



COLDPower: **El agente inteligente** **mexicano de** **mercadeo de energía**



Estimación de precios de vivienda mediante IA

El uso malicioso de la IA en Twitter

Interfaz gráfica del algoritmo basado en el forrajeo de bacterias

Conceptos básicos de RGN y PRGN

Módulo de rutas para la entrega de comida a través de un algoritmo PSO

Control de interferencia en una red de comunicaciones móviles





©Komputer Sapiens, Año XII Volumen II, mayo-agosto 2020, es una publicación cuatrimestral de la Sociedad Mexicana de Inteligencia Artificial, A.C., con domicilio en Ezequiel Montes 56 s/n, Fracc. los Pilares, Metepec, Edo. de México, C.P. 52159, México, <http://www.komputersapiens.org>, correo electrónico: editorial@komputersapiens.org, tel. +52 (833)357.48.20 ext. 3024, fax +52 (833) 215.85.44. Impresa por Sistemas y Diseños de México S.A.

de C.V., calle Aragón No. 190, colonia Álamos, delegación Benito Juárez, México D.F., C.P. 03400, México, se terminó de imprimir el 31 de agosto de 2020, este número consta de 1000 ejemplares.

Reserva de derechos al uso exclusivo número 04-2009-111110040200-102 otorgado por el Instituto Nacional de Derechos de Autor. ISSN 2007-0691.

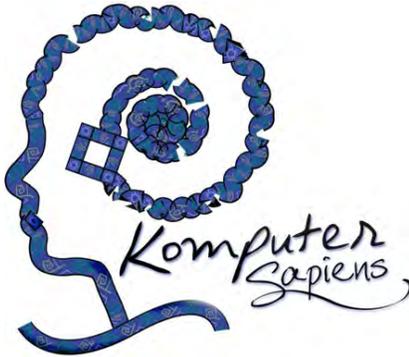
Los artículos y columnas firmados son responsabilidad exclusiva de los autores y no reflejan necesariamente los puntos de vista de la Sociedad Mexicana de Inteligencia Artificial. La mención de empresas o productos específicos en las páginas de Komputer Sapiens no implica su respaldo por la Sociedad Mexicana de Inteligencia Artificial.

Queda estrictamente prohibida la reproducción total o parcial por cualquier medio, de la información aquí contenida sin autorización por escrito de los editores.

Komputer Sapiens es una revista de divulgación en idioma español de temas relacionados con la inteligencia artificial. Creada en \LaTeX , con la clase **papertex** disponible en el repositorio *CTAN*: Comprehensive TeX Archive Network, <http://www.ctan.org/>

Indizada en el IRMDCT de CONACYT y en Latindex.

Presidente	Directorio SMIA Félix A. Castro Espinoza	Directores Fundadores Carlos Alberto Reyes García Ángel Kuri Morales	Jorge Rodas Osollo Grigori Sidorov
Vicepresidente	Ildar Batyrshin		Víctor Germán Sánchez Arias
Secretario	Oscar Herrera Alcántara		Aurora Torres
Tesorero	María de Lourdes G. Martínez Villaseñor		Gregorio Toscano Pulido
Vocales:	Sofía Natalia Galicia Haro Sabino Miranda Jiménez Roberto A. Vázquez Espinoza de los Monteros Hiram Erendin Ponce Espinosa Francisco Viveros-Jiménez Noé Alejandro Castro Sánchez Obdulia Pichardo Lagunas Omar Montaña Rivas Antonio Marín Hernández Néstor Velasco Bermeo Iris Iddaly Méndez Gurrola Gustavo Arroyo Figueroa Leobardo Morales Tiburcio Iván Olvera Rodríguez Juan Martínez Miranda	Comité Editorial Félix A. Castro Espinoza Jesús Favela Vara Sofía Natalia Galicia Haro Miguel González Mendoza Oscar Herrera Alcántara Raúl Monroy Borja Eduardo F. Morales Manzanares Leonardo Garrido Luna Carlos Alberto Reyes García Angélica Muñoz Meléndez Antonio Sánchez Aguilar Luis Enrique Sucar Succar Ángel Kuri Morales José A. Martínez Flores Juan Manuel Ahuactzin Larios Manuel Montes y Gómez Ofelia Cervantes Villagómez Alexander Gelbukh Grigori Sidorov Laura Cruz Reyes Elisa Schaeffer Ramon Brena Pinero Juan Humberto Sossa Azuela	Tania Turrubiates López Esua Villatoro Tello Juan Carlos Osorio Gómez Julio Ponce
Director general	Komputer Sapiens Félix A. Castro Espinoza	Árbitros Federico Alonso Pecina Giner Alor Hernández Ángel G. Andrade Héctor Avilés Ramón F. Brena Antonio Camarena Ibarrola Luis Castro Jaime Cerda Jacobo Ofelia Cervantes César Cárdenas Hugo Jair Escalante Luis Felipe Fernández Karina Figueroa Claudia Gómez Yasmín Hernández José Alberto Hernández Óscar Herrera José Francisco Martínez Trinidad Efrén Mezura Montes Daniela Moctezuma Raúl Monroy C. Alberto Ochoa Zezatti Francisco Javier Ornelas Nora Reyes	
Editora en jefe	Karina Mariela Figueroa Mora		
Editor científico	Héctor Gabriel Acosta Mesa		
Editores asociados	Marco Antonio Aguirre Lam Laura Cruz Reyes		
Coordinadora de producción e-Tlakuiló	Viridiana Mena Gómez Marco Antonio Aguirre Lam		
Estado del Arte	Reyna Carolina Medina Ramírez Jorge Rafael Gutiérrez Pulido Claudia G. Gómez Santillán		
Sakbe	Laura Cruz Reyes Laura Gómez Cruz		
IA & Educación	María Yasmín Hernández Pérez María Lucía Barrón Estrada J. Julieta Noguez Monroy		
Deskubriendo Konocimiento	Alejandro Guerra Hernández Leonardo Garrido Luna		
Asistencia técnica	Alan G. Aguirre Lam		
Corrección de estilo	Gilberto Rivera Zárate Marcela Quiroz Castellanos Ruth Esmeralda Barreda Guajardo Denisse Alvarado Guillermo de Jesús Hoyos Rivera Susana Aurora Velasco Montiel Nicandro Cruz Ramírez Enrique Gómez Cruz Claudia Leticia Díaz González Leonardo Romero Muñoz Sonia Lilia Mestizo Gutiérrez Nelson Rangel Valdez Laura Gómez Cruz Irene Morales Pagaza, Mopi Diseño		
Edición de imagen			
Portada			



Contenido

ARTÍCULO ACEPTADO

COLDPower: el agente inteligente mexicano de mercadeo de energía

Ansel Y. Rodríguez González, Eduardo Morales y Luis Enrique Sucar

pág. 6 ⇒ Introducción de conceptos básicos y muestras de resultados alcanzados en el Torneo PowerTAC.

ARTÍCULO ACEPTADO

Estimación automática de precios de vivienda mediante inteligencia artificial

Roberto Quintero, Cynthia B. Perez, Laura P. López-Arredondo, Jesus R. Villavicencio, Cesar Puente y Juan Villegas

pág. 12 ⇒ Proceso para la estimación automática de precios de viviendas.

ARTÍCULO ACEPTADO

Interfaz gráfica del algoritmo basado en el forrajeo de bacterias para optimización numérica con restricciones

J. Adrian García-López, Betania Hernandez-Ocaña, Oscar Chávez-Bosquez y José Hernández-Torruco

pág. 19 ⇒ Diseño de una interfaz gráfica para el algoritmo TS-MBFOA.

ARTÍCULO ACEPTADO

El uso malicioso de la Inteligencia Artificial en Twitter

Israel Morales Trigueros y Carlos Adolfo Piña García

pág. 25 ⇒ El contenido que nos presenta Twitter representa un beneficio propagandista para aquellos que logran controlar el flujo de información.

ARTÍCULO ACEPTADO

Algoritmos inspirados en la naturaleza para control de interferencia en una red de comunicaciones móviles celular

Jesús Alberto Gómez Avilés, Anabel Martínez-Vargas y Ángel G. Andrade

pág. 30 ⇒ Algoritmos inspirados en la naturaleza para una red móvil celular.

ARTÍCULO ACEPTADO

Conceptos básicos de los generadores de números aleatorios (RGN) y pseudoaleatorios (PRGN)

Elena Fabiola Ruiz- Ledesma, Elizabeth Acosta-Gonzaga y Tonáhtiu Arturo Ramírez-Romero

pág. 35 ⇒ La generación de números aleatorios tiene un campo amplio de uso y la selección de un programa PRNGs depende de diversos factores.

ARTÍCULO ACEPTADO

Módulo de rutas para la entrega de comida a través de un algoritmo Optimizado por enjambre de partículas (PSO) y con tecnología de Google maps

Sergio Soto Rodríguez, Edmundo Bonilla Huerta, Alberto Portilla Flores, José Crispín Hernández Hernández y Oscar Atriano Ponce

pág. 40 ⇒ Módulo para la optimización de rutas de reparto.

Columnas

Sapiens Piensa.
Editorial [pág. 2](#)

e-Tlakuilo [pág. 3](#)

Estado del IArte [pág. 4](#)

Sakbe [pág. 5](#)

IA & Educación [pág. 47](#)

Deskubriendo
Konocimiento [pág. 49](#)

Sapiens Piensa

Karina Mariela Figueroa Mora

Esta edición de Komputer Sapiens mantuvimos la temática libre a diversos temas relacionados con la inteligencia artificial en el ámbito de los negocios, en algoritmos inspirados en la naturaleza e incluso en la influencia que se podría tener en las redes sociales.

El campo de la inteligencia artificial tiene un alto impacto en el ámbito de los negocios. Actualmente, existen diversos simuladores de mercados de toda índole, entre ellos existe uno para energía llamado *PowerTAC*. Cada año, desde el 2012 se lleva a cabo un torneo anual donde distintos agentes se validan y compiten entre sí. En 2016 se destacó el agente COLDPower que obtuvo el 2o. lugar haciendo uso de aprendizaje por refuerzo y múltiples agentes. En el artículo aceptado titulado “COLDPower: el agente inteligente mexicano de mercadeo de energía”, nos muestra la arquitectura general de los agentes usados y cómo éstos aprenden sobre las estrategias a seguir en mercados de energías reales.

Otro ejemplo del uso de la Inteligencia Artificial en los negocios se presenta en el artículo “Estimación automática de precios de vivienda mediante inteligencia artificial”, donde los autores presentan un análisis de los principales factores para la estimación de precios de viviendas en el mercado mexicano mediante redes neuronales artificiales. Este tipo de herramientas automáticas son una realidad en mercados internacionales como Estados Unidos, Filipinas, China, entre otros, donde las predicciones de precios juega un papel que impacta en la economía nacional.

Un área de la inteligencia artificial que trata de encontrar soluciones a problemas complejos basado en procesos naturales o evolutivos se conoce como *algoritmos bio-inspirados*. Dichos algoritmos basan su funcionamiento en comportamientos cooperativos de algunos organismos (i.e. las bacterias). Este tipo de investigación se muestra en el artículo aceptado “Interfaz gráfica del algoritmo basado en el forrajeo de bacterias para optimización numérica con restricciones”; este artículo presenta los procesos importantes de algoritmos bio-inspirados, y describe el desarrollo de una interfaz gráfica para hacer uso de ellos.

Otro ejemplo de un algoritmo bio-inspirado es el presentado en “Algoritmos inspirados en la naturaleza para control de interferencia en una red de comunicaciones

móviles celular”. Este artículo aborda un problema de impacto, particularmente nos detalla la complejidad en una red de comunicaciones y propone una manera de resolver el problema de interferencia por medio de algoritmos genéticos, algoritmos basados en movimientos coordinados de aves y algoritmos basados en enjambres (el tipo de búsqueda que realizan las abejas para recolectar el néctar).

