plazo” apoyándose en un árbol de decisión que determina la acción a realizar (ver figura 32).

...

La Computational Intelligence en los modelos de conducción

La mayoría de los modelos de comportamiento existentes basan su funcionamiento en técnicas pertenecientes al Hard Computing; es decir, están basados en fórmulas matemáticas y reglas de la lógica convencional con parámetros que se ajustan a partir de la observación de datos reales. Sin embargo, desde mediados de los años 90 empezó a crecer el interés por las técnicas de la CI.

...

Según [Aghabayk et al., 2015], los modelos basados en Inteligencia Artificial funcionan en base a dos técnicas principalmente, las Redes Neuronales Artificiales y la Lógica Difusa.

Modelos basados en lógica difusa

Los modelos de car following que se basan en la lógica difusa suelen apoyarse en la convicción de que la información que maneja el conductor a la hora de tomar decisiones proviene de un análisis no

³⁷ El modelo GHR es el modelo más conocido antes de la introducción del modelo de Gipps. Desarrollado a finales de los años 50, calcula el valor de la aceleración a en un instante t como:

$$a(t) = cv^m(t) \frac{\Delta v(t - \tau)}{\Delta x^l(t - \tau)} \quad (3)$$

Siendo t es el instante actual, $a(t)$ la aceleración del vehículo, $\delta v(t)$ y $\delta x(t)$ son la velocidad y distancia relativas al siguiente coche respectivamente, v la velocidad del vehículo y c, m, l y τ constantes, siendo ésta última el tiempo de reacción del conductor.

demasiado detallado de la situación que le rodea y que, por tanto, su razonamiento parte de conceptos imprecisos y vagos que conducen a respuestas no demasiado bien definidas.

[Kikuchi and Chakroborty, 1992] fueron los primeros en aplicar lógica difusa a modelos de conducción. Usaron el modelo Gazis-Herman-Rothery (GHR)³⁷ ([Chandler et al., 1958]). Las entradas al modelo eran distancia al coche delantero, diferencia de velocidades, aceleración y deceleración del coche delantero. La aceleración y deceleración se toman como entradas diferentes porque postulaban que el comportamiento antes ambos casos era diferente (aunque creo que esto es equivalente a usar particiones no simétricas de la variable lingüística. Como salida, la aceleración/deceleración del coche actúa

Otros trabajos que trabajan con lógica difusa: (Evaluation of the General Motors based car-following models and a proposed fuzzy inference model, Calibrating the membership functions of the fuzzy inference system: instantiated by car-following data, A FUZZY LOGIC MODEL OF FREEWAY DRIVER BEHAVIOR, The Modeling and Simulation of the Car-following Behavior Based on Fuzzy Inference, Fuzzy parameters estimation for car-following modelling, A Fuzzy Logic Approach for Car-Following Modelling, Variable response time lag module for car-following models, Development of a fuzzy logic based microscopic motorway simulation model, Establishment of car following theory based on fuzzy-based sensitivity parameters. Advances in multimedia modelling, Application of fuzzy systems in the car-following behaviour analysis, The validation of a microscopic simulation model: A methodological case study). Ninguno de estos estudios considera diferentes tipologías de vehículos para el car following

Problemas:

1. ¿Qué reglas usan los humanos para modelar su comportamiento? Desconocerlas implica modelos no realistas. En (The validation of a microscopic simulation model: A methodological case study) intenta suplir este problema con encuestas a conductores.
2. Los problemas inherentes de los controladores difusos. ¿Cómo validar las funciones de pertenencia? ¿cómo determinar las reglas difusas?

Modelos basados en redes neuronales artificiales

Las redes neuronales se han aplicado mucho sobre el campo de las ITS en general, y sobre la conducción autónoma y el análisis del comportamiento de los conductores (A review of neural networks applied to transport).

En (Modelling human performance with neural networks) implementaron un controlador basado en redes neuronales para el com-

portamiento del car following en un microsimulador entrenando dicho modelo previamente con datos extraídos de un conductor en dicho simulador.

En (The use of neural networks to recognise and predict traffic congestion) usan redes para determinar el nivel de congestión en la vía.

(Develop a car-following model using data collected by 'five-wheel system') son los primeros en usar datos reales de un coche instrumentado usando el método Five-Wheel-System, que está especificado en su paper de aquella manera y que no me entero de nada. A partir de las entradas correspondientes a velocidad relativa, espacio relativo, velocidad y velocidad deseada (para ello, clasifican al conductor de agresivo, normal, conservador) determinan la aceleración/deceleración del vehículo. No lo aplican a ningún simulador, sólo que los valores se ajustan.

(Neural agent car-following models) redes neuronales usando el dataset de (Traffic simulation supporting urban control system development) desarrollan una red neuronal para mantener la distancia con el siguiente vehículo. Este modelo sí se evaluó en el simulador AIMSUN, y los resultados muestran una buena correspondencia entre los datos y la realidad. No replica sin embargo el comportamiento de frenar hasta parar o de acelerar desde parado.

