

Competición de Vídeos Divulgativos

Presidentes del Comité de Competición:

JOSÉ ANTONIO GÁMEZ

ALBERTO BUGARÍN

Jurado:

SENÉN BARRO AMENEIRO (PRESIDENTE),

JUAN MANUEL CORCHADO, ÓSCAR CORDÓN,

MARÍA JOSÉ DEL JESUS, JOSÉ A. GÁMEZ





Competición Vídeo: ¿Cómo puedo saber si el resultado de un clustering es lo suficientemente bueno?

José María Luna-Romera and José C. Riquelme Santos
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Sevilla
Sevilla, España

Abstract—En este vídeo divulgativo se hace una introducción a una de las técnicas de análisis de datos más usada, el clustering. En él, se explica a través de sencillos ejemplos qué es el clustering y cómo se analizan las distintas soluciones que ofrece a través de diferentes índices de validación.

Index Terms—Análisis de clustering, validación de clustering

I. HERRAMIENTAS Y MATERIALES

Para la elaboración del vídeo se han usado dos herramientas principalmente: Microsoft Power Point, para la elaboración de las diferentes pantallas incluyendo las animaciones de las figuras; Pinnacle Studio [1], que es un programa de edición de vídeo que nos ha permitido añadir las pistas de audio, así como controlar los tiempos de una manera más cómoda a Power Point.

II. LINK DEL VÍDEO

<https://youtu.be/ERDjUM78H1Q>

III. JUSTIFICACIÓN

El tema principal del vídeo es la validación de clustering. Se ha escogido esta técnica con el fin de acercar a una audiencia no especializada el clustering, una de las técnicas de análisis de datos más usada y nuclear en el aprendizaje no supervisado. Además, con el objetivo de profundizar en el tema, se da una visión de los índices de validación de clustering para motivar al espectador a plantearse algunas preguntas y cómo llegar a responderlas. A pesar del amplio uso del clustering, sus técnicas de validación son bastante desconocidas, si las comparamos, por ejemplo, a la amplia literatura sobre cómo acreditar la bondad de un clasificador.

La validación del clustering es una de las cuestiones más importantes en el análisis clustering. De hecho, se podría afirmar que medir la calidad de una solución de clustering es tan importante como el método de clustering en sí [2]. Para ello, en la literatura se definen índices de validación de clustering (CVI) que miden la calidad de la solución. Estas medidas podrían clasificarse en CVI internos y externos [3], [4]. Los CVI internos se basan en cómo se distribuyen los objetos en los distintos clusters utilizando sólo la distancia entre los objetos. Por otro lado, los índices externos se apoyan en posibles etiquetas de los objetos que puedan proporcionar

algún tipo de información externa para medir la calidad del clustering [5].

IV. RECURSOS

En primera instancia se quiso realizar el vídeo usando algún tipo de herramienta que facilitara la elaboración de material didáctico por medio de animaciones. Existen en el mercado servicios online como VideoScribe [6], Doodly [7] o Animaker [8], que a través de una sencilla e intuitiva interfaz permiten realizar vídeos con animaciones, figuras, iconos e incluso personajes. Todos estos servicios disponen de un periodo de prueba en el cual puedes probar la herramienta e incluso algunas de ellas permiten la exportación del vídeo a baja resolución. El inconveniente que encontramos al usar estas herramientas es que al tratarse de versiones de prueba, solo algunas permitían exportar el vídeo y lo hacían dejando una marca de agua.

Como se ha comentado en el apartado anterior, finalmente se ha usado Microsoft Power Point para elaboración de las diferentes pantallas que van apareciendo en el vídeo y Pinnacle Studio para añadir las pistas de audio y realizar el control de tiempos de cada pantalla.

V. UTILIDAD

Se ha elaborado un vídeo que permite a una audiencia no especializada introducirse en una de las técnicas de aprendizaje no supervisado más usadas de una manera fácil de entender. Además, se ha querido dar un paso más, presentando un nuevo concepto como puede ser la validación del clustering, y qué tipos de técnicas existen. Todo ello se muestra a través de sencillos ejemplos y se ilustra de manera gráfica qué es el clustering y cómo se puede validar la solución.