En los últimos años el mercado de entrega de alimentos se vio obligado a un importante crecimiento, lo cual provocó un crecimiento en el desafío de encontrar las mejores rutas para cumplir el objetivo. En el artículo “Módulo de rutas para la entrega de comida a través de un algoritmo Optimizado por enjambre de partículas (PSO) y con tecnología de Google maps”, se propone un módulo para una aplicación móvil que intenta minimizar los costos de reparto y maximiza el número de entregas. Los autores describen la implementación de un algoritmo bio-inspirado en el comportamiento de enjambres, parvadas o cardumen.

Las redes sociales se han vuelto parte de nuestro día, y pueden generar tendencias diariamente como es el caso de Twitter. Debido a la importancia de marcar una tendencia, existen actores interesados en polarizar la opinión pública. En el artículo “El uso malicioso de la Inteligencia Artificial en Twitter”, los autores presentan un amplio análisis del caso de estudio para el hashtag #PrensaProstituida y donde se discute la influencia derivada agentes automáticos.

Finalmente, pero no menos importante, el artículo aceptado con el título de “Conceptos básicos de los generadores de números aleatorios (RGN) y pseudoaleatorios (PRGN)”, nos describe la importancia de los números aleatorios en nuestro día a día y nos muestra una comparación entre distintos algoritmos para la generación de dichos números.

Este número fue preparado con mucho cuidado, dedicación y esmero para nuestros amables lectores de Komputer Sapiens. Los invitamos a disfrutar de cada uno de los interesantes artículos seleccionados así como de nuestras acostumbradas columnas. Por supuesto, como siempre, extendemos la invitación a leer nuestras ediciones pasadas disponibles en nuestro sitio web:

<http://komputersapiens.smia.mx>*



Dra. Karina Mariela Figueroa Mora es Editora en Jefe de la revista de Komputer Sapiens desde enero del 2020, profesora e investigadora de la Facultad de Ciencias Físico-Matemáticas de la Universidad Michoacana de San Nicolás de Hidalgo. Sus áreas de interés son algoritmos en general, recuperación de información, bases de datos métricas, así como desarrollo de software para la enseñanza de las matemáticas.

e-Tlakuilo: Cartas de nuestros lectores

Marco A. Aguirre L.
etlakuilo@komputersapiens.org

En *Komputer Sapiens* nos hemos esforzado por estar “a sólo un click de distancia” a través de diferentes medios como Facebook, Twitter y correo electrónico. Les presentamos algunas de las preguntas que hemos recibido a través de estos medios.

**Laura G. López – Alumna de nivel Universidad.
(vía correo electrónico)**

La Inteligencia Artificial es ampliamente utilizada en la actualidad ¿Cuál podría ser su futuro y las consecuencias de su uso?

Estimada Laura, gracias por la pregunta. Las tecnologías de Inteligencia Artificial han sufrido un gran cambio en los últimos años, por lo cual es difícil tener una caracterización completa al respecto.

Desde los inicios de los modelos de Inteligencia Artificial (IA) se ha buscado que los computadores puedan realizar tareas con un nivel de complejidad similar al de los humanos, tratando de resolver problemas que las personas pueden realizar de manera natural. Hasta la fecha, es difícil decir que éstos puedan tener una inteligencia superior a la que posee un humano.

La manera en la que podemos decir que una máquina es inteligente es por medio del aprendizaje observado. Un ejemplo de las técnicas de aprendizaje más utilizadas es el Machine Learning, la cual es una de las ramas de la IA en donde las computadoras aprenden a hacer una tarea sin ser programadas para ello, es decir, pueden recibir un entrenamiento de un conjunto de ejemplos y de esta manera aprender a realizar alguna acción. Del Machine Learning surge el Deep Learning, que es una tecnología ampliamente utilizada en la actualidad, ésta se basa en redes neuronales que contienen varios niveles jerárquicos. Cada nivel procesa información que va

desde la información más simple hasta llegar a un grado mucho más complejo de aprendizaje. Actualmente esta tecnología ha sido un foco de atención por su utilidad al aplicarse a grandes volúmenes de datos para el descubrimiento y aplicación del conocimiento, así como para la predicción de datos en infinidad de aplicaciones.

Las técnicas de IA han permitido un gran avance en muchas áreas, sin embargo, no existe un organismo de control que regule el manejo de estas tecnologías, y pueden llegar a tener un mal uso (p. ej. delitos informáticos, discriminación, influencias y tendencias negativas, entre otros). Este campo requiere de un amplio estudio, ética y la gestión de las políticas públicas que permitan su uso apropiado.

El avance de estas tecnologías ha cambiado la manera en la que se realiza el trabajo en las empresas, esto puede afectar en la economía de muchos trabajadores debido a la automatización, sobre todo en trabajos de tipo operativo y también para aquellos trabajadores que no tienen conocimientos computacionales avanzados. Por otra parte, esto podría abrir oportunidades para aquellos que puedan adaptarse al cambio y aquellos que posean habilidades de programación.

La IA se encuentra prácticamente en todo nuestro entorno (dispositivos móviles y en muchos otros dispositivos electrónicos). Es probable que el futuro de la IA nos alcance a todos, independientemente de nuestras actividades, nivel social o económico. Tal avance tecnológico puede ayudarnos o perjudicarnos, es complejo poder dar una opinión certera al respecto. Las consecuencias de la transición entre la vida cotidiana y las tecnologías de la IA pueden depender de su importancia y utilidad en la sociedad, así como la manera en el que el sistema educativo afronte esta situación.



Estado del IArte

Jorge Rafael Gutiérrez Pulido (@jrgpulido) y Reyna Carolina Medina Ramírez
estadoiarte@komputersapiens.org

La Inteligencia Artificial nuestra

La inteligencia artificial (IA) que nos ocupa, no es la que nos presentan en las películas, sino aquella que se está creando en los centros de investigación y desarrollo. Son algoritmos especializados que permiten a un agente de hardware o software realizar una tarea. En esta edición queremos abordar el tema de la IA en nuestro país. Los problemas a los que se enfrenta la comunidad científica que está interesada en la IA son principalmente la falta de recursos, de incentivos, de coordinación, de evaluación, y de una visión a largo plazo. Con restricciones de espacio trataremos de hacerlo de manera sucinta desde las siguientes perspectivas.

Gente

Existe un número importante de investigadores con formación de posgrado en el área de la IA. Muchos de ellos se han formado en el extranjero. Existe también un número importante de emprendedores, egresados de universidades mexicanas con formación en áreas de la computación que incursionan en la IA. Todavía más, hay muchos mexicanos en el extranjero trabajando en diversas áreas de la IA. Para darle un mayor impulso a la IA en México la formación de la gente debe iniciar desde el nivel primaria, incorporando materias de computación, ciencia de datos, idioma inglés, y otras relacionadas. No abundaremos en las carencias del sistema educativo, pero es un aspecto importante que debe mejorar.

Instituciones

Hace un par de décadas un esfuerzo importante del gobierno hacia las instituciones educativas del nivel superior del país fue el denominado PROMEP (SEP-CONACyT). Su intención era impulsar a las universidades públicas en las diferentes áreas del conocimiento, incluyendo la IA, y habilitar a los profesores de dichas instituciones con grado preferente de doctorado para realizar docencia e investigación, y así en cierta medida evitar la fuga de cerebros. Un plus importante es que los profesores formados en el extranjero, no de habla hispana, escribían en otro idioma: tesis, reportes, artículos, avances de investigación, entre otros. En lo que quedó corto el esfuerzo fue en dar continuidad al proyecto. Se suponía que los profesores tendrían todas las condiciones que tuvieron durante sus estudios para regresar a su trabajo,

estrechar lazos de colaboración, y continuar haciendo investigación de frontera.

Otros

Vale la pena mencionar que existen otros esfuerzos que se realizan en el área de la IA. Primero, la Sociedad Mexicana de Inteligencia Artificial (SMIA), que por cierto publica esta revista, es un esfuerzo continuo de más de 12 años de existencia. Los congresos COMIA y MICAI son esfuerzos de alrededor de una década el primero y dos décadas el segundo. Atraen investigadores de todo el mundo y México para presentar avances de sus desarrollos en el área.

Un esfuerzo muy reciente es la coalición multisectorial denominada **IA2030Mx** cuya intención es conjuntar gente, instituciones, industria, gobierno, y otras entidades relacionadas e interesadas en el desarrollo de la IA en nuestro país a través de una consulta nacional. Conviene estar pendiente de los resultados que presenten. Esta y otras iniciativas merecen la atención de inversionistas. En todas, como hemos mencionado, existe el riesgo de la falta de seguimiento.

Propuesta

La IA definitivamente es un factor clave para impulsar y mejorar la competitividad de nuestro país en todos los sectores. Gobierno, academia, e industria, deben involucrarse activamente y de manera coordinada en toda iniciativa relacionada con la IA. El gobierno debe apoyar decididamente con recursos económicos *semilla* suficientes. Visualizarlos como una inversión a largo plazo, aplicando auditorías permanentes para evitar que se dispersen, y dar puntual seguimiento a las iniciativas para impulsar lo bien hecho. La academia debe coordinar el esfuerzo, impulsar la creación de comités de IA en las universidades, bachilleratos, secundarias, y primarias para promover su adopción. La industria debe formar parte del comité coordinador, participar en la creación de los ejes de desarrollo y guiar las estrategias para su consecución, y por supuesto también debe inyectar recursos económicos de manera decidida. Considerar también otros actores importantes para el desarrollo de la IA en México tales como pensadores de las áreas sociales y humanas que permitan delinear su desarrollo en un marco ético y humanista.*

Sakbe

Laura Cruz Reyes, Claudia Guadalupe Gómez Santillán y Laura Gómez Cruz
sakbe@komputersapiens.org

Empresas mexicanas de IA (3ª parte)

En los dos números anteriores, se ha hablado en esta columna de las empresas mexicanas que actualmente están aplicando Inteligencia Artificial (IA). Ya se ha mencionado a un grupo de empresas que están innovando mediante el uso de bots. Ahora nos enfocaremos en GESEDIG, empresa lanzada por Grupo BPGurus que es la primera en México en tener robots de servicio capaces de ayudar en la detección, tratamiento y recuperación de pacientes con COVID-19.

Mauricio Corona, sociodirector de BPGurus, nos explica que su compañía se dedica a la consultoría de gestión de servicios digitales, al desarrollo de IA, Robótica, Big Data y chatbots en México y Latinoamérica; además señala que él imparte conferencias internacionales sobre transformación digital. En este marzo 2020 mediante la recién creada GESEIG, se hizo el lanzamiento en México de la línea RaaS (Robots as a Service), la cual cuenta con tres modelos de robots humanoides de servicio que pueden ayudar al cuerpo médico a atender pacientes que ingresan como casos sospechosos o confirmados de COVID-19. Por ejemplo, un RaaS puede hacer entrevistas para realizar un prediagnóstico; también puede circular un área mostrando videos informativos y dando indicaciones (como “quédate en casa”) y guiar a los usua-

rios para llegar a sus destinos por medio de sus sensores de movimiento, así como llevar alimentos y medicinas para evitar el contacto entre personas, además de dar seguimiento médico a través de monitores.

Los robots de GESEDIG son de diseño japonés, su ensamblaje se hace en el clúster de robótica e IA más grande del mundo ubicado en China y todos sus modelos de IA están hechos por ingenieros en cibernética mexicanos. Los robots incluyen tecnología de procesamiento de lenguaje natural, sistemas de reconocimiento facial, lectores de código de barras, procesos de CRM (*Customer Relationship Management*) y Click to Call. Estos robots ya se han introducido en el mercado asiático, europeo y de Medio Oriente. Para saber más sobre ellos y conocer las empresas mencionadas, recomendamos revisar los siguientes vínculos.