Problemas:

1. Es imposible determinar por qué la red funciona como está funcionando.
2. Los clásicos de los problemas de redes, el no aprendizaje y la especialización.

Otras técnicas

Aproximaciones híbridas

(Simulation of car-following decision using fuzzy neural network system) y (Toward an integrated car-following and lane-changing model based on neural-fuzzy approach) usan aproximaciones de fuzzy neural networks y neuro-fuzzy respectivamente. No proveen sin embargo de documentación y no se investiga la aplicación de estos modelos a microsimuladores de tráfico.

(Exploring a local linear model tree approach to car-following) usan el modelo de árboles lineales locales (LOLIMOT, (Nonlinear system identification: From classical approaches to neural networks and fuzzy models)) que no deja de ser una aproximación neuro-fuzzy del comportamiento. Intenta incorporar imperfecciones perceptuales en un modelo de car following. El modelo está basado

usando datasets reales y los resultados indican que se ajusta lo predicho con la realidad, pero no hay pruebas realizadas en microsimuladores.

En [Ma, 2004], bajo la suposición de que el ser humano toma múltiples decisiones relacionadas basándose en su percepción (imprecisa) del entorno propone un método masado en un sistema de inferencia difusa para tomar decisiones tanto para el problema del *car following* como para el del *lane changing*, calibrando y ajustando dicho controlador mediante el uso de redes neuronales (aproximación Neuro-Fuzzy).

Por procesar

SUMO usa (al menos así lo indican en el paper del 2002) el modelo Gipps[Krajzewicz et al., 2002]. No sé si ellos han hecho una extensión del modelo o están referenciando la extensión y ellos sólo la implementan. En el paper del 2012 citan que el modelo car-following que usan por defecto es el desarrollado por Stefan Krauß[Jin et al., 2016], debido a su simplicidad y su velocidad de ejecución. El modelo ha probado ser válido, pero tiene algunos defectos, por lo que existe un API para implementar otros modelos. En la actualidad están incluidos en el sistema los modelos IDM[Treiber et al., 2000] (*Intelligent Driver Model*), el modelo de tres fases de Kerner[Kerner et al., 2008] y el modelo de Wiedemann[Wiedemann, 1974].

En [Sekizawa et al., 2007] describen modelos supervisados offline para capturar el comportamiento del conductor basados en auto-regresión a trozos. Más adelante lo extienden en [Terada et al., 2010], aunque los datos de entrenamiento son extraídos directamente de simulaciones, no del “mundo realTM”.

En [Bando et al., 2013] describen otro modelo no supervisado offline basado en un modelo bayesiano no paramétrico para la clusterización, combinado con un LDA (Latent Dirichlet Allocation, sea lo que sea esto) para la clusterización a más alto nivel. (TODO! ¿este método usa datos reales?)

Estos trabajos ([Sekizawa et al., 2007], [Terada et al., 2010] y [Bando et al., 2013]) tienen la desventaja de ser computacionalmente muy costosos y con poca precisión en el caso de variar mucho los escenarios de entrenamiento y de test.

En [Maye et al., 2011] se presenta un modelo online donde se infiere el comportamiento del conductor haciendo uso de una IMU (Intertial Measurement Unit) y una cámara. Primero de la IMU se sacan datos que se separan en fragmentos para luego relacionarlos con las imágenes obtenidas de la cámara. (TODO! ¿Hacia dónde mira la cámara?). Los modelos propuestos en [Johnson and Trivedi, 2011] y [Van Ly et al., 2013] también se apoyan en el funcionamiento de

clasificar las segmentaciones de una IMU, pero con técnicas distintas y sin cámara. Además hacen uso de señales externas y umbrales de activación para hacer más efectiva la clasificación (**TODO!** corroborar).

En el artículo [Bender et al., 2015] se usa también un modelo no supervisado online con una aproximación bayesiana para identificar los puntos de cambio sin depender de parámetros externos (e.g. umbrales o señales). Se basa también en (1) segmentar los datos de conducción y (2) asignar estos datos a clases que se corresponde n con comportamientos de conducción. Tiene la ventaja de ser más eficiente y rubusto que los anteriores.

La idea de estos métodos desde el [Sekizawa et al., 2007] hasta el [Bender et al., 2015] creo que es el de un sistema que traduzca datos en crudo a datos de más alto nivel. Esto es debido a que la cantidad de datos que se pueden generar en un sólo coche (no digamos ya una flota de ellos) es tan grande que para determinados sistemas disponer de información de más alto nivel haría más sencillo su desempeño (por ejemplo, Sistema Avanzado de Ayuda a la Conducción (ADAS, Advanced Driver Assistance System) que funcionasen con datos de “adelantando” que sus valores de giro, aceleración en una ventana temporal).