El vídeo ha sido visualizado por personas con desconocimiento total de la materia, y aunque ha habido diferentes valoraciones, por lo general han sido positivas, destacando la facilidad del lenguaje empleado y el acercamiento de algo complejo a través de ejemplos gráficos.

VI. CONCLUSIONES

Se ha elaborado un vídeo divulgativo de corta duración para acercar a una audiencia general, una técnica fundamental en el análisis de datos. El vídeo incluye una explicación



general del clustering además de dos tipos de técnicas de validación. A pesar de ser el clustering una técnica muy usada en múltiples ámbitos, no es muy abundante los trabajos divulgativos sobre su validación, a diferencia de las técnicas de aprendizaje supervisado, con mucha mayor presencia en ámbitos científicos.

REFERENCES

- [1] P. Studio, “Pinnacle Studio: software de edición de vídeo y grabación de pantalla,” <https://www.pinnaclesys.com/es/products/studio/>, 2018, [Online; accessed september-2018].
- [2] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, jun 2010. [Online]. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0167865509002323>
- [3] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona, “An extensive comparative study of cluster validity indices,” *Pattern Recognition*, vol. 46, no. 1, pp. 243 – 256, 2013. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S003132031200338X>
- [4] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, “On clustering validation techniques,” *Journal of Intelligent Information Systems*, vol. 17, no. 2, pp. 107–145, Dec 2001. [Online]. Available: <https://doi.org/10.1023/A:1012801612483>
- [5] Y. Lei, J. C. Bezdek, S. Romano, N. X. Vinh, J. Chan, and J. Bailey, “Ground truth bias in external cluster validity indices,” *Pattern Recognition*, vol. 65, pp. 58 – 70, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0031320316303910>
- [6] VideoScribe, “VideoScribe - Whiteboard Animation Software | Whiteboard Video Maker,” <https://www.videoscribe.co/>, 2018, [Online; accessed september-2018].
- [7] Doodly, “Doodly - Easily Create Whiteboard Doodle Videos In Minutes!” <https://www.doodly.com/>, 2018, [Online; accessed september-2018].
- [8] Animaker, “Animaker, Crea video-animaciones en la nube gratis,” <https://www.animaker.es>, 2018, [Online; accessed september-2018].



Competición Vídeo: Inteligencia artificial aplicada al desarrollo de software

Aurora Ramírez Quesada

Dpto. Informática y Análisis Numérico, Universidad de Córdoba

aramirez@uco.es

Resumen—Los profesionales de la medicina, las finanzas o la logística han sabido incorporar los avances que ofrece la inteligencia artificial, convirtiéndola en una herramienta de apoyo para la toma de decisiones o la optimización de sus procesos. Normalmente, la creación de estos procesos inteligentes recae en ingenieros informáticos con conocimientos en estas técnicas. Sin embargo, los propios profesionales de la informática también se enfrentan a situaciones en las que la inteligencia artificial puede servir de apoyo para, por ejemplo, acelerar el tiempo de ejecución de los proyectos o mejorar los productos software. El propósito de este vídeo es presentar de manera divulgativa una de las técnicas de inteligencia artificial más conocidas, los algoritmos evolutivos, y cómo estos pueden aplicarse en el ámbito del desarrollo software.

I. INTRODUCCIÓN

La aplicación de técnicas de inteligencia artificial (IA) al proceso de desarrollo software no se limita a un único tipo de técnica. Si bien se utilizan ya métodos propios del aprendizaje automático o basados en redes bayesianas, la aplicación de técnicas de optimización y búsqueda han despertado un especial interés en los últimos años. La denominada ingeniería del software basada en búsqueda (*Search-based Software Engineering*, SBSE) [1] propone la reformulación de las tareas propias de la ingeniería del software como problemas de optimización, para a continuación abordar su resolución mediante técnicas como las metaheurísticas [2].