- Lanzamiento en México de los robots de servicio de GESEDIG: <https://www.youtube.com/watch?v=1fQHmDMIZtw&t=453s>
- Videos de los robots de servicio de GESEDIG: https://www.youtube.com/channel/UC_JSQ82SrXeITf0iF4G7hSw/videos
- Empresa BPGurus: <https://bpgurus.com>
- Empresa GESEDIG: <https://gesedig.com>



Modelos de robots de servicio de GESEDIG.

ARTÍCULO ACEPTADO

COLDPower: el agente inteligente mexicano de mercadeo de energía

Ansel Y. Rodríguez-González, Eduardo F. Morales y Luis Enrique Sucar

En este artículo se presenta a COLDPower: el agente inteligente mexicano de mercadeo de energía, que en diferentes versiones ha competido en el torneo anual PowerTAC desde 2013. Se describe su arquitectura general, los agentes expertos en diferentes mercados y tipos de clientes que lo componen y cómo estos agentes aprenden. El 2do lugar alcanzado en el torneo PowerTAC 2016, y las ganancias obtenidas en las diferentes ediciones del torneo, validan la rentabilidad de COLDPower y la pertinencia del aprendizaje por refuerzo y del uso de múltiples agentes.

Introducción

Un paso importante en el camino hacia las ciudades y las sociedades sostenibles es la penetración masiva de vehículos eléctricos y energía renovable (RE) que utilizan como fuentes primarias el viento y el sol. Este cambio en la matriz energética de los países genera un mayor dinamismo en el creciente fenómeno que representan los mercados de energía. En particular, la oferta y la demanda de energía están influenciadas por factores cambiantes como las condiciones climáticas y los comportamientos de los clientes. Como consecuencia, la adaptabilidad es una habilidad requerida para los agentes de mercadeo también conocidos como corredores. Adicionalmente, la competencia entre los corredores y su adaptabilidad generan mayor dinamismo. En este contexto, es relevante el desarrollo de agentes autónomos que puedan aprender estrategias efectivas al combinar la experiencia con la exploración de nuevos cursos de acción.

Para evitar el riesgo de probar las estrategias de los agentes autónomos en mercados de energía reales, la comunidad científica desarrolló un simulador de mercados de energía llamado PowerTAC [1] y, a partir de 2012, se lleva a cabo un torneo anual (también llamado PowerTAC [2]). En él, agentes diseñados por expertos de diferentes campos de todo el mundo (por ejemplo, AgentUDE [3], COLDPower [4], CrocodileAgent [7], Mercator [8] y SPOT [9]) se validan y compiten entre sí.

Desde 2013, México presenta cada año un agente en el torneo PowerTAC, siendo el único representante de Latinoamérica. COLDPower [4,5,6], el agente inteligente mexicano de mercadeo de energía, no es un desarrollo concluido; ha ido evolucionando desde estar compuesto por un solo agente (desarrollado en el INAOE) hasta convertirse en un sistema multiagente compuesto por múltiples agentes expertos en los diferentes mercados y

tipos de clientes que cooperan para lograr en conjunto mayores ganancias; y a la vez que, como sistema, compite con otros agentes de compra-venta de energía (desarrollado en colaboración entre CICESE e INAOE).

Simulador PowerTAC y Mercados

PowerTAC es un simulador que se compone en dos grupos de elementos. En un grupo se encuentran los corredores que compiten unos contra otros, mientras que en el otro se simulan clientes minoristas, productores, consumidores, grandes productores de energía, la distribución de la energía y los mercados minorista, mayorista y de balance (véase Figura 1). Cada simulación representa aproximadamente dos meses y se realiza en dos horas. La unidad de tiempo usada en PowerTAC es el intervalo de tiempo que representa una hora simulada. Durante cada intervalo de tiempo cada corredor puede publicar tarifas a las que los clientes pueden suscribirse. Posteriormente, pueden negociar la energía sobrante o faltante en el mercado mayorista. Si después de este paso un corredor se encuentra desbalanceado respecto a la energía, puede hacer uso del mercado de desbalance. Al final de cada intervalo de tiempo, los corredores reciben ganancias dependiendo de sus ingresos por ventas, gastos por compras y pagos por desbalance.

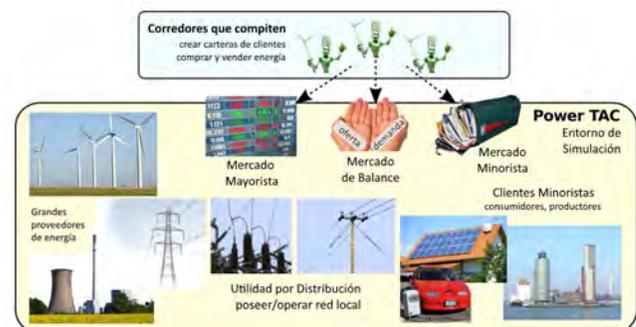


Figura 1. Principales elementos de un escenario PowerTAC [2].

En PowerTAC, el **mercado minorista** es también conocido como mercado de tarifas. En este mercado los corredores ofrecen contratos de tarifas a los clientes. Es importante mencionar que se cobra al corredor una cuota por registrar una tarifa y distribuirla a los clientes. Los clientes pueden ser tanto productores de energía como

consumidores. Al tener una variedad de tarifas ofrecidas por muchos corredores, cada cliente evalúa periódicamente las mismas para decidir cuál es la más conveniente y, si así lo deciden, cambiar su suscripción. Esta conveniencia se relaciona con las características de la tarifa. La principal característica de una tarifa es el precio, que es la cantidad pagada o cobrada por el agente, cuando un MWh de energía se consume o produce. Otras características de la tarifa son la fecha de vencimiento, la cuota de suscripción y la cuota por retirarse de la misma antes de la fecha de vencimiento. Los corredores también pueden reaccionar creando nuevas tarifas para atraer suscripciones de clientes.

El **mercado mayorista** de PowerTAC [2] es una abstracción de los mercados mayoristas diarios e internos, en los que los corredores pueden realizar ofertas continuas (cada hora) para la entrega futura de energía dentro de las siguientes 24 horas. Los corredores interactúan con grandes proveedores y otros participantes en el mercado mayorista y, por lo tanto, el precio de la energía depende exclusivamente de la oferta y demanda de energía en cada subasta. En general, el precio de la energía en el mercado mayorista es más bajo en comparación con los precios en el mercado minorista debido al mayor volumen de energía comercializada.

El **mercado de balance** es un tipo de mercado creado por la incertidumbre asociada con factores difíciles de predecir (como el clima). A pesar de los esfuerzos de los corredores para mantener un balance entre la energía vendida y comprada, la incertidumbre dificulta un balance perfecto [10]. Para garantizar la estabilidad general de la red eléctrica, algunos productores y consumidores proporcionan servicios de compensación a los corredores por una comisión, creando el mercado de balance.

El **torneo PowerTAC** que se realiza cada año consta de cuatro rondas: dos rondas de prueba, una ronda clasificatoria, y una ronda final. Cada ronda tiene una duración de dos semanas, y entre cada ronda y la siguiente transcurren otras dos semanas. Los resultados son puestos a cero al comienzo de cada ronda y solo los diez primeros lugares de la ronda clasificatoria avanzan a la ronda final. Cada ronda consiste de tres bloques de simulaciones, en cada simulación del bloque 1 participan todos los corredores, mientras que en cada simulación del bloque 2 y del bloque 3 participa una combinación de 5 y 3 corredores, respectivamente. El número de simulaciones por bloque ha sido diferente cada año, siendo en el 2019, 40, 224 y 224 simulaciones por cada bloque. Para cada bloque, las ganancias o pérdidas de cada corredor se acumulan tras cada simulación. El resultado por bloque de cada corredor es normalizado teniendo en cuenta la media y la desviación estándar de los resultados de los demás competidores. Mientras mayor es la suma de los resultados normalizados de un corredor, mejor es su lugar en la tabla de posiciones de la ronda. Cada año

crece la complejidad del torneo al cambiar la matriz energética simulada mediante el incremento de productores de energía basadas en viento y sol, la inclusión de clientes con capacidad de disminuir o mover su consumo así como clientes con capacidad de almacenamiento, y mayores factores de penalización por desbalance.

COLDPower por dentro

Anteriormente habíamos mencionado que COLDPower es un corredor compuesto por múltiples agentes expertos en los diferentes mercados y tipos de clientes. Veamos qué es un agente inteligente.

Un agente (Figura 2) es un sistema informático que se encuentra en algún entorno, y que es capaz de actuar de forma autónoma en este entorno para cumplir sus objetivos de diseño [11]. La inteligencia de un agente está dada por su capacidad de actuar de forma flexible, entendiéndose por flexible que es reactivo (responde oportunamente a los cambios), proactivo (exhibe un comportamiento dirigido a un objetivo tomando la iniciativa) y social (interactúa con otros agentes o humanos).

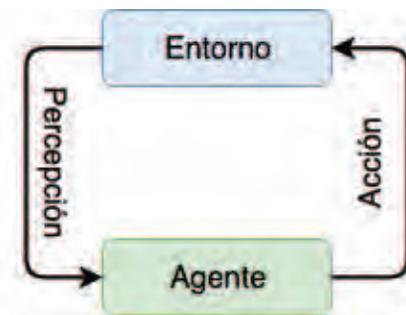


Figura 2. Esquema general de un agente.

COLDPower se compone, desde el punto de vista funcional, de dos partes: una enfocada al manejo de datos, y otra enfocada a los agentes inteligentes con capacidades específicas. Los componentes de software enfocados al manejo de datos proveen información oportuna del entorno a los agentes expertos en los mercados, abstraéndolos del manejo de mensajes del simulador. La última versión del COLDPower (COLDPower 2019) incluye dos agentes expertos en el mercado minorista: uno para clientes productores y otro para clientes consumidores; y, adicionalmente, otro agente para el mercado mayorista. Los dos agentes expertos en el mercado minorista están basados en aprendizaje por refuerzo, mientras el agente para el mercado mayorista utiliza descriptores estadísticos. Los tres agentes se ejecutan en paralelo y, aunque ninguno conoce las acciones realizadas por los demás, observan el mismo entorno y tienen objetivos particulares que aportan al objetivo general: incrementar las ganancias.

Los **agentes expertos en el mercado minorista** (agente para clientes productores y agente para clien-

tes consumidores) ofrecen tarifas para comprar o vender energía. Cada agente comienza a partir de un conjunto de una única tarifa predefinida. En cada paso pueden publicar una nueva tarifa menos (o más) atractiva que las tarifas en el conjunto actual de tarifas propias. El precio es el parámetro usado para variar el atractivo de las tarifas. Además, en cada paso, los agentes propuestos pueden eliminar la tarifa más atractiva (o la menos atractiva) del conjunto de tarifas propias, o simplemente no hacer nada. Dependiendo de la acción ejecutada, los agentes pueden atraer o repeler más o menos clientes. Así pueden lograr que las ganancias en el mercado minorista sean mayores que las penalizaciones por el desbalance de energía y tratar de maximizar su utilidad en cada paso de decisión. Esto es un reto debido a la complejidad del entorno, en el cual varían las condiciones climáticas que afectan la producción y el consumo de energía y los corredores que compiten reaccionan a los cambios y, en particular, a las acciones de otros competidores. En este contexto, los agentes deben aprender qué acción es más conveniente a partir del estado actual del entorno. Para ello, usan aprendizaje por refuerzo, incorporando al modelo de agente inteligente una función de recompensa, conocida como R , que mide el resultado de ejecutar una acción desde un estado, en términos del objetivo que se desee alcanzar. Cada agente es modelado mediante un conjunto de estados, un conjunto de acciones y una función de recompensa (Figura 3).

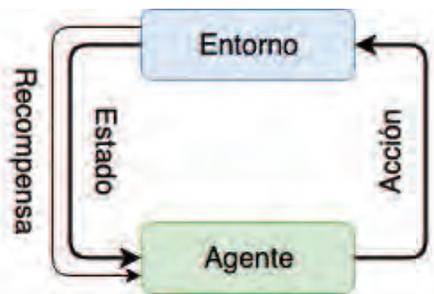


Figura 3. Esquema general de un agente basado en aprendizaje por refuerzo.