En [Satzoda et al., 2013], haciendo uso de la información combinada de bus CAN, cámaras, GPS e información digitalizada el mapa donde se circula se determina una amplio abanico de información crítica en diferentes condiciones de la carretera. La información que sacan es: número de cambios de carril a la derecha, a la izquierda, tiempo en autopista y carretera urbana, distancia recorrida, velocidades medias en autopista y urbano, paradas, giros a la derecha, a la izquierda, incorporaciones y salidas de autopsta, tiempo gastado en un sólo carril, curvas a la izquierda, curvas a la derecha y distancia media al carril central

En [Al-Shihabi and Mourant, 2001] describen un framework para la modelización de comportamiento de conductores dentro de simuladores. Se basa en cuatro unidades de funcionamiento interconectadas, la de percepción (percibe el entorno en términos locales y globales), la de emoción (cómo responde emocionalmete al entorno), la de decision-making (investiga posibles acciones que podrían servir a las necesidades del módulo emocional) y la de decision-implementation (intenta implementar las decisiones cuando emergen condiciones de tráfico lo suficientemente seguras para llevarlas a cabo). Tengo que volver a leerlo después de hacer una primera introducción en el tema de agentes, porque me parece poner nombrecitos a un tipo de agente que funciona de esa misma manera, pero lo mismo no.

En [Terroso-Sáenz et al., 2015] analizan lo que ellos denominan el concepto del Reconocimiento del Entorno Intravehicular (IvCA, Intra-vehicular Context Awareness), lo que viene a ser el contexto definido

³⁸ Un CEP es un método por el cual se lee un flujo de información compuesta de flujos de distintas fuentes (de ahí el *complex*) para detectar eventos o patrones que pueden indicar la presencia de situaciones a analizar lo más rápido posible.

dentro del vehículo, llegando a intentar predecir no sólo el número de ocupantes (ese es un problema sobradamente superado) sino la tipología o clase de ocupante (e.g. niños, adultos, ... **TODO!** buscar cuáles son las tipologías). Para ello hacen uso de un **Procesamiento Complejo de Eventos (CEP, Complex Event Processing)**³⁸ para procesar los datos de los diferentes sensores del vehículo y así detectar y analizar patrones característicos.

El artículo [Muñoz et al., 2010] es muy interesante ya que para la competición *2010 Simulated Car Racing Championship* desarrollaron un controlador no para minimizar el tiempo en realizar las carreras, sino para hacerlo lo más parecido posible a cómo se comporta un humano. Para ello hicieron uso de redes neuronales para calcular trayectorias y de un proceso de aprendizaje a partir de información extraída de un conductor real en el simulador TORCS.

Aunque TORCS es usado como entorno de simulación en diversos concursos e investigaciones, se trata de un juego, y es que los juegos son un sandbox perfecto como entorno de simulación, ya que presentan una abstracción del dominio sobre el que trabajar. Otros trabajos en esta línea (entrenamiento de redes neuronales en este simulador) son los de [Munoz et al., 2009] y [Van Hoorn et al., 2009], el primero entrenando perceptrones multicapa (**TODO!** verificar) con un back-propagation haciendo uso de un dataset proporcionado por un bot y otro haciendo uso de un algoritmo evolutivo multiobjetivo para optimizar la red de acuerdo a un conjunto de datos proporcionado por un conductor real. Sin embargo, este tipo de modelos se encuentran más cercanos al nivel de control que al nivel de maniobra descritos en la figura 30.

En [Van Ly et al., 2013] hacen uso de los sensores de inercia de un coche (a saber qué coche tienen) para construir un perfil de conductor. Concluyen que frenar y girar caracterizan mejor a los conductores frente a acelerar.

En [Das et al.,] se realiza una simulación de comportamiento de vehículos en autopista. Los agentes basan su comportamiento en un sistema difuso donde las reglas definen la conclusión en autopista (i.e. car-following y lane-changing). Llamam a este simulador AASIM (Autonomous Agent Simulator),.

En [Dia, 2002] Los parámetros relativos al comportamiento, es decir, los que definen características, forma de razonar, etcétera son extraídos de encuestas a conductores reales. De acuerdo a los autores, cada conductor con sus propias características su forma de razonar sus percepciones y sus objetivos se puede modelar como un agente.

En [Ehlert and Rothkrantz, 2001a] (ver si este otro paper del autor cuenta lo mismo y me quito de una de las dos referencias: [Ehlert and Rothkrantz, 2001a]) se realiza una simulación donde los agentes son de tipo reactivo. Además, poseen diferentes estilos de conducción. El agente continuamente va realizando decisiones de control para mantenerse en la vía y llegar a su

destino.

En [] definen dos capas, “física” y “mental”. La física es aquella donde los agentes se mueven de acuerdo a sus magnitudes físicas, evitando accidentes y, en resumen, comportándose al volante. La mental es la que se encarga de las estrategias a más alto nivel, como por ejemplo la actividad planeada o la ruta escogida. Los parámetros de los agentes y de la simulación son extraídos de censos y encuestas.

Parte III

Desarrollo de la tesis

Sistemas desarrollados

Este capítulo describe todos los sistemas y el software desarrollados e implementados para realizar la tesis. Éstos son tanto los encargados de la captura de datos de los conductores, los que trabajan directamente con el simulador para integrar los controladores generados y los desarrollos para la generación de Software.