Actualmente, es posible encontrar propuestas SBSE para abordar tareas en la mayoría de fases del ciclo de vida del software:

1. Requisitos, con estudios centrados en la selección y priorización de los requisitos a implementar en la siguiente versión del software [3].
2. Análisis y diseño, tanto de sistemas orientados a objetos como basados en servicios [4].
3. Codificación, donde se persigue corregir errores y mejorar aspectos no funcionales del código fuente [5].
4. Pruebas, donde el objetivo es principalmente la generación automática de casos de prueba [6].
5. Mantenimiento, que engloba procesos de modularización y refactorización del código [7].
6. Gestión del proyecto, donde destacan problemas de planificación temporal y de predicción de costes [8].

Desde el punto de vista de las técnicas empleadas, los algoritmos evolutivos son posiblemente los más populares en SBSE [9]. Se trata de técnicas metaheurísticas inspiradas en la teoría de la evolución de Darwin [10]. Un algoritmo evolutivo

realiza un proceso de optimización iterativo, el cual parte de una población de individuos generados de forma aleatoria. Cada individuo representa una solución candidata al problema, cuya calidad es medida por la función de *fitness*. En cada iteración, conocida como generación, algunos individuos son seleccionados para producir nuevas soluciones al problema. En concreto, el proceso más común es aplicar un operador de cruce, que combina las características de pares de soluciones buscando converger hacia la solución óptima. A continuación, se ejecuta un operador de mutación que altera pequeñas partes de una solución a fin de introducir diversidad en el proceso.

A la hora de aplicar un algoritmo evolutivo en el contexto de SBSE, es necesaria la definición de los dos elementos específicos del problema: la representación de las soluciones y la función de fitness [1]. En el primer caso, se suelen emplear las estructuras en forma de vector habituales en los algoritmos evolutivos, aunque también se pueden utilizar codificaciones en forma de árbol, adecuados para representar partes de un código fuente, o estructuras compuestas para problemas que necesitan manejar más información. Respecto a la función de *fitness*, las propuestas SBSE suelen utilizar medidas software para medir la calidad de los artefactos software producidos. Por ejemplo, la generación de casos de prueba se basa en criterios de cobertura del código, mientras que medidas de complejidad, cohesión o acoplamiento son frecuentes en las áreas de diseño y mantenimiento.

Se ha elegido esta temática porque permite mostrar a los profesionales de la informática que pueden beneficiarse de las soluciones que aporta la IA, no solo desarrollarlas. En cuanto a la técnica elegida, los algoritmos evolutivos son una de las técnicas pioneras en el ámbito de las metaheurísticas basadas en población, que a menudo resultan muy llamativas y son relativamente fáciles de entender por su símil biológico. Además, como se mencionó anteriormente, son muy populares en SBSE, por lo que es posible encontrar más ejemplos de aplicación.

II. DESCRIPCIÓN DEL VÍDEO

El vídeo tiene como principal propósito presentar una visión global de las posibilidades que ofrece la IA a los profesionales software como método de apoyo a la decisión. Es por ello por lo que el vídeo comienza planteando diversas situaciones en las que los ingenieros deberían valorar distintas alternativas antes de elegir la mejor solución. De esta forma se busca hacer ver al espectador la dificultad que conlleva el desarrollo de

software y la variedad de casuísticas para las que la IA puede ser útil. Para mantener una separación clara entre el dominio de aplicación y la técnica de IA empleada, se utilizan colores de fondo diferentes como recurso visual.

Una vez realizada la introducción a la temática, el vídeo entra en una fase más teórica que tiene dos objetivos principales: (1) centrar al espectador en las técnicas de búsqueda, como una de las ramas de la IA; y (2) explicar los fundamentos básicos de los algoritmos evolutivos. En este punto, la mayor parte del vídeo se dedica a explicar el proceso iterativo de búsqueda que efectúa un algoritmo evolutivo. Para ello se introduce un pequeño robot que sirve para representar a la IA como una entidad “tangibile” encargada de realizar los distintos pasos del ciclo evolutivo: generar soluciones, seleccionar las soluciones “padre” mediante torneo, generar soluciones “hijo” aplicando cruce y mutación, y reemplazar parte de la población para la siguiente generación. Dado el carácter teórico de esta parte del vídeo, se ha optado por utilizar diseños basados en pizarras y esquemas, así como fuentes que simulan la escritura manual.