En cada paso el agente decide, con cierta probabilidad especificada, ejecutar la acción que para el estado actual tiene mayor calidad. La “calidad” de ejecutar una acción en un estado dado, conocida como valor Q , depende del último valor Q para esa acción, de la recompensa alcanzada la última vez que se ejecutó la acción, y del valor Q máximo entre las posibles acciones futuras que se pueden realizar desde el estado resultante de ejecutar la acción. En cada paso, el valor Q de la acción realizada es actualizado siguiendo la Ecuación 1.

$$Q_t(s, a) = (1 - \alpha)Q_{t-1}(s, a) + \alpha \left(R_t(s, a) + \gamma \max_{a'} Q(s', a') \right) \quad (1)$$

donde s es el estado desde el que se ejecutó la acción a , $Q_{t-1}(s, a)$ es el valor Q para s y a antes de la actualización, α es un factor conocido como factor de aprendizaje, $R_t(s, a)$ es la recompensa alcanzada al ejecutar la acción a desde el estado s , γ es un factor de descuento y $\max_{a'} Q(s', a')$ es el valor Q máximo entre las posibles acciones futuras que se pueden realizar desde el estado resultante s' .

El algoritmo descrito se denomina Q-learning [12]. A mayor α , más se tiene en cuenta la recompensa y el valor Q máximo entre las posibles acciones futuras. Cuando $\alpha = 0$, el valor Q no varía después de la actualización y el agente no aprende, ejecuta lo aprendido.

A las simulaciones de las rondas de prueba y clasificatoria del torneo PowerTAC las consideramos como entrenamiento y fijamos α en un valor cercano a 0.5. En las simulaciones de la ronda final, aunque disminuimos el factor de aprendizaje, no lo fijamos en 0 como podría pensarse en primera opción, pues se espera que los otros agentes cambien su comportamiento y aparezcan nuevas situaciones no previstas.

Para representar los **estados** utilizamos siete variables discretizadas en intervalos. Cada combinación de los valores (intervalos) de las variables representa un estado (resultando en 12,288 estados). Las variables usadas fueron:

- Porcentaje de clientes productores en la cartera de clientes de COLDPower, del total de clientes productores.
- Porcentaje de clientes consumidores en la cartera de clientes de COLDPower, del total de clientes consumidores.
- Relación entre el precio de la última tarifa de consumo y el promedio de los precios en mercado mayorista de la última semana.
- Relación entre el precio de la última tarifa de producción y el promedio de los precios en mercado mayorista de la última semana.
- Variación de las ganancias en el último día.
- Variación de las ganancias en la última semana.
- Relación entre la desviación estándar del balance de energía de COLDPower en la última semana y la diferencia entre el valor de energía máximo (y el valor de energía mínimo) y el promedio de balance de energía de COLDPower en la última semana.

Como **función de recompensa** utilizamos una combinación entre la variación de las ganancias en el último día (desde la última acción) y cuanto representan la misma de la variación diaria promedio de las ganancias en la última semana.

Tabla 1. Resultados del torneos PowerTAC desde 2013 hasta 2019.

Corredor	Institución	País	2013	2014	2015	2016	2017	2018	2019
TacTex	University of Texas at Austin	E.U.A	1		2				
cwiBroker	Centrum Wiskunde & Informatica	Holanda	2	2	7				
MLLBroker	University of Freiburg	Alemania	3						
CrocodileAgent	University of Zagreb	Croacia	4	3	11	7	8	3	3
AstonTAC	Aston University	Inglaterra	5						
Mertacor	Aristotle University	Grecia	6	5	8	5			1
COLDPower	CICESE/INAOE	México	7	6	6	2	6	5	5
AgentUDE	Universitaet Duisburg-Essen	Alemania		1	4	3	1	1	2
maxon16	Westfaelische Hochschule	Alemania		4	1	1	7		
AgentCU	Chinese University of Hong Kong	China			3	6			
Sharpy	Hebrew University of Jerusalem	Israel			5				
Bunnie	Nanyang Technological University	Singapur			9			6	6
SPOT	University of Texas at El Paso/NMS	E.U.A			10	4	3	4	4
fimtac	Universität Augsburg,	Alemania					2		8
Vidyut Vanika	IIIT Hyderabad & TCS	India					4	2	7
EWIIS	University of Cologne	Alemania					5	7	6

Como **función de recompensa** utilizamos una combinación lineal de dos variables: la variación de las ganancias en el último día (desde la última acción) y la variación de las ganancias diarias promedio en la última semana.

El agente experto propuesto para el **mercado mayorista** predice el desbalance energético que pueden causar los agentes expertos en tarifas, así como el precio de la energía del mercado mayorista, y decide cuándo comprar o vender y en qué cantidades para, primero, lograr el balance y, segundo, realizar operaciones especuladoras (comprar energía cuando los precios son bajos y vender energía cuando los precios son altos). Los pasos seguidos por el agente experto en el mercado mayorista son:

1. Obtener una proyección estimada de los futuros precios de la energía y de la cantidad de energía que podría ser negociada para cada una de las 24 oportunidades asociadas a la hora objetivo. La proyección es obtenida a partir del historial de precios de la energía y de las cantidades de energía negociada en dicho mercado para la hora objetivo deseada. Aunque existen técnicas más complejas, nosotros usamos simplemente la media ponderada, con pesos que disminuyen en la medida en que los valores son más viejos.
2. Decidir, a partir de la proyección estimada de precios de la energía y para cada una de las oportunidades asociadas a la hora objetivo, si es una oportunidad de compra, de venta, o no se negociará en la misma. Para esto se usa la media y la desviación estándar de los 24 precios estimados para una hora objetivo. Si el precio estimado se encuentra por encima de la media más un factor de la desviación estándar, se considera como una oportunidad de venta. Si el precio estimado se encuentra debajo de la media menos un factor de la desviación estándar, se considera como una oportunidad de compra. En otro caso, no se negocia.

3. Calcular el precio de la orden de compra o venta de energía para la hora objetivo (solo en caso de ser la hora actual una oportunidad de compra o venta), a partir de la proyección estimada de precios de la energía. El precio de la orden es obtenido haciendo más atractivo el precio estimado, disminuyéndolo o aumentándolo, según sea el caso.
4. Calcular la cantidad de energía a negociar en la hora actual, a partir de la proyección estimada de energía a negociar, de la energía de las ordenes ejecutadas, del balance estimado de energía de los clientes para la hora objetivo y de un umbral de cantidad máxima de energía para especular.
5. En caso de ser la hora actual una oportunidad de compra o venta de energía, enviar la orden al mercado por el precio y cantidad calculados.

Resultados

Las diferentes versiones de COLDPower han sido validadas en los torneos PowerTAC desde 2013 (Tabla 1). Las dos primeras apariciones del agente mexicano fueron infructuosas, quedando en la última posición. Posteriormente los resultados de las nuevas versiones de COLDPower han sido mejores, siendo muy destacable el resultado de 2016 en el que se logró el 2do lugar.

En la Tabla 2 se puede observar que las ganancias de COLDPower en 2016 fueron superiores a las de la mayoría de los agentes, posicionándose como 2do lugar en cada bloque. Nótese que varios de los corredores tienen pérdidas.

Tabla 2. Resultados del torneo PowerTAC 2016.

Corredor	Bloque 1	Bloque 2	Bloque 3	Posición
maxon	28150889	58222359	162300823	1
COLDPower	18919929	35255197	86548360	2
AgentUDE	9545250	13382258	79616833	3
SPOT	-91040	9152999	52695182	4
Mertacor	628193	-139322	-404205	5
AgentCU	-22164073	-22755943	81582603	6
CrocodileAgent	-40352748	-78593294	-186380292	8

En 2019, COLDPower logró el 3er lugar en la última fase de ronda clasificatoria que incluyó 100 simulaciones de ocho corredores (Tabla 3), obteniendo un saldo positivo superior a la mayoría de los competidores. Durante esta ronda los equipos pueden ajustar sus corredores. Debido al aprendizaje por refuerzo, COLDPower pudo adaptarse a estos cambios.

Tabla 3. Resultados ronda clasificatoria PowerTAC 2019.

Corredor	País	Ganancias
CrocodileAgent19	Croacia	280839063
Mertacor2019	Grecia	82193180
COLDPower	México	20532549
SPOT	E.U.A	17026871
Bunnie19	Singapur	5351931
EWIIS3	Alemania	-3379208
Vidyut Vanika	India	-34470448
AgentUDE	Alemania	-2524117349

Las dos semanas entre la ronda clasificatoria y la final fueron aprovechadas por los equipos para realizar cambios de mayor envergadura a los corredores, provocando nuevos comportamientos en la ronda final. Aun así, COLDPower pudo adaptarse, requiriendo 27 simulaciones del bloque de ocho corredores de la ronda final, actualizando lo aprendido, para lograr un crecimiento sostenido del saldo acumulado (véase Figura 3). A diferencia de la ronda clasificatoria, que incluyó 100 simulaciones para este bloque, la ronda final solo incluyó 40, lo que impidió a COLDPower recuperarse del impacto de los errores cometidos durante el aprendizaje y alcanzar un saldo acumulado positivo. Para los bloques de cinco y tres corredores de la ronda final que incluyeron muchas más simulaciones en las que COLDPower participó (140

por cada una) su saldo acumulado fue positivo (véase Tabla 4).

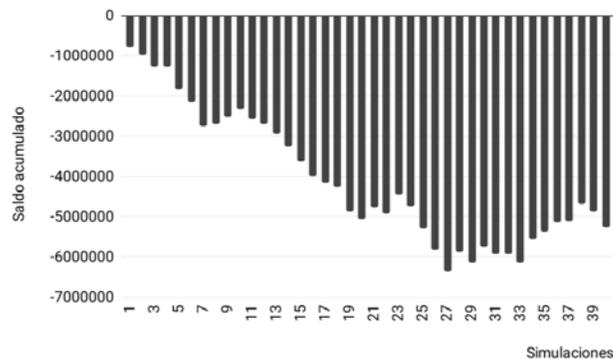


Figura 3. Saldo acumulado de COLDPower en bloque de ocho corredores de la ronda final PowerTAC 2019.

Tabla 4. Resultados del torneo PowerTAC 2019.

Corredor	Bloque 1	Bloque 2	Bloque 3	Posición
Mertacor	19944504	165444489	174069791	1
AgentUDE	645558	58173593	230899300	2
CrocodileAgent	3641931	45136148	204060495	3
SPOT	3091614	72426787	158913062	4
COLDPower	-5278487	10278966	44510150	5
Bunnie	2199767	283143	-671940	6
EWIIS3	-32357691	-36562710	164939470	7
VidyutVanika	-20215079	-188244022	-31821230	8

El saldo total de COLDPower en la ronda final del torneo PowerTAC 2019 fue positivo, posicionándose como un corredor rentable en el 5to lugar, lo cual muestra la efectividad de su composición como un sistema basado en múltiples agentes expertos, así como en el uso del aprendizaje por refuerzo.

COLDPower, es un sistema multiagente compuesto por múltiples agentes expertos en los diferentes mercados y tipos de clientes que cooperan para lograr en conjunto mayores ganancias.

Conclusiones

En este artículo hemos presentado a COLDPower: el agente inteligente mexicano de mercadeo de energía. Se introdujeron al lector algunos conceptos como mercados de energía, mercado minorista, mayorista y de balance, y el funcionamiento del simulador PowerTAC. Finalmente, se mostraron los resultados alcanzados en cada participación de COLDPower en el torneo PowerTAC.