ScanBUS

ScanBUS es un software para la identificación de paquetes enviados por dispositivos a través del Bus CAN del vehículo.

Sistema para la captura de datos multidispositivo

Para la obtención de los datos de conducción se ha desarrollado un sistema que permite la conexión a múltiples dispositivos desde diferentes interfaces. Las razones para su desarrollo son las siguientes

- Sincronización automática de datos de dispositivo en intervalos configurables de tiempo. El sistema permite la configuración de la frecuencia de captura sincronizando los datos recibidos a esa frecuencia.
- Diseño extensible a otros dispositivos. El software está diseñado para facilitar en la medida de lo posible la introducción de nuevos dispositivos usando, para ello, las interfaces apropiadas.
- Hardware compacto. El sistema está integrado en un ordenador de tipo Raspberry PI, aunque es factible su integración en otros sistemas siempre y cuando funcionen con un sistema GNU/Linux e incluyan el hardware necesario para las capturas.

Biblioteca para la incorporación de modelos de conductor personalizados en SUMO

Hace uso de [TraCI](#).

Modelos de comportamiento

Entrenamiento de controladores difusos mediante [EC](#)

Estudio de modelos de comportamiento

Creo que además de crear modelos adaptados al conductor de car following y de lane change, estaría bien adaptar los parámetros de los modelos existentes y comparar.

Cosas que se me ocurren. Habría que tener un módulo que se plantea qué acción tomar en función del entorno y qué acciones ocurren alrededor nuestro. Quiero cambiar de carril, está entrando un usuario al carril,

El vehículo como agente inteligente

En nuestra tesis el trabajo se basa en la representación del comportamiento de un conductor. Para simplificar el problema asumiremos que los términos “conductor” y “vehículo” son equivalentes y que se refieren a la tupla “(vehículo, conductor)” (Hay un paper que lo denomina DVU como driver-vehicle-unit [[Dia, 2002](#)])

No sé cómo ponerlo, pero esta información puede ser útil aquí. A lo mejor hay que separarlo, no sé. Yo lo pongo aquí porque me ha venido ahora.

- **Percepción.** Característica intrínseca a todo ser vivo para la obtención de información del entorno. Pues para los agentes inteligentes, lo mismo. Puede ser en forma de sensores (e.g. sensor de velocidad, acelerómetro, sensor de distancia, lidar, cámaras, termómetros, gps, ...), conocimiento (e.g. controlador difuso, sistema experto, red neuronal, es decir, información procesada que ad valor añadido), datos GIS, comunicación con otros vehículos (e.g. V2V, V2E, ...). Un simulador nos ofrece estos sensores casi gratis, pero nuestro problema es que para la generación de modelos necesitamos previamente haber capturado información con dichos sensores y en físico ya es otra cosa.
- **Toma de decisiones.** Un agente racional en un sistema multiagente ha de ser capaz de razonar acerca del mundo, de su propio estado y del estado del resto de agentes.

- **Actuación.** La actuación es la consecuencia natural de las anteriores características. Dado un estado del mundo y un proceso cognitivo surgen acciones a realizar, tras las cuales se llega a un nuevo estado del entorno que provee de información actualizada para seguir actuando.

Resultados

Factores a mirar del tráfico. En un paper ponen esto: the simulation results indicate that the proposed model is explainable, and it can represent equilibrium and disequilibrium states of microscopic and macroscopic traffic, such as: stable traffic, unstable traffic, equilibrium speed-flow relationship, closing-in, shying-away, capacity drop, and traffic hysteresis.

Realizar comparativa de modelos existentes contra modelos de parámetros ajustados contra modelos creados.

Resultados

Conclusiones

¿Pierde rendimiento el sistema cuando se aplican los modelos a escenarios significativamente diferentes de los escenarios de test? Si sí, un trabajo futuro y algo para escribir en conclusiones sería hablar de este defecto y de cómo subsanarlo.

Aportaciones

Futuras líneas de investigación

¿A lo mejor se podría tirar por el campo de las V2X desde esta tesis?

¿Entrar en el tema de la mesosimulación?

Intersection model (lo he visto nombrar por primera vez en http://elib.dlr.de/89233/1/SUMO_Lane_change_model). Habrán creado también un concepto así rotondas?

No sé, pero me parece lógico tratar de realizar una disociación entre vehículo y conductor en lugar de contemplar el binomio vehículo/conductor como uno sólo. Es más, creo que resultaría interesante evaluar comportamientos de conductores sobre diferentes tipologías de vehículos. La librería de todas formas debería soportar esta disociación (y se debería indicar).