Tras la explicación del proceso evolutivo de forma genérica, el vídeo aborda su adaptación a problemas concretos del ámbito del desarrollo software. Para ello se realiza un recorrido por el ciclo de vida clásico del software, planteando un problema para alguna de sus fases. En concreto, el vídeo retoma las preguntas que se planteaban los personajes al comienzo del mismo, transformándolas en la definición de los siguientes problemas de búsqueda:

- Planificación de proyectos software, cuyo objetivo es asignar tareas al equipo de desarrollo para minimizar el tiempo y coste del proyecto [11]. Este problema permite mostrar un ejemplo de codificación real.
- Diseño orientado a objetos, donde se pueden aplicar algoritmos evolutivos para encontrar la mejor asignación de propiedades y métodos a clases en base a criterios de cohesión y acoplamiento [12]. En este caso, se opta por una representación con vectores enteros.
- Selección de casos de prueba, que consiste en elegir un subconjunto de casos de prueba que alcancen una cobertura del código determinada [13]. Este problema se puede abordar como un problema de codificación binaria.
- Refactorización de código, donde se persigue encontrar una secuencia de operaciones de refactorización que eliminen el mayor número de defectos de diseño posibles [14]. Este ejemplo permite introducir una codificación basada en listas de distinta longitud.

Además de describir brevemente el objetivo que se persigue con la optimización y cómo se puede representar el problema, el vídeo aborda la definición de la función de *fitness* y un ejemplo de posibles operadores de cruce y mutación. De esta forma se consigue mostrar la flexibilidad de los algoritmos evolutivos y la variedad de problemas a los que se puede aplicar.

Finalmente, también se hace una mención a enfoques evolutivos más avanzados que tienen gran aplicabilidad en SBSE, como son los algoritmos multiobjetivo [15], que pueden

optimizar más de una función objetivo, y los modelos interactivos [16], que permiten incorporar la opinión del experto.

III. HERRAMIENTAS Y MATERIALES

Para la elaboración del vídeo se han utilizado herramientas software gratuitas y, en su mayoría, de código abierto. En primer lugar, la edición del vídeo se ha realizado con Powtoon¹. Se trata de una aplicación web que permite la creación de vídeos con dibujos animados. Aunque se trata de una aplicación de pago, la versión gratuita ofrece la funcionalidad necesaria para la edición de vídeos de corta duración. Además de los recursos gráficos y sonoros proporcionados por la propia herramienta, el usuario puede incorporar sus propias imágenes, vídeos y audios. Powtoon pone a disposición del usuario una gran variedad de plantillas y ejemplos, que facilitan la creación de escenas más complejas. A su vez, los elementos gráficos (objetos, personajes, rótulos, etc.) están disponibles en diferentes estilos, lo que permite adecuar el contenido a diferentes propósitos y audiencias.

Para complementar el catálogo de imágenes de Powtoon, se han utilizado imágenes con licencia *creative common* disponibles en la plataforma Pixabay². En ella pueden descargarse multitud de imágenes en formato PNG o vectorial de forma gratuita. Cuando ha sido necesario, las imágenes han sido editadas mediante herramientas como Inkscape³ o Gimp⁴.

IV. UTILIDAD DIDÁCTICA DEL VÍDEO

El vídeo presentado a la competición puede ser un buen recurso didáctico por varios motivos. En primer lugar, proporciona una introducción ágil y práctica a las técnicas de búsqueda, que suelen formar parte de asignaturas sobre IA. Además, presenta los fundamentos básicos de la que posiblemente es la técnica más estudiada, los algoritmos evolutivos. Gracias a la presentación de varios ejemplos, el vídeo puede servir de complemento a la hora de mostrar ámbitos de aplicación menos conocidos. Igualmente, el vídeo también deja entrever la necesidad de desarrollar modelos más avanzados con los que abordar problemas complejos del mundo real, lo cual puede despertar el interés por estudiarlas en más detalle.