El 2do lugar alcanzado en el torneo PowerTAC 2016, y las ganancias obtenidas en diferentes ediciones del torneo, aun en escenarios donde las estrategias de los otros corredores cambian, validan la composición de COLDPower

como un sistema basado en múltiples agentes expertos, así como el uso del aprendizaje por refuerzo. No obstante, COLDPower no es un trabajo concluido y no es el fin. Por un lado, la complejidad de los mercados crece cada año con el cambio en la matriz energética, los mecanismos de predicción de COLDPower son mejorables, acelerar el aprendizaje es una necesidad, y la implementación de agentes expertos en el mercado de balance es una oportunidad no explorada. Por otro lado, la experiencia obtenida durante el desarrollo de COLDPower abre nuevas líneas de trabajo, como el diseño de un simulador de los mercados energéticos apegado a la

nueva normatividad mexicana, y el desarrollo de corretores capaces de entender la dinámica de los mercados mexicanos y hacer ofertas exitosas de mercadeo de energía en los mismos.*

REFERENCIAS

1. Ketter, W., Collins, J. y Reddy P. (2013). Power TAC: A competitive economic simulation of the smart grid. *Energy Economics*, 39, 262-270.
2. Ketter, W., Collins, J., y Weerdt, M.D. (2017). The 2018 power trading agent competition. ERIM Report Series Reference No. 2017-016-LIS. Recuperado de: <https://ssrn.com/abstract=3087096>.
3. Özdemir, S. y Unland, R. (2018). AgentUDE17: a genetic algorithm to optimize the parameters of an electricity tariff in a smart grid environment. *Lectures Notes in Computer Science*, 10978, 224-236.
4. Serrano, J., Rodríguez-González, A. Y., y de Cote, E.M. (2017). Fixed-Price Tariff Generation Using Reinforcement Learning. *Studies in Computational Intelligence*, 674, 121-136.
5. Serrano, J., Rodríguez-González, A.Y., Alonso M.P., de Cote, E.M., y Sucar, L.E. (2015). Distributed energy procurement and management in smart environments. En *2015 IEEE First International Smart Cities Conference (ISC2)*, Guadalajara. 1-6.
6. Rodríguez-González, A.Y., Alonso, M.P., Lezama, F., Rodríguez, L., de Cote, E.M., Morales, E.F., Sucar, L.E., y Crockett, D.D. (2019). A competitive and profitable multi-agent autonomous broker for energy markets. *Sustainable Cities and Society*, 49, 101590.
7. Grgic, D., Vdovic, H., Babic, J., y Podobnik, V. (2019). CrocodileAgent 2018: Robust agent-based mechanisms for power trading in competitive environments. *Computer Science and Information Systems*, 16(1), 105-129.
8. Ntagka, E., Chrysopoulos, A., y Mitkas, P.A. (2014). Designing tariffs in a competitive energy market using particle swarm optimization techniques. *Lecture Notes in Business Information Processing*, 187, 129-143.
9. Chowdhury, M.M.P., Folk, R.Y., Fioretto, F., Kiekintveld, C., y Yeoh, W. (2015). Investigation of learning strategies for the spot broker in power tac. *Lecture Notes in Business Information Processing*, 271, 96-111.
10. Casazza, J., y Delea, F. (2003). Understanding electric power systems: an overview of the technology and the marketplace. John Wiley & Sons. ISBN 978-0-471-66786-5, 232.
11. Wooldridge, M. (1999). Intelligent agents. En Weiss G. (Ed.) *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*. MIT Press, 27-77.
12. Watkins, C., y Dayan, P. (1992). Q-Learning. *Machine learning*, 8, 279-292.

SOBRE LOS AUTORES



Ansel Y. Rodríguez González. SNI Nivel I. Recibió los grados de Licenciado en Ciencias de la Computación y Maestro en Ciencias Matemáticas en la Universidad de la Habana, Cuba en 2004 y 2007 respectivamente. Recibió el grado de Doctor en Ciencias Computacionales en el INAOE en 2011. Ha participado en 15 proyectos de investigación y desarrollo. Actualmente tiene una cátedra CONACyT en la Unidad de Transferencia Tecnológica (UT3) del CICESE. Sus intereses de investigación incluyen tópicos de inteligencia artificial como minería de datos, reconocimiento de patrones, aprendizaje por refuerzo y algoritmos evolutivos.



Eduardo F. Morales. SNI Nivel III. Licenciatura en Ingeniería Física de la UAM-Azcapotzalco, maestría en Inteligencia Artificial de la Universidad de Edimburgo y Doctorado en Computación del Turing Institute - Universidad de Strathclyde. Ha sido responsable de 28 Proyectos de investigación y desarrollo y ha participado en otros 22. Tiene más de 150 publicaciones en foros internacionales. Actualmente labora en el INAOE en donde realiza investigación en Aprendizaje Computacional y Robótica.



Luis Enrique Sucar Succar. Premio Nacional de Ciencias 2016. SNI Nivel III. Tiene una maestría en Ingeniería Eléctrica por la Universidad de Stanford y el Doctorado en Computación por el Imperial College, Londres. Es Investigador Titular del INAOE, especialista en técnicas probabilísticas en inteligencia artificial y sus aplicaciones en visión computacional, robótica, biomedicina y energía. Cuenta con más de 30 años de experiencia en investigación y desarrollo. Es miembro de la Academia Mexicana de Ciencias, la Academia de Ingeniería y Senior Member del IEEE.

ARTÍCULO ACEPTADO

Estimación automática de precios de vivienda mediante inteligencia artificial

Roberto Quintero, Cynthia B. Pérez, Laura P. López-Arredondo, Jesús R. Villavicencio, Cesar Puente y Juan Villegas

La Importancia de la valuación de viviendas en México

En la actualidad, la industria inmobiliaria presenta un constante crecimiento en el mercado, ya que resulta esencial para diversos sectores económicos alrededor del mundo contar con tasaciones precisas de bienes inmuebles [1,2]; para esto, el proceso de valuación de viviendas consiste en establecer el valor que le corresponde tomando en cuenta distintos factores o atributos de la propiedad. Sin embargo, llevar a cabo la estimación del precio de las viviendas de manera acertada y precisa puede resultar bastante complejo debido a la variabilidad de los factores. Es por ello que la naturaleza volátil del mercado inmobiliario ha generado la necesidad de mejorar los métodos para valuar los precios de los inmuebles, involucrando características de viviendas, modelos estadísticos y predictivos confiables donde se busca reducir al máximo el error de las estimaciones, es decir, de la valuación del bien inmueble.

Para denotar su importancia global basta con analizar algunas cifras obtenidas de diversos estudios inmobiliarios realizados en distintos países [3]. En Estados Unidos, se ha registrado un incremento promedio anual del 2.2% del PIB (Producto Interno Bruto) desde el tercer trimestre de 2009. En Filipinas, el PIB presentó un crecimiento del 6.7% en 2016, lo cual representó un incremento bastante acelerado con respecto a los años anteriores, ya que muchos inversionistas alrededor del mundo han comprado diversas propiedades en todas las islas [4]. En México, la participación de los servicios inmobiliarios en la economía ha ido en aumento y en 2019 ya representaba el 11% del PIB total [5]. Así mismo, para finales de 2017 hubo un notable incremento en la construcción de viviendas multifamiliares dentro del núcleo urbano de las ciudades, correspondiente al 23% de todas las unidades en construcción y posicionando las mejores oportunidades de desarrollo en los suburbios [6]. De igual manera, existen muchos otros países en donde la industria y los proyectos inmobiliarios juegan un rol muy importante en la economía. Un estudio reciente [7] analiza el crecimiento de esta industria en Beijín, China, acorde a un modelo multivariable para poder hacer predicciones de precios en bienes raíces ante situaciones cambiantes y complejas del mercado. Debido a la importancia que sigue generando el mercado inmobiliario

en la economía de los países ha surgido la necesidad de implementar métodos de valuación de bienes inmuebles, que permitan estimar los precios de viviendas de forma rápida y precisa a través de plataformas tecnológicas. En la revista Expansión [8], se argumenta que el rol de los valuadores se ha complementado por el interés de las empresas de tecnologías en los procesos inmobiliarios. Por lo tanto, el estudio de modelos predictivos, así como el uso de tecnologías y plataformas que implementan algoritmos de aprendizaje automático para analizar cientos de datos sobre cada propiedad ha ganado popularidad, generando nuevas oportunidades para el crecimiento económico, la innovación y la construcción del contexto inmobiliario. Muestra de ello es que, en los últimos años en México, más del 50% de las búsquedas inmobiliarias fueron realizadas a través de plataformas de Internet pasando de 21 millones a 42 millones de búsquedas [4]. Lo anterior se debe a que dichas tecnologías sirven como apoyo a los propietarios de inmuebles, agentes de bienes raíces, bancos, instituciones gubernamentales, entre otros, para conocer el valor estimado de las viviendas en tiempo real a través de la automatización basada en herramientas de cómputo, marcando una tendencia clara hacia la implementación de la Inteligencia Artificial (IA).

Estimación automática de precios de viviendas

La dificultad más relevante en el pronóstico de los precios inmobiliarios siempre ha sido la subjetividad de los juicios a procesar para obtener valores inmobiliarios acertados y confiables [9]. En ese sentido, Esquer [10] menciona que los métodos manuales de valuación suelen ser bastante lentos y tediosos debido a que requieren de una investigación fundamentada que suele tomar tiempo, además de la participación de personas capacitadas en valuación de viviendas. Schram (2006) y Diaz & Hansz (2002) [11,12] explican ocho pasos necesarios para llevar a cabo la valuación de viviendas de forma tradicional (ver Figura 1). En general, los procedimientos actuales deben basarse en información confiable para obtener resultados más precisos.

En Estados Unidos, los emprendedores han intentado incrementar la eficiencia de los corredores hipotecarios disminuyendo costos de transacción y el acceso a la información de los mercados. Esto conduce a la necesidad

de incorporar herramientas tecnológicas durante el proceso de valuación de tal forma, que permitan interpretar los fenómenos típicos del mercado inmobiliario [13,14]. De esta manera, los datos son transformados en conocimiento mediante el uso de inteligencia artificial con el fin de automatizar los procesos y visualizar las estimaciones en tiempo real. Es por ello que la implementación de algoritmos predictivos y de aprendizaje automático en el contexto de bienes raíces ha tenido un notable crecimiento en la industria inmobiliaria extranjera. No obstante, la búsqueda por robustecer el ajuste de precios de viviendas disminuyendo el margen de error en dichas predicciones es continuo, esto con la finalidad de ofrecer estimaciones más acertadas y cercanas a la realidad sobre el valor de los bienes inmuebles. En consecuencia, existen diversos estudios en la literatura que analizan la precisión de las estimaciones haciendo uso de los algoritmos de aprendizaje automático. Entre los más comunes destacan las redes neuronales artificiales (RNA) [15-18], los modelos de regresión [19], las máquinas de soporte vectorial [20], y los algoritmos evolutivos [9], entre otros. Para que este tipo de técnicas obtengan una estimación certera es crucial la selección adecuada de los factores implicados en la valoración de viviendas. Actualmente, el estudio de tecnologías y plataformas que implementan algoritmos predictivos y de aprendizaje automático para estimar el valor de propiedades es de gran interés para este mercado; aunque, son pocas las empresas inmobiliarias en México que hacen uso de este tipo de tecnología, así como escasos los estudios que realizan un análisis previo sobre la importancia de los factores involucrados en la estimación de precios de viviendas [21,22].

Es por ello que este trabajo se enfoca en analizar los principales factores para la estimación de precios de viviendas mediante el uso de herramientas de inteligencia artificial, utilizando solo aquellos factores predominantes sin afectar la precisión de la estimación en el contexto del mercado inmobiliario en México. Lo anterior, ayudaría a establecer la estimación automática por medio de una plataforma web donde el usuario (comprador, vendedor, agente) introduciría solamente la información de los factores más relevantes y con ello, se desplegaría automáticamente el precio que le correspondería a esa vivienda en particular.