Bibliografía

- [par, 2010] (2010). Directive 2010/40/eu of the european parliament and of the council of 7 july 2010 on the framework for the deployment of intelligent transport systems in the field of road transport and for interfaces with other modes of transport text with eea relevance. *Official Journal of the European Union*, 50:207.
- [Aghabayk et al., 2015] Aghabayk, K., Sarvi, M., and Young, W. (2015). A state-of-the-art review of car- following models with particular considerations of heavy vehicles. *Transport Reviews*, 35(May 2015):37–41.
- [Al-Shihabi and Mourant, 2001] Al-Shihabi, T. and Mourant, R. R. (2001). A framework for modeling human-like driving behaviors for autonomous vehicles in driving simulators. *The 5th International Conference on Autonomous Agents.*, (June):286–291.
- [Artyukhin and Avery, 2014] Artyukhin, A. and Avery, L. (2014). Collective behavior of *C.elegans* and *C.briggsae* starved L1 larvae. *The Worm Breeder's Gazette*, 20(1):19–20.
- [Balmer et al., 2004] Balmer, M., Cetin, N., Nagel, K., and Raney, B. (2004). Towards truly agent-based traffic and mobility simulations. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, 1:60–67.
- [Bando et al., 2013] Bando, T., Takenaka, K., Nagasaka, S., and Taniguchi, T. (2013). Unsupervised drive topic finding from driving behavioral data. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 177–182. IEEE.
- [Barlovic et al., 1998] Barlovic, R., Santen, L., Schadschneider, A., and Schreckenberg, M. (1998). Metastable states in cellular automata for traffic flow. *Eur. Phys. J. B*, 5:793–800.
- [Behrisch et al., 2011] Behrisch, M., Bieker, L., Erdmann, J., and Krajzewicz, D. (2011). Sumo-simulation of urban mobility: an overview. In *Proceedings of SIMUL 2011, The Third International Conference on Advances in System Simulation*. ThinkMind.
- [Bender et al., 2015] Bender, A., Agamennoni, G., Ward, J. R., Worrall, S., and Nebot, E. M. (2015). An unsupervised approach

- for inferring driver behavior from naturalistic driving data. *IEEE Transactions on Intelligent Transportation Systems*, 16(6):3325–3336.
- [Bezdek, 1993] Bezdek, J. C. (1993). Intelligence: Computational versus artificial. *IEEE Transactions on Neural Networks*, 4(5):737.
- [Brilon and Wu, 1999] Brilon, W. and Wu, N. (1999). Evaluation of cellular automata for traffic flow simulation on freeway and urban streets. *Traffic and Mobility*.
- [Casas et al., 2011] Casas, J., Perarnau, J., and Torday, A. (2011). The need to combine different traffic modelling levels for effectively tackling large-scale projects adding a hybrid meso/micro approach. *Procedia-Social and Behavioral Sciences*, 20:251–262.
- [Chandler et al., 1958] Chandler, R. E., Herman, R., and Montroll, E. W. (1958). Traffic Dynamics: Studies in Car Following. *Operations Research*, 6(2):165–184.
- [Clymer, 2002] Clymer, J. (2002). Simulation of a vehicle traffic control network using a fuzzy classifier system. In *Proceedings 35th Annual Simulation Symposium. SS 2002*, pages 285–291. IEEE Comput. Soc.
- [Cybenko, 1989] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.
- [Das et al.,] Das, S., Bowles, B., Houghland, C., Hunn, S., and Yun-Long Zhang. Microscopic simulations of freeway traffic flow. In *Proceedings 32nd Annual Simulation Symposium*, pages 79–84. IEEE Comput. Soc.
- [Dia, 2002] Dia, H. (2002). An agent-based approach to modelling driver route choice behaviour under the influence of real-time information. *Transportation Research Part C: Emerging Technologies*, 10(5):331–349.
- [Dijkstra, 1972] Dijkstra, E. W. (1972). The humble programmer. *Communications of the ACM*, 15(10):859–866.
- [Dingus et al., 2006] Dingus, T. A., Klauer, S. G., Neale, V. L., Petersen, A., Lee, S., Sudweeks, J., Perez, M., Hankey, J., Ramsey, D., Gupta, S., et al. (2006). The 100-car naturalistic driving study, phase ii-results of the 100-car field experiment. Technical report.
- [Dresner and Stone, 2004] Dresner, K. and Stone, P. (2004). *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems : AAMAS 2004 : New York City, New York, USA : July 19-23, 2004*. IEEE Computer Society.
- [Du and Swamy, 2006] Du, K. and Swamy, M. (2006). Neural networks in a softcomputing framework.