Cabe destacar también el marcado carácter interdisciplinar del vídeo, que lo hace muy interesante para los estudiantes de las distintas ramas de la informática. En este sentido, los estudiantes verán que aprender los fundamentos de la IA les puede ser de utilidad para distintos perfiles, tanto investigadores como profesionales. Esto es especialmente relevante dada la evolución que está sufriendo la informática y la necesidad que tienen las empresas de contar con perfiles interdisciplinares. Finalmente, el vídeo también puede ser utilizado de forma divulgativa fuera de las aulas, por ejemplo, para favorecer la transferencia de conocimiento con empresas.

¹<https://powtoon.com>

²<https://pixabay.com>

³<http://inkscape.com>

⁴<https://gimp.com>



V. CONCLUSIONES

El vídeo presentado a la competición CAEPIA'18 muestra de forma amena cómo la inteligencia artificial puede resolver problemáticas que surgen a menudo en el proceso de desarrollo software. La intención de este vídeo ha sido presentar a la IA como un método de apoyo a la decisión, capaz de abordar tareas tan diversas como encontrar la mejor planificación de equipos de desarrollo, seleccionar los requisitos que más beneficio reportarán a la empresa, optimizar el diseño de complejos sistemas informáticos o automatizar su entorno de pruebas. Debido a la limitación temporal, el vídeo se centra únicamente en la aplicación de algoritmos evolutivos, sobre los que se explica su proceso iterativo de búsqueda y cómo sus diferentes elementos se deben adaptar al problema a resolver. Para la realización del vídeo se ha utilizado Powtoon, una potente aplicación web para la creación de vídeos mediante dibujos animados.

A modo de valoración personal, la elaboración de este vídeo ha supuesto una gran oportunidad para dar a conocer un dominio de aplicación de la IA que no es tan popular en entornos académicos y, menos aún, entre los profesionales de la informática. También ha supuesto un ejercicio de reflexión sobre la dificultad a la hora de transmitir conceptos avanzados a una audiencia sin conocimientos profundos de IA y en un tiempo muy limitado. Esto ha requerido buscar el lenguaje apropiado, adaptar los contenidos sin perder rigor, recopilar ejemplos prácticos y ser capaz de explicarlos de forma sencilla, etc.

Finalmente, participar en esta competición también ha permitido a su autora aprender a utilizar herramientas software con gran potencial, como es Powtoon. Las habilidades adquiridas pueden ser de gran utilidad en el futuro para la elaboración de materiales didácticos y divulgativos relacionados con esta temática.

ENLACE AL VÍDEO

El vídeo se encuentra disponible en el siguiente enlace:
https://www.youtube.com/watch?v=vfpY8_AEB6A

AGRADECIMIENTOS

La autora desea expresar su agradecimiento a José Raúl Romero por sus ideas y comentarios sobre las primeras versiones del vídeo.