El análisis del mercado inmobiliario no solo es relevante para las estimaciones de bienes raíces, sino también para las inversiones y cuestiones fiscales [9]. Es por ello, que existe un gran interés por el uso de algoritmos de inteligencia artificial, ya que ayudan a interpretar los fenómenos típicos del mercado inmobiliario reduciendo el tiempo del proceso de estimación de bienes inmuebles. Además de poder visualizar los resultados de las estima-

ciones desde una plataforma web a través de cualquier dispositivo con conexión a Internet. Para esto, algunas plataformas estadounidenses como Intelimétrica¹, Redfin² y Zillow³, están revolucionando el mercado inmobiliario aplicando una filosofía de análisis intensivo de datos mediante el uso de algoritmos de inteligencia artificial para democratizar el acceso a la información facilitando la toma de decisiones.



Figura 1. Proceso de valuación tradicional en bienes raíces [11,12].

Selección de factores para la estimación de precios de viviendas

El proceso relacionado con la estimación del precio de venta de las viviendas agrupa diversas características o factores relevantes que influyen directamente en la estimación del valor del bien inmueble. Esta información corresponde, en su mayoría, a los factores o atributos principales de la vivienda como aquellos relacionados con: la ubicación, el tipo de propiedad, las condiciones de la vivienda, las proporciones de la vivienda, sus instalaciones

¹www.intelimetrica.com

²www.redfin.com

³www.zillow.com

y los sitios cercanos a la vivienda (ver Tabla 1). De manera más específica, Komagome-Towne [23], menciona que las principales características utilizadas para la predicción de los precios corresponden a los pies cuadrados de la vivienda, el tamaño del lote, el número de habitaciones, el número de baños, el año de construcción de la casa y la fecha de venta. No obstante, muchos otros factores pueden llegar a afectar en mayor o menor medida las estimaciones del precio de los inmuebles.

Tabla 1. Descripción de los factores.

Factor	Descripción de los factores
ID	Identificador del registro del avalúo de la vivienda.
Código Postal	Código postal de la vivienda
Calle	Calle en la que se sitúa la vivienda
Lote	Descripción del lote si es que lo requiere
Manzana	Descripción o nombre de la manzana
Colonia	Nombre de la colonia
Valor concluido	Representa el valor que definieron los valuadores de la empresa inmobiliaria
Numero	Número de casa
Clase	Nivel del interés social
Tipo Inmueble	Si se trata de departamento, casa, etc.
Años	Años de la vivienda
Vida Útil Remanente (VUR)	Periodo de tiempo probable, expresado en años, que se estima funcionará la vivienda
Recamaras	Número de recámaras
Baños	Número de baños
Niveles	Número de pisos
Estacionamientos	Si cuenta con estacionamiento y cuántos tiene
Superficie Terreno	Superficie en metros cuadrados
Superficie Construida	Superficie en metros cuadrados construido
Superficie Accesoría	Superficie de construcción en metros cuadrados (área no habitable: cocheras, porches, pórticos, terrazas)
Superficie Vendible	Total de metros cuadrados de la construcción (área habitable)
Valor Físico del Terreno	Valor en pesos (MXN) del terreno
Valor Físico Construido	Valor en pesos (MXN) de lo construido
Valor del Mercado	Valor en pesos (MXN) de acuerdo al criterio del mercado
Valor Unitario Homologado de Ventas	Valor unitario aplicable con afectación según los comparables de mercado (la afectación se deriva de la comparación del sujeto con los comparables según la edad, estado de conservación, ubicación, superficie de construcción etc.)
Valor Unitario Homologado del Terreno	Valor unitario aplicable con afectación según los comparables de mercado (la afectación se deriva de la comparación del sujeto con los comparables según la ubicación y deseabilidad del terreno etc.)
Valor Reposición Nuevo	Es el valor por metros cuadrado de construcción en estado de conservación nuevo. Se aplica según el tipo de construcción
Fecha Avalúo	Fecha en que se realizó el avalúo
Latitud	Latitud de la ubicación
Longitud	Longitud de la ubicación

En los últimos años, las técnicas predictivas han sido de gran utilidad para ayudar a facilitar y precisar la valoración de viviendas, así como conocer cuántos y cuáles factores considerar para la estimación del inmueble. Para esto, la IA provee de herramientas que ayudan a automatizar el proceso de selección de factores reduciendo

su número a aquellos que tengan un mayor impacto en la estimación de precios sin afectar su precisión. En ese sentido, existen diversos métodos para la selección de factores mediante la reducción de la dimensionalidad [24]. En este trabajo se utilizaron cuatro métodos: la técnica de filtrado (FM, Filter Method), envoltura por eliminación hacia atrás (WM-BE, Wrapper Method-Backward Elimination), envoltura por eliminación recursiva (WM-RFE, Wrapper Method-Recursive Feature Elimination) y método embebido (EM, Embedded Method). Para esto, la empresa inmobiliaria cuenta con 29 factores en su base de datos, los cuales describen las características de 1842 viviendas ubicadas en Ciudad Obregón, Sonora (ver Tabla 1). De estos 29 factores se redujo a 11 mediante la aplicación de los métodos mencionados anteriormente (ver Tabla 2). Si bien es deseable que la variable tiempo u otra relacionada con la depreciación del inmueble fuera considerada de forma explícita, ésta es contemplada en el cálculo del factor “valor del mercado”, presentado en la Tabla 1, la cual es calculada por el personal experto en el área de valuación de la empresa bajo estudio.

Tabla 2. Comparación de la selección de factores.

Factores	FM	WM-BE	WM-RFE	EM	Frecuencia de factores seleccionados
ID					
Código Postal		1	1		2
Calle					
Lote					
Manzana					
Colonia					
Tipo Inmueble					
Clase					
Años		1	1		2
Vida Útil Remanente (VUR)		1			1
Recamaras	1	1	1		3
Baños	1		1		2
Niveles			1		1
Estacionamientos			1		1
Superficie Terreno	1	1			2
Superficie Construida		1			1
Superficie Accesoría		1	1		2
Superficie Vendible	1	1	1		3
Valor Físico del Terreno	1	1			2
Valor Físico Construido		1			1
Valor del Mercado	1	1		1	3
Valor Unitario Homologado		1	1		2
Valor Unitario Homologado del Terreno	1	1	1		3
Valor Reposición Nuevo		1			1
Fecha Avalúo					
Latitud					
Longitud					

Como el número de factores es muy grande para establecer automáticamente el precio que le corresponde a cada una de las viviendas, es necesario reducir su dimensionalidad utilizando algoritmos de IA como el método de filtrado, de envoltura y embebido. Al hablar de reducción de dimensionalidad se entiende desde dos

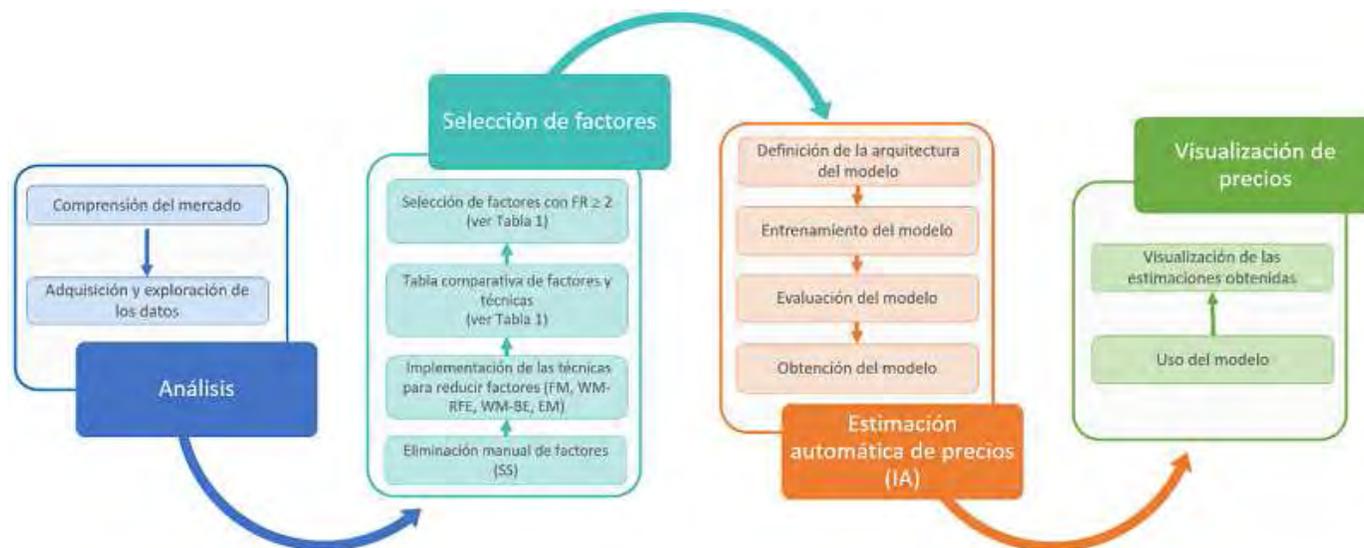


Figura 2. Proceso de estimación automática de precios de viviendas.

perspectivas, la primera es aplicar algoritmos del tipo PCA donde se hace una reducción de variables analizando las más representativas y a partir de ellas rehacer o declarar un nuevo conjunto de variables reducido; la segunda perspectiva es reducir la cantidad de variables por discriminación de representatividad, sin alterar los valores de las variables seleccionadas. En nuestra propuesta estamos considerando el segundo enfoque [25]. Para esto, se siguió la metodología que se presenta en la Figura 2, donde primero fue necesario llevar a cabo un análisis y comprensión del mercado inmobiliario, así como la exploración de los datos. Posteriormente, se realiza la selección de factores donde es necesario que primero se eliminen manualmente aquellos factores que contengan información nula o inconsistente. Una vez preprocesada la información se aplican cuatro métodos de reducción de dimensionalidad (FM, WM-BE, WM-RFE, EM), para seleccionar aquellos factores relevantes que serán utilizados en la estimación de precio. Esta selección se lleva a cabo mediante la comparación del número de factores seleccionados por cada uno de los métodos, donde la frecuencia sea igual o mayor a dos factores (ver Tabla 2). Por lo tanto, los factores con mayor influencia fueron: código postal, años, recámaras, baños, superficie del terreno, superficie accesoria, superficie vendible, valor físico del terreno, valor de mercado, valor unitario homologado y valor unitario homologado del terreno. Así mismo, el método de reducción de dimensionalidad seleccionado fue el WM ya que, en conjunto, el WM-BE y WM-RFE, es donde se presentaron la mayor cantidad de factores.

Estimación automática de precios mediante redes neuronales artificiales (RNA)

La estimación automática de precios se llevó a cabo mediante el uso de RNAs (ver Figura 2). Para esto, los datos registrados en la base de datos se dividieron para la etapa de entrenamiento y para la etapa de prueba. En total son 1842 datos de los cuales, el 80 % fue utilizado para entrenamiento y el 20 % para prueba, con una arquitectura en la red como se muestra en la Tabla 3. Un ejemplo de los resultados obtenidos se presenta en la Tabla 4, ahí se observa la estimación del precio de 10 viviendas seleccionadas aleatoriamente de las 368 correspondientes al 20 % de pruebas. Estos resultados muestran la estimación obtenida mediante RNAs y la estimación obtenida por medio de los expertos en bienes raíces; en promedio la valoración realizada por el algoritmo de red neuronal tiene una correlación (R2) o efectividad del 95 % con un error del 5 % para todos los datos de prueba, lo que significa que el algoritmo puede dar una estimación con un error en promedio del 5 % con respecto al valor real de la propiedad (ver Tabla 5).

Tabla 3. Arquitectura de la RNA.

Característica	Descripción
No. de Neuronas por Capas	10-7-1
Función	ReLU
Método de optimización	L-BFGS-B
Épocas	600
Tasa de aprendizaje (α)	0.0001

Tabla 4. Resultados de la estimación de precios de viviendas mediante redes neuronales.