- [Ehlert and Rothkrantz, 2001a] Ehlert, P. A. and Rothkrantz, L. J. (2001a). Microscopic traffic simulation with reactive driving agents. In *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*, number April, pages 0–5. IEEE.
- [Ehlert and Rothkrantz, 2001b] Ehlert, P. a. M. and Rothkrantz, L. J. M. (2001b). A reactive driving agent for microscopic traffic simulation. *Modelling and Simulation 2001*, pages 943–949.
- [Finin et al., 1994] Finin, T., Fritzson, R., McKay, D., and McEntire, R. (1994). KQML as an agent communication language. *Proceedings of the third international conference on Information and knowledge management - CIKM '94*, pages 456–463.
- [Galis and Rao, 2000] Galis, a. and Rao, S. (2000). -"Application of Agent Technology to Telecommunication Management Services".
- [Gipps, 1986] Gipps, P. G. (1986). A model for the structure of lane-changing decisions. *Transportation Research Part B*, 20(5):403–414.
- [Gu et al., 2015] Gu, M., Billot, R., Faouzi, N.-e. E., Lyon, D., and Hassas, S. (2015). Multi-Agent Dynamic Coupling for Cooperative Vehicles Modeling. pages 4276–4277.
- [Hebb, 1968] Hebb, D. (1968). The organization of behavior.
- [Hinton, 2006] Hinton, G. E. (2006). Reducing the Dimensionality of Data with Neural Networks. *Science*, 313(5786):504–507.
- [Holland, 1975] Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An introductory analysis with applications to biology, control, and artificial intelligence.*, volume Ann Arbor. U Michigan Press.
- [Imprialou et al., 2016] Imprialou, M.-I. M., Quddus, M., Pitfield, D. E., and Lord, D. (2016). Re-visiting crash–speed relationships: A new perspective in crash modelling. *Accident Analysis & Prevention*, 86:173–185.
- [Jin et al., 2016] Jin, J., Ma, X., Koskinen, K., Rychlik, M., and Kosonen, I. (2016). Evaluation of fuzzy intelligent traffic signal control (fits) system using traffic simulation. In *Transportation Research Board 95th Annual Meeting*, number 16-4359.
- [Johnson and Trivedi, 2011] Johnson, D. A. and Trivedi, M. M. (2011). Driving style recognition using a smartphone as a sensor platform. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1609–1615. IEEE.
- [Kerner et al., 2008] Kerner, B., Klenov, S., and Brakemeier, A. (2008). Testbed for wireless vehicle communication: A simulation approach based on three-phase traffic theory. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 180–185. IEEE.

- [Kikuchi and Chakroborty, 1992] Kikuchi, S. and Chakroborty, P. (1992). *Car-following model based on fuzzy inference system*. Number 1365.
- [Krajzewicz et al., 2012] Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. (2012). Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138.
- [Krajzewicz et al., 2002] Krajzewicz, D., Hertkorn, G., Rössel, C., and Wagner, P. (2002). Sumo (simulation of urban mobility) – an open-source traffic simulation. In *Proceedings of the 4th Middle East Symposium on Simulation and Modelling (MESM20002)*, pages 183–187.
- [Krauss et al., 1997] Krauss, S., Wagner, P., and Gawron, C. (1997). Metastable states in a microscopic model of traffic flow. *Physical Review E*, 55(5):5597–5602.
- [Lerner et al., 2010] Lerner, N., Jenness, J., Singer, J., Klauer, S., Lee, S., Donath, M., Manser, M., and Ward, N. (2010). An exploration of vehicle-based monitoring of novice teen drivers. *Final Report. NHTSA, Report No. DOT HS, 811:333*.
- [Ma, 2004] Ma, X. (2004). Toward An Integrated Car-following and Lane- changing Model by A Neural Fuzzy Approach. pages 1–15.
- [Margolus, 1993] Margolus, N. (1993). CAM-8: a computer architecture based on cellular automata *.
- [Maye et al., 2011] Maye, J., Triebel, R., Spinello, L., and Siegwart, R. (2011). Bayesian on-line learning of driving behaviors. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 4341–4346. IEEE.
- [McCarthy et al., 1956] McCarthy, J., Minsky, M., Rochester, N., and Shannon, C. (1956). Dartmouth conference. In *Dartmouth Summer Research Conference on Artificial Intelligence*.
- [McCulloch and Pitts, 1943] McCulloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4):115–133.
- [Michon, 1985] Michon, J. A. (1985). A critical view of driver behavior models: what do we know, what should we do? In *Human behavior and traffic safety*, pages 485–524. Springer.
- [Minderhoud, 1999] Minderhoud, M. (1999). *Supported Driving: Impacts on Motorway Traffic Flow*. PhD thesis.
- [Minsky and Papert, 1969] Minsky, M. and Papert, S. (1969). Perceptrons.
- [Munoz et al., 2009] Munoz, J., Gutierrez, G., and Sanchis, A. (2009). Controller for torcs created by imitation. In *2009 IEEE Symposium on Computational Intelligence and Games*, pages 271–278. IEEE.