REFERENCIAS

- [1] M. Harman, S. A. Mansouri, and Y. Zhang, "Search Based Software Engineering: Trends, Techniques and Applications," *ACM Computing Surveys*, vol. 45, no. 1, pp. 11:1–61, 2012.
- [2] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Information Sciences*, vol. 237, pp. 82–117, 2013.
- [3] A. M. Pitangueira, R. S. P. Maciel, and M. Barros, "Software requirements selection and prioritization using SBSE approaches: A systematic review and mapping of the literature," *Journal of Systems and Software*, vol. 103, pp. 267–280, 2015.
- [4] O. Riih a, "A survey on search-based software design," *Computer Science Review*, vol. 4, no. 4, pp. 203–249, 2010.
- [5] J. Petke, S. O. Haraldsson, M. Harman, W. B. Langdon, D. R. White, and J. R. Woodward, "Genetic Improvement of Software: A Comprehensive Survey," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 3, pp. 415–432, 2018.
- [6] P. McMinn, "Search-based software test data generation: A survey," *Software Testing Verification and Reliability*, vol. 14, no. 2, pp. 105–156, 2004.
- [7] T. Mariani and S. R. Vergilio, "A systematic review on search-based refactoring," *Information and Software Technology*, vol. 83, pp. 14–34, 2017.
- [8] F. Ferruci, M. Harman, and F. Sarro, "Search-Based Software Project Management," in *Software Project Management in a Changing World*, pp. 373–399, Springer-Verlag Berlin Heidelberg, 2014.
- [9] M. Harman, "Software Engineering Meets Evolutionary Computation," *IEEE Software*, no. October, pp. 31–39, 2011.
- [10] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. Springer-Verlag Berlin Heidelberg, 2nd ed., 2015.
- [11] E. Alba and J. F. Chicano, "Software project management with GAs," *Information Sciences*, vol. 177, no. 11, pp. 2380–2401, 2007.
- [12] M. Bowman, L. C. Briand, and Y. Labiche, "Solving the class responsibility assignment problem in object-oriented analysis with multi-objective genetic algorithms," *IEEE Transactions on Software Engineering*, vol. 36, no. 6, pp. 817–837, 2010.
- [13] L. S. de Souza, R. B. Prud ncio, F. d. A. Barros, and E. H. d. S. Aranha, "Search based constrained test case selection using execution effort," *Expert Systems with Applications*, vol. 40, no. 12, pp. 4887–4896, 2013.
- [14] A. Ouni, M. Kessentini, H. Sahraoui, and M. Boukadoum, "Maintainability defects detection and correction: A multi-objective approach," *Automated Software Engineering*, vol. 20, no. 1, pp. 47–79, 2013.
- [15] A. S. Sayyad and H. Ammar, "Pareto-optimal search-based software engineering (POSBSE): A literature survey," in *Proceedings of the 2nd International Workshop on Realizing Artificial Intelligence Synergies in Software Engineering (RAISE)*, pp. 21–27, 2013.
- [16] A. Ram rez, J. R. Romero, and C. Simons, "A Systematic Review of Interaction in Search-Based Software Engineering," *IEEE Transactions on Software Engineering*, 2018. In press.

Video Competition: Solving Smart Mobility Problems Using Bio-inspired Techniques

1st Daniel H. Stolfi

Departamento de Lenguajes y Ciencias de la Computación
University of Malaga
Malaga, Spain
dhstolfi@lcc.uma.es

2nd Enrique Alba

Departamento de Lenguajes y Ciencias de la Computación
University of Malaga
Malaga, Spain
eat@lcc.uma.es

I. TOOLS AND ASSETS

This video was made using Open Source tools such as Ubuntu, Kdenlive, Audacity, Gimp, LibreOffice, and the traffic microsimulator SUMO. Graphic resources were obtained from OpenStreetMap, Google Earth, Bing Maps, and Pixabay. Finally, sound effects were downloaded from Free Sound Clips. The selected soundtrack is Chill Police by Anttis Instrumentals. The rest of pictures and sounds were created by the authors. Link: <https://youtu.be/ssE0RDitQvY>

II. A.I. TECHNIQUE

We have chosen our epiGenetic Algorithm (epiGA) [5] as an A.I. technique to optimize our Yellow Swarm architecture [3], [4], [6] to obtain the optimal time slots for each sign and LED panel with the aim of reducing travel times, greenhouse gas emissions, and fuel consumption in the city of Malaga (Spain).

Yellow Swarm suggests possible detours to vehicles by showing different cyclical indications to drivers using LED panels. It has two stages: The Offline stage in which the system is configured, and the Online stage in which drivers are informed about the suggested detours. In the Offline stage, our epiGA analyzes different scenarios (traffic distributions) of our case studies, using the SUMO traffic simulator [1] which is controlled by the TraCI module [7], in order to implement the decisions that drivers make during their journey. The urban maps used to build the case studies have been imported from OpenStreetMap [2] so that we can test our system in realistic city districts, including traffic lights, roundabouts, etc. The training carried out in this phase results in the configuration of the LED panels to be used in the next stage.