Estimación del precio de viviendas			
	RNA	Valor Concluido	Diferencia
1	1983339.8	1905200	78139.8
2	655326.9	655190	136.9
3	972386.8	972790	-403.2
4	266524.9	265680	844.9
5	690236.8	690410	-173.2
6	1804916.9	1804900	16.9
7	2595231.3	2512300	82931.3
8	880107.9	880600	-492.1
9	523908.7	523700	208.7
10	337977.9	337900	77.9

Tabla 5. Resultados del error en la estimación de precios de viviendas mediante redes neuronales.

Modelo	MSE	RMSE	MAE	R2
RNA para la estimación de precios	60688.124	246.350	114.492	0.95

MSE= Error cuadrático Medio (Mean Squared Error)
 RMSE= Raíz del Error cuadrático Medio (Root Mean Squared Error)
 MAE= Error Absoluto Medio (Mean Absolute Error)
 R2= Correlación

Conclusiones

Este artículo presenta el proceso para la estimación automática de precios de viviendas mediante RNA. Para ello, se colaboró con una empresa de bienes raíces ubicada en Ciudad Obregón, Sonora, con más de 20 años de experiencia. En ese sentido, primero fue necesario seleccionar los factores más relevantes mediante cuatro métodos de reducción de dimensionalidad para llevar a cabo la estimación de forma automática mediante redes neuronales. Se entrenó la red neuronal con un total de 1474 datos históricos de la empresa para posteriormente probar el modelo con 368 datos.

Los resultados confirman que es posible llevar a cabo la estimación automática de precios de viviendas ya que el porcentaje de error máximo permitido por la empresa es del 10 % y se obtuvo mediante la técnica de IA un error del 5 % muy por debajo del establecido por la empresa. Adicionalmente, el tiempo de estimación fue reducido de 72 hrs a 2 segundos, tomando en cuenta el tiempo que tardó la plataforma al usar el modelo obtenido por la red neuronal y presentar la estimación del inmueble. Así mismo, se está trabajando en la implementación de una plataforma geovisual interactiva la cual utiliza el algoritmo de RNA descrito para estimar automáticamente los precios de viviendas. Se espera que esta plataforma pueda ser utilizada por otras inmobiliarias para estimar automáticamente los precios de viviendas reduciendo con ello, costos y tiempos requeridos en este proceso.*

Agradecimientos. Publicación financiada con recursos PFCE 2019.

REFERENCIAS

- Ghysels, E., Plazzi, A., Valkanov, R., y Torous, W. (2013). Forecasting Real Estate Prices. *Handbook of Economic Forecasting*, 509-580.
- The Economist (2019). For now, residential-property prices are likely to keep rising. Recuperado el 22 de octubre de 2019, de <http://www.economist.com/graphic-detail/2019/06/29/for-now-residential-property-prices-are-likely-to-keep-rising>.
- Guo, H., Jiang, H., Zhang, W., Cui, F., Zhang, R., He, Z., Lun, Y., Chen, Q., y Chen, Q. (2019). Real Estate Level Forecasting - Review. *Business, Management and Economics Research*. 5(54), 57-61.
- Lamudi. (2017). Informe del Mercado Inmobiliario 2017: Internet, penetración y tendencias. México. Recuperado el 8 de octubre de 2018, de <https://www.lamudi.com.mx/informe-mercado-inmobiliario-2017>.
- Balbuena, F., Vázquez S., y Serrano C. BBVA Research (2019). Situación Inmobiliaria México. Primer semestre 2019. Recuperado el 8 de octubre de 2019, de <https://www.bbvaesearch.com/publicaciones/situacion-inmobiliaria-mexico-primer-semestre-2019/>.
- CBRE Research. (2018). 2018 Real Estate Market Outlook - United States. Recuperado de <https://www.cbre-ea.com/docs/default-source/default-document-library/2018-u-s-real-estate-market-outlook.pdf>.
- Li, Y., Xiang, Z., y Xiong, T. (2020). The Behavioral Mechanism and Forecasting of Beijing Housing Prices from a Multiscale Perspective. *Discrete Dynamics in Nature and Society*, 2020, 1-13.
- Expansión (2017). Los start-ups que saben, exactamente, cuánto cuesta tu casa. Recuperado el 4 de marzo de 2017, de <https://www.pressreader.com/mexico/expansion-m%C3%A9xico/20170115/281621010028736>.
- Del Giudice, V., De Paola, P., y Forte, F. (2017). Using Genetic Algorithms for Real Estate Appraisals. *Buildings*, 7(2), 1-12.
- Esquer, J.A. (2012). Estimación del precio de venta de la vivienda en la ciudad de Hermosillo aplicando el método de precios hedónicos. Tesis de maestría en Ciencias de la Ingeniería. Universidad de Sonora.
- Schram, J.F. (2006). Real Estate Appraisal. Rockwell Publishing.
- Diaz, J., y Hansz, J. A. (2002). Behavioral Research into the Real Estate Valuation Process: Progress Toward a Descriptive Model. En *Real Estate Valuation Theory*, 3-29.
- Manganelli, B., De Mare, G., y Nesticò, A. (2015). Using genetic algorithms in the housing market analysis. En *International Conference on Computational Science and Its Applications*, 36-45.
- Muralidharan, S., Phiri, K., Sinha, S.K., y Kim, B. (2018). Analysis and prediction of real estate prices: A case of the Boston housing market. *Issues in Information Systems*, 19(2), 109-118.
- Bency, A.J., Rallapalli, S., Ganti, R.K., Srivatsa, M., y Manjunath, B.S. (2017). Beyond Spatial Auto-Regressive Models: Predicting Housing Prices with Satellite Imagery. En *IEEE winter conference on Applications of Computer Vision (WACV)*. Santa Rosa, California, USA, 320-329.
- Han, S., Ko, Y., Kim, J., y Hong, T. (2018). Housing Market Trend Forecasts through Statistical Comparisons based on Big Data Analytic Methods. *Journal of Management in Engineering*, 34(2), 1-12.

17. Wang, L., Chan, F. F., Wang, Y., y Chang, Q. (2016). Predicting public housing prices using delayed neural networks. En *IEEE Region 10 Conference (TENCON)*, Marina Bay Sands, Singapore, 3589-3592.
18. Lim, W. T., Wang, L., Wang, Y., y Chang, Q. (2016). Housing price prediction using neural networks. En *IEEE 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. Changsha, China, 518-522.
19. Ozgur, C., Hughes, Z., Rogers, G., y Parveen, S. (2016). Multiple Linear Regression Applications in Real Estate Pricing. *International Journal of Mathematics and Statistics Invention (IJMSI)*, 4(8), 39-50.
20. Mu, J., Wu, F., y Zhang, A. (2014). Housing value forecasting based on machine learning methods. En *Abstract and Applied Analysis*, 2014. Hindawi.
21. Ma, Y., Zhang, Z., Ihler, A., y Pan, B. (2018). Estimating Warehouse Rental Price using Machine Learning Techniques. *International Journal of Computers, Communications & Control*, 13(2).
22. Lopez-Arredondo, L. P., Perez, C. B., Castro, L. A., y Rodriguez, L. F. (2019). Estudio sobre la Percepción de los Factores Involucrados en la Estimación de Precios de Viviendas: El Caso de Cajeme. *Información tecnológica*, 30(2), 11-24.
23. Komagome-Towne, A. (2017). Models and Visualizations for Housing Price Prediction. Tesis Doctoral, California State Polytechnic University, Pomona, USA.
24. Guyon, I., y Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157-1182.
25. Fukunaga, K. (1990). Introduction to statistical pattern recognition. *Academic Press Professional*. San Diego, CA, USA.

SOBRE LOS AUTORES



Roberto Quintero es estudiante de la Maestría en Tecnologías de la Información para los Negocios del Instituto Tecnológico de Sonora. Sus intereses de investigación incluyen aprendizaje automático y ciencia de datos.



Cynthia B. Pérez obtuvo el grado de Doctor en Ciencias de la Computación en la Universidad de Extremadura en Mérida, España en 2014. Actualmente es profesor investigador en el Instituto Tecnológico de Sonora, Unidad Guaymas. La Dra. Pérez es miembro del Sistema Nacional de Investigadores y participa como revisor de artículos científicos en el área de Inteligencia Artificial. Sus intereses de investigación incluyen aprendizaje automático, cómputo evolutivo, visión por computadora e inteligencia de negocios.



Laura López-Arredondo es maestra en Tecnologías de la Información para los Negocios por el Instituto Tecnológico de Sonora. Actualmente trabaja como ingeniero de pruebas y calidad en Novutek SC, Ciudad Obregón, Sonora.



Jesús R. Villavicencio obtuvo el grado de Doctor en Ciencias en Electrónica y Telecomunicaciones por el Centro de Investigación Científica y de Educación Superior de Ensenada en el 2015. Actualmente es profesor en el Instituto Tecnológico de Sonora, Unidad Guaymas. Sus intereses de investigación incluyen aprendizaje automático, internet de las cosas y procesamiento de señales.



Cesar Puente obtuvo el grado de Doctor en Ciencias de la Tierra por el Centro de Investigación Científica y de Educación Superior de Ensenada en el 2011. Actualmente es profesor investigador en la Universidad Autónoma de San Luis Potosí. El Dr. Puente participa como revisor de artículos científicos en el área de inteligencia artificial y percepción remota. Sus intereses de investigación incluyen reconocimiento de patrones, visión computacional, cómputo evolutivo y percepción remota.



Juan Villegas Cortez es profesor titular en el Departamento de Sistemas en la UAM Azcapotzalco desde 2010. Obtuvo su grado de Doctor en Ciencias de la Computación en el Centro de Investigación en Computación del Instituto Politécnico Nacional. Sus intereses actuales de investigación incluyen reconocimiento de Patrones, visión por computadora, computación evolutiva, memorias asociativas y el procesamiento digital de señales.

Publique en Komputer Sapiens



ARTÍCULO ACEPTADO

Interfaz gráfica del algoritmo basado en el forrajeo de bacterias para optimización numérica con restricciones

J. Adrian García-López, Betania Hernández-Ocaña, Oscar Chávez-Bosquez y José Hernández-Torruco

Resumen

Existen algoritmos de inteligencia colectiva que emulan el comportamiento del forrajeo de bacterias *E. Coli* de manera computacional, como el TS-MBFOA. Sin embargo, son complejas su adaptación e implementación al resolver un problema de optimización particular para aquellos usuarios que desconocen la codificación del algoritmo. En este artículo es presentado el diseño de una interfaz gráfica para el algoritmo TS-MBFOA, con el objetivo de calibrar los parámetros del algoritmo de manera más rápida y sin la necesidad de manipulación de las líneas de código por el usuario final. Para el diseño de la interfaz se hace uso de diagramas UML. En la interfaz, el usuario podrá configurar las ejecuciones, calibrar los parámetros del algoritmo y visualizar los resultados obtenidos a través de tablas y estadísticas básicas. La interfaz es implementada con éxito en un problema de optimización numérica con restricciones conocido como el problema del resorte de tensión/compresión.

Introducción

Los problemas de la vida real, de cualquier área, que pueden ser modelados matemáticamente tienen solución a través de programación matemática o haciendo uso de alguna Metaheurística. Estos problemas buscan minimizar o maximizar una función de calidad y son conocidos como problemas de optimización y pueden ser modelados como un problema general de programación lineal o no-lineal. Los problemas de optimización, específicamente los no-lineales, son complejos y resulta caótico resolverlos con programación matemática. Actualmente existen las metaheurísticas, algoritmos que permiten resolver problemas de optimización con o sin restricciones, combinatorios o numéricos de manera aproximada, es decir, generan una o más soluciones factibles cercanas al óptimo en tiempos razonables, estos problemas son conocidos como Problemas de Optimización Numérica con Restricciones (PONRs).