- [Muñoz et al., 2010] Muñoz, J., Gutierrez, G., and Sanchis, A. (2010). A human-like torcs controller for the simulated car racing championship. In *Proceedings of the 2010 IEEE Conference on Computational Intelligence and Games*, pages 473–480. IEEE.
- [Munoz et al., 2001] Munoz, L., Gomes, G., Yi, J., Toy, C., Horowitz, R., and Alvarez, L. (2001). Integrated meso-microscale traffic simulation of hierarchical ahs control architectures. In *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*, pages 82–87. IEEE.
- [Nagel and Schreckenberg, 1992] Nagel, K. and Schreckenberg, M. (1992). A cellular automaton model for freeway traffic.
- [Nagel et al., 1998] Nagel, K., Wolf, D. E., Wagner, P., and Simon, P. (1998). Two-lane traffic rules for cellular automata: A systematic approach. *Physical Review E*, 58(2):1425–1437.
- [OICA, 2015] OICA (2015). Motorization rate 2014 – worldwide.
- [Osogami et al., 2012] Osogami, T., Imamichi, T., Mizuta, H., Morimura, T., Raymond, R., Suzumura, T., Takahashi, R., and Idé, T. (2012). IBM Mega Traffic Simulator.
- [Pakkenberg and Gundersen, 1997] Pakkenberg, B. and Gundersen, H. J. (1997). Neocortical neuron number in humans: effect of sex and age. *The Journal of comparative neurology*, 384(2):312–20.
- [Pipes, 1953] Pipes, L. (1953). An operational analysis of traffic dynamics. *Journal of applied physics*.
- [Poslad, 2007] Poslad, S. (2007). Specifying Protocols for Multi-Agent Systems Interaction. *ACM Transactions on Autonomous and Adaptive Systems*, 2(4, Article 15):25.
- [Quaassdorff et al., 2016] Quaassdorff, C., Borge, R., Pérez, J., Lumbreras, J., de la Paz, D., and de Andrés, J. M. (2016). Microscale traffic simulation and emission estimation in a heavily trafficked roundabout in madrid (spain). *Science of The Total Environment*, 566:416–427.
- [Ramón and Cajal, 1904] Ramón, S. and Cajal, S. (1904). *Textura del Sistema Nervioso del Hombre y de los Vertebrados*, volume 2. Madrid Nicolas Moya.
- [Rechenberg, 1973] Rechenberg, I. (1973). *Evolutionsstrategie Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*.
- [Rumelhart et al., 1985] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1985). Learning internal representations by error propagation. Technical report, DTIC Document.
- [Russell et al., 2003] Russell, S. J., Norvig, P., Canny, J. F., and Malik, Jitendra M, E. D. D. (2003). *Artificial intelligence: a modern approach*, volume 2. Prentice hall Upper Saddle River.

- [Satzoda et al., 2013] Satzoda, R., Martin, S., Van Ly, M., Gunaratne, P., and Trivedi, M. M. (2013). Towards automated drive analysis: A multimodal synergistic approach. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, pages 1912–1916. IEEE.
- [Sekizawa et al., 2007] Sekizawa, S., Inagaki, S., Suzuki, T., Hayakawa, S., Tsuchida, N., Tsuda, T., and Fujinami, H. (2007). Modeling and recognition of driving behavior based on stochastic switched ARX model. *IEEE Transactions on Intelligent Transportation Systems*, 8(4):593–606.
- [Shiose et al., 2001] Shiouse, T., Onitsuka, T., and Taura, T. (2001). Effective information provision for relieving traffic congestion. In *Proceedings Fourth International Conference on Computational Intelligence and Multimedia Applications. ICCIMA 2001*, pages 138–142. IEEE.
- [Siddique and Adeli, 2013] Siddique, N. and Adeli, H. (2013). *Computational intelligence: synergies of fuzzy logic, neural networks y evolutionary computing*. John Wiley & Sons.
- [Suzumura and Kanezashi, 2012] Suzumura, T. and Kanezashi, H. (2012). Highly scalable x10-based agent simulation platform and its application to large-scale traffic simulation. *Proceedings of the 2012 IEEE/ACM 16th*.
- [Terada et al., 2010] Terada, R., Okuda, H., Suzuki, T., Isaji, K., and Tsuru, N. (2010). Multi-scale driving behavior modeling using hierarchical pwarx model. In *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pages 1638–1644. IEEE.
- [Terroso-Sáenz et al., 2015] Terroso-Sáenz, F., Valdés-Vela, M., Campuzano, F., Botia, J. A., and Skarmeta-Gómez, A. F. (2015). A complex event processing approach to perceive the vehicular context. *Information Fusion*, 21:187–209.
- [Toledo, 2007] Toledo, T. (2007). Driving Behaviour: Models and Challenges. *Transport Reviews*, 27(1):65–84.
- [Toledo et al., 2003] Toledo, T., Koutsopoulos, H. N., and Ben-Akiva, M. (2003). Modeling Integrated Lane-Changing Behavior. *Transportation Research Record*, 1857(1):30–38.
- [Toledo et al., 2007] Toledo, T., Koutsopoulos, H. N., and Ben-Akiva, M. (2007). Integrated driving behavior modeling. *Transportation Research Part C: Emerging Technologies*, 15(2):96–112.
- [Tordeux et al., 2011] Tordeux, A., Lassarre, S., and Roussignol, M. (2011). A study of the emergence of kinematic waves in targeted state car-following models of traffic. <http://cybergeog.revues.org>.
- [Trask ANDREWTRASK et al.,] Trask ANDREWTRASK, A., Gilmore DAVIDGILMORE, D., and Russell MATTHEWRUSSELL, M. Modeling Order in Neural Word Embeddings at Scale.