In the Online stage, the LED panels show the different detour options to drivers depending on the time slots calculated in the previous stage, using the Panel Manager. The possible signs are: go straight on, turn left, and turn right. However, their availability depends on the type of junction the vehicles are approaching, i.e. the possible detour options, and the street where the panel is placed. The first sign is visible during a previously calculated time interval, after that the next sign in the sequence will be presented to the drivers. Once the cycle has finished it again starts with the first sign of the sequence.

By using this strategy, Yellow Swarm is able to prevent possible traffic jams in the city as well as improve the use

of most of the available secondary streets. Furthermore, as it uses LED panels, no personal devices (GPS navigators, mobile phones, etc.) are needed. Yellow Swarm also promotes road safety as drivers do not have to lose eye contact with the street while driving.

The epiGenetic Algorithm (epiGA) [5] consists of a set of strategies, based on evolutionary computation, inspired in nature, especially in epigenetics, with the aim of solving complex combinatorial problems. DNA and histones are collapsed to form nucleosomes, and this affects the gene replication during reproduction, while the epigenetic mechanisms modify the gene expression through methylation. We use this bio-inspired model to build the operators of our algorithm.

III. NARRATIVE RESOURCES

The video was intended to be understood by everyone, so general concepts were presented and at the end of it, a set of references are given in order to provide more technical information. The first part is meant to present the road traffic in the city and its negative consequences as a real world problem to be solved, then our smart mobility solution (Yellow Swarm) is explained. After that, the A.I. technique (epiGenetic Algorithm) takes over to answer one research question: How the Yellow Swarms time slots can be optimally calculated? Finally, the improvements achieved, i.e. shorter travel times, less emissions and fuel consumption, are shown in a bar graph.

IV. DISCUSSION

We have used several sound effects, animations, and transitions to make the video more dynamic and keep the viewers engaged. Our goal here is to present our smart mobility solution in which an A.I. technique is used to make it effective and efficient. Viewers should be able to understand our proposal and learn how new algorithms inspired in nature can be used to improve their lives. Additionally, it may be useful to show that Artificial Intelligence is not *Skynet*, i.e. that it is not here to end with humans, but to help us.

ACKNOWLEDGMENT

This research has been partially funded by the Spanish MINECO and FEDER projects TIN2014-57341-R, TIN2016-81766-REDT, and TIN2017-88213-R. University of Malaga,



Andalucía TECH. Daniel H. Stolfi is supported by a FPU grant (FPU13/00954) from the Spanish MECD.

REFERENCES

- [1] Krajzewicz, D., Erdmann, J., Behrisch, M., Bieker, L.: Recent Development and Applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements* **5**(3), 128–138 (2012)
- [2] OpenStreetMap Foundation: OpenStreetMap (2018), <http://www.openstreetmap.org/>
- [3] Stolfi, D.H., Alba, E.: Smart Mobility Policies with Evolutionary Algorithms: The Adapting Info Panel Case. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation. pp. 1287–1294. GECCO '15, ACM, New York, NY, USA (2015)
- [4] Stolfi, D.H., Alba, E.: Un Algoritmo Evolutivo para la Reducción de Tiempos de Viaje y Emisiones Utilizando Paneles LED. In: X Congreso Español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados. pp. 27–34. MAEB 2015, Mérida - Almendralejo (2015)
- [5] Stolfi, D.H., Alba, E.: Epigenetic algorithms: A New way of building GAS based on epigenetics. *Information Sciences* **424**(Supplement C), 250–272 (2018)
- [6] Stolfi, D.H., Armas, R., Alba, E., Aguirre, H., Tanaka, K.: Fine Tuning of Traffic in Our Cities with Smart Panels: The Quito City Case Study. In: Proceedings of the Genetic and Evolutionary Computation Conference 2016. pp. 1013–1019. GECCO '16, ACM, New York, NY, USA (2016)
- [7] Wegener, A., Piórkowski, M.: TraCI: An Interface for Coupling Road Traffic and Network Simulators. In: Proceedings of the 11th Communications and Networking Simulation Symposium. pp. 155–163. CNS '08, ACM, New York, NY, USA (2008)