- [Treiber et al., 2000] Treiber, M., Hennecke, A., and Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805.
- [Turing, 1950] Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, 59(236):433–460.
- [Van Hoorn et al., 2009] Van Hoorn, N., Togelius, J., Wierstra, D., and Schmidhuber, J. (2009). Robust player imitation using multi-objective evolution. In *2009 IEEE Congress on Evolutionary Computation*, pages 652–659. IEEE.
- [Van Ly et al., 2013] Van Ly, M., Martin, S., and Trivedi, M. M. (2013). Driver classification and driving style recognition using inertial sensors. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 1040–1045. IEEE.
- [Wegener et al., 2008] Wegener, A., Piórkowski, M., Raya, M., Hellbrück, H., Fischer, S., and Hubaux, J.-P. (2008). TraCI. *Proceedings of the 11th communications and networking simulation symposium on - CNS '08*, (August 2016):155.
- [Wiedemann, 1974] Wiedemann, R. (1974). Simulation des strassenverkehrsflusses.
- [Wooldridge et al., 1995] Wooldridge, M., Jennings, N. R., Adorni, G., Poggi, A., Allen, J. F., Bates, J., Bell, J., BELNAP, N., PERLOFF, M., Bratman, M. E., Israel, D. J., Pollack, M. E., Brooks, R., Brooks, R. A., Bussmann, S., Demazeau, Y., Castelfranchi, C., Chaib-Draa, B., Moulin, B., Mandiau, R., Millot, P., Chang, E., Chapman, D., Chellas, B. F., Cohen, P. R., Levesque, H. J., Cohen, P. R., Perrault, C. R., Cutkosky, M., Engelmores, R., Fikes, R., Genesereth, M., Gruber, T., Mark, W., Tenenbaum, J., Weber, J., Downs, J., Reichgelt, H., Emerson, E. A., Halpern, J. Y., Fagin, R., Halpern, J. Y., Vardi, M. Y., Fisher, M., Gasser, L., Gasser, L., Braganza, C., Herman, N., Genesereth, M. R., Ketchpel, S. P., Georgeff, M. P., Greif, I., Guha, R. V., Lenat, D. B., Haas, A. R., Halpern, J. Y., Halpern, J. Y., Moses, Y., Halpern, J. Y., Vardi, M. Y., Hayes-Roth, B., Hewitt, C., Huang, J., Jennings, N. R., Fox, J., Jennings, N. R., JENNINGS, N. R., Jennings, N., Varga, L., Aarnts, R., Fuchs, J., Skarek, P., Kaelbling, L. P., Kraus, S., Lehmann, D., Kripke, S. A., Maes, P., Maes, P., McCabe, F. G., Clark, K. L., Mukhopadhyay, U., Stephens, L. M., Huhns, M. N., Bonnell, R. D., Müller, J. P., Pischel, M., Thiel, M., Newell, A., Simon, H. A., Norman, T. J., Long, D., PAPAOGLOU, M. P., LAUFMANN, S. C., SELLIS, T. K., Perlis, D., Perlis, D., Perloff, M., Poggi, A., Reichgelt, H., Sacerdoti, E. D., Searle, J. R., Shoham, Y., Thomas, B., Shoham, Y., Schwartz, A., Kraus, S., Thomason, R. H., Varga, L., Jennings, N. R., Cockburn, D., Vere, S., Bickmore, T., Wavish, P., Graham, M., Weerasooriya, D., Rao, A., Ramamohanarao, K., and Wooldridge, M. (1995). Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10(02):115.

[y Cajal, 1888] y Cajal, S. R. (1888). *Estructura de los centros nerviosos de las aves*.

[Yang and Koutsopoulos, 1996] Yang, Q. and Koutsopoulos, H. N. (1996). A microscopic traffic simulator for evaluation of dynamic traffic management systems. *Transportation Research Part C: Emerging Technologies*, 4(3 PART C):113–129.

[Yang, 2010] Yang, X. S. (2010). A new metaheuristic Bat-inspired Algorithm. In *Studies in Computational Intelligence*, volume 284, pages 65–74. Springer Berlin Heidelberg.

Índice alfabético

neodarwinismo, 39

car-following, 70, 71

fitness, 42, 44

free-flow, 70, 71

gap-acceptance, 72

lane-change, 72

lane-selection, 72

neodarwinismo, 39

síntesis evolutiva moderna, 39

selección por torneo, 43

Acerca del código fuente

La presente tesis lleva consigo numerosas horas de programación y, por tanto, muchísimas líneas de código. Éste se encuentra en formato electrónico como datos adjuntos a la memoria y no como capítulo o anexo a ésta, una forma más manejable para su consulta y a la vez respetuosa con el medio ambiente. No obstante sí es posible que existan pequeños fragmentos de código para apoyar explicaciones. En caso de necesitar los fuentes y no estar disponibles los datos anexos a la memoria, se puede contactar directamente conmigo, el autor, en .

Cómo citar esta tesis

Si deseas citar esta tesis, lo primero gracias. Me alegro de que te sirva para tu investigación. Si lo deseas, incluye el siguiente código en bibtex:

TODO A ver cómo coño meto en el paquete listings caracteres acentuados...