Algunas de estas técnicas emulan o simulan procesos naturales o evolutivos y son conocidos como algoritmos bio-inspirados [1], los cuales son parte del cómputo bio-inspirado. Estos algoritmos surgen con la motivación de mejorar máquinas de búsqueda y resolver problemas de optimización a través de la simulación de procesos inteligentes y colaborativos. De acuerdo al fenómeno natu-

ral en que basan su diseño se clasifican en dos grupos: los Algoritmos Evolutivos (AE) cuyo funcionamiento se basa en emular el proceso de evolución natural y la supervivencia del más apto, y los Algoritmos de Inteligencia Colectiva (AIC), los cuales basan su funcionamiento en comportamientos cooperativos de ciertos organismos simples e inteligentes, como las abejas, hormigas, bacterias, aves, monos, lobos, entre otros. Entre los AIC se encuentra el algoritmo basado en el forrajeo de bacterias (BFOA, por su sigla en inglés), este algoritmo ha sido escasamente estudiado y analizado para resolver problemas de optimización, primero, sin restricciones y posteriormente, con restricciones usando diferentes técnicas para su manejo como las reglas de factibilidad, función de penalización, operadores especiales, entre otras. Las ideas iniciales de este algoritmo fueron propuestas de manera completa en [2] donde se propone el Algoritmo de Optimización basado en el Forrajeo de Bacterias BFOA. Su proceso natural se basa en que cada bacteria *E. Coli* trata de maximizar su energía obtenida por unidad de tiempo empleada en el proceso de forrajeo, donde también evade sustancias nocivas. Más aún, las bacterias se pueden comunicar entre sí mediante la segregación de sustancias. En el BFOA se tienen cuatro procesos principales: quimiotaxis (nado-giro), agrupamiento, reproducción y eliminación-dispersión. Las bacterias son soluciones potenciales al problema, y su ubicación representa los valores de las variables de decisión del problema. Las bacterias pueden moverse (generar nuevas soluciones) mediante el ciclo quimiotáctico se genera además un movimiento mediante la atracción que ejercen soluciones en zonas prometedoras a otras soluciones del espacio de búsqueda, se permite la reproducción de las mejores soluciones y finalmente se eliminan del cúmulo aquella o aquellas bacterias localizadas en zonas de baja calidad.

El algoritmo de optimización basado en el forrajeo de bacterias modificado (MBFOA, por su sigla en inglés) es un algoritmo basado en BFOA, las modificaciones realizadas incluyen un mecanismo para el manejo de las restricciones basado en reglas de factibilidad y una disminución de parámetros respecto a los del BFOA original al agrupar en un ciclo generacional los cuatro procesos básicos del algoritmo, los cuales en su versión original eran un ciclo independiente [3].

Finalmente, encontramos la propuesta *Two-Swim Modified Bacterial Foraging Optimization Algorithm* (TS-MBFOA) un mecanismo de sesgo es implementado para generar la población inicial de bacterias en tres grupos esparcidos con una distribución uniforme y aleatoria. Finalmente, el tamaño de paso es calculado generacionalmente dentro del rango de las variables de diseño del problema a resolver y escalado con un parámetro definido por el usuario. En el proceso quimiotáctico se intercalan dos nados, el primero es el nado original de BFOA, a excepción del tamaño de paso (el cual se propone sea aleatorio), y el segundo nado incluye el operador de mutación usado en los algoritmos evolutivos, la inclusión de ambos con el objetivo de mejorar la capacidad de exploración y explotación del algoritmo [4].

El TS-MBFOA es una de las estrategias que permite tener resultados competitivos y favorables al solucionar PONRs. Esta propuesta recientemente se ha utilizado con éxito para resolver diversos problemas [4,5]. Dicho algoritmo posee sus propios parámetros que corresponden a cada uno de sus procesos, los cuales son independientes de las variables del problema a resolver y deben ser calibrados o ajustados de tal forma que permitan un mejor rendimiento del algoritmo, es decir, que el algoritmo encuentre soluciones óptimas globales con un costo computacional moderado, no excesivo, al solucionar el problema.

El tiempo de ejecución está implícito, encontrar la calibración perfecta de los parámetros del algoritmo es una tarea difícil de completar y más aún si el ajuste de parámetros es directo en las líneas de código sin ayuda de una interfaz gráfica. Una interfaz para la calibración de parámetros de algún modo permite que el usuario final realice esta tarea de manera más rápida y con el mínimo esfuerzo, por lo tanto el uso de una Interfaz Gráfica de Usuario (GUI, por su sigla en inglés), en el contexto de la interacción humano-computadora, es como un artefacto interactivo, que por su diseño y a través de ciertas interfaces, posibilita la interacción de una persona con el sistema informático, haciendo uso de las gramáticas visuales y verbales [6]. Es por ello que la implementación del algoritmo en una GUI es de suma importancia para los que tienen la necesidad de utilizar el algoritmo ya que permite estar aislado de conocimientos de programación y así el usuario final no tiene la necesidad de interactuar directamente con las líneas de código permitiendo ahorrar tiempo en la calibración de los parámetros y, por ende, en la ejecución. Además, permite una mejor toma de decisiones y un mejor análisis del rendimiento del algoritmo en PONRs.

En la revisión de la literatura especializada existen propuestas del desarrollo de interfaz para la implementación del BFOA en problemas particulares. De acuerdo a las propuestas revisadas que implementan BFOA con ayuda de una interfaz gráfica, se observa que el software

común de desarrollo es Matlab, los autores hacen uso de la interfaz para ingresar los datos de entrada para el problema a resolver y/o mostrar los resultados en las gráficas que se permite generar en Matlab. En ninguno de los casos revisados se hace uso de la interfaz para ingresar y/o calibrar los parámetros de entrada del algoritmo. Tampoco se visualiza el rendimiento del algoritmo usando alguna métrica de desempeño o gráficas como la de convergencia. Finalmente, en ninguna de las propuestas es posible exportar los datos generados por el algoritmo [7,8].

Algoritmo TS-MBFOA

Este algoritmo intercala dos nados en el proceso quimiotáctico, el primero es el nado original con tamaño de paso aleatorio y el segundo nado incluye el operador de mutación usado en los algoritmos evolutivos para mejorar la capacidad de exploración y explotación del algoritmo. En TS-MBFOA, una bacteria representa una solución potencial para el PONR. Una generación del algoritmo consta de un proceso quimiotáctico, agrupamiento, reproducción y eliminación-dispersión, que a continuación se describen:

Quimiotaxis: Dos nados se intercalan, en cada ciclo solo un nado de explotación o exploración es realizado. El nado de exploración usa la mutación de dos bacterias diferentes seleccionadas aleatoriamente de la población. El nado de explotación se calcula con el operador de giro original de BFOA usando un tamaño de paso aleatorio dentro del rango de cada variable de decisión del problema a resolver.

Agrupamiento: En el ciclo medio del proceso quimiotáctico un agrupamiento de bacterias es realizado con dirección escalada hacia la mejor bacteria de la población en la generación actual del algoritmo.

Reproducción: Se ordenan las bacterias con base a las reglas de factibilidad, se eliminan a las peores bacterias de la población y se duplican a las mejores cada cierto número de ciclos, definido por el usuario final.

Eliminación-dispersión: Se elimina a la peor bacteria de la población, basado en las reglas de factibilidad y se genera una nueva aleatoriamente.

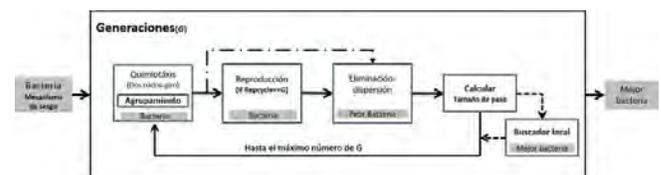


Figura 1. Procesos generales de TS-MBFOA.

La estructura de TS-MBFOA se muestra en la Figura 1, donde los parámetros de entrada son: número de bacterias (S_b), límite del ciclo quimiotáctico (N_c), número de bacterias a reproducir (S_r), factor de escalamiento (β)

para el operador de agrupamiento, tamaño de paso (R) y frecuencia de la reproducción (RepCycle).

Modelado UML de TS-MBFOA

Para el diseño de la interfaz se hace uso de un lenguaje gráfico de propósito general conocido como Lenguaje Unificado de Modelado (UML, por su sigla en inglés), el cual es considerado como una herramienta que permite capturar la idea de un sistema para comunicarla a quien esté involucrado en su proceso de desarrollo; esto se lleva a cabo mediante un conjunto de símbolos y diagramas. Uno de los principales usos del UML es para visualizar, especificar, construir y documentar los componentes de un sistema de software [9]. Para el desarrollo de la interfaz de TS-MBFOA se utilizaron seis diagramas del modelo UML: diagrama de clases (Figura 2), diagrama de casos de uso (Figura 3), diagrama de actividades (Figura 4), diagrama de secuencia (Figura 5), diagrama de tiempo (Figura 6) y diagrama de despliegue (Figura 7).

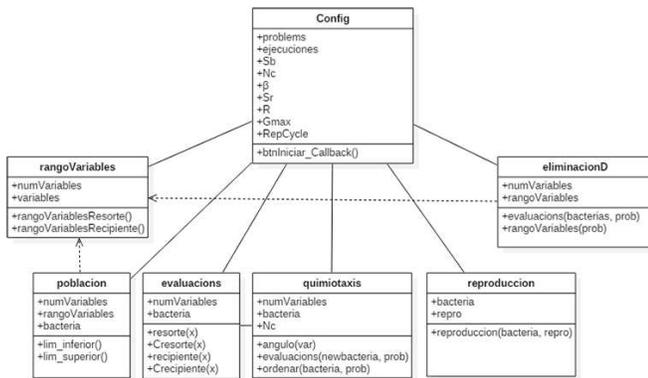


Figura 2. Diagrama de clases.

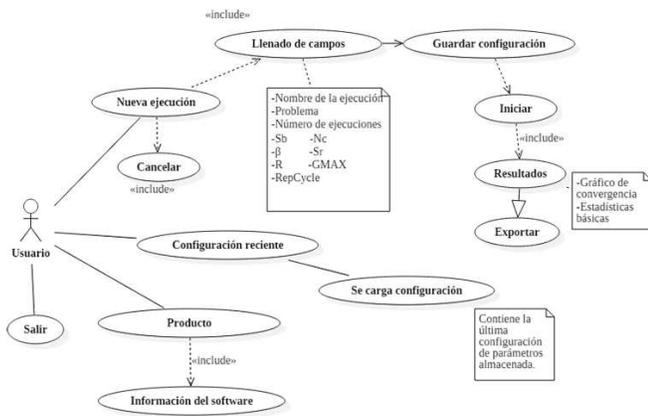


Figura 3. Diagrama de casos de uso.

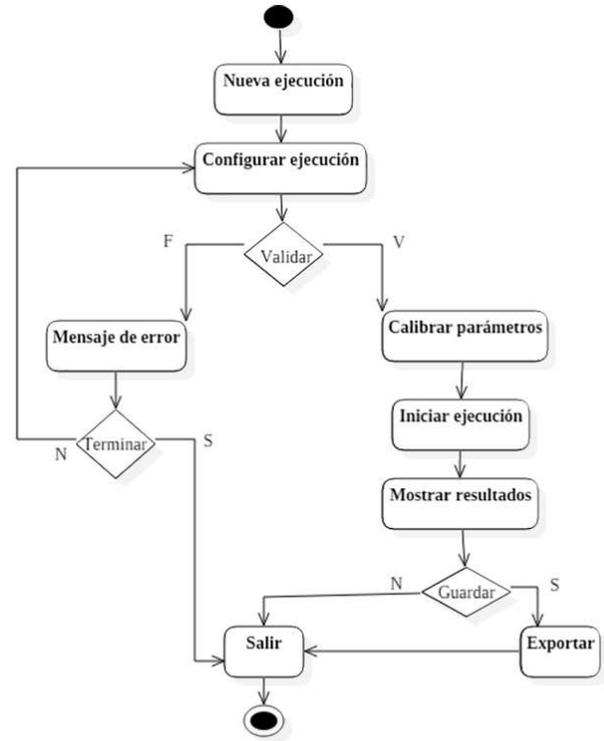


Figura 4. Diagrama de actividades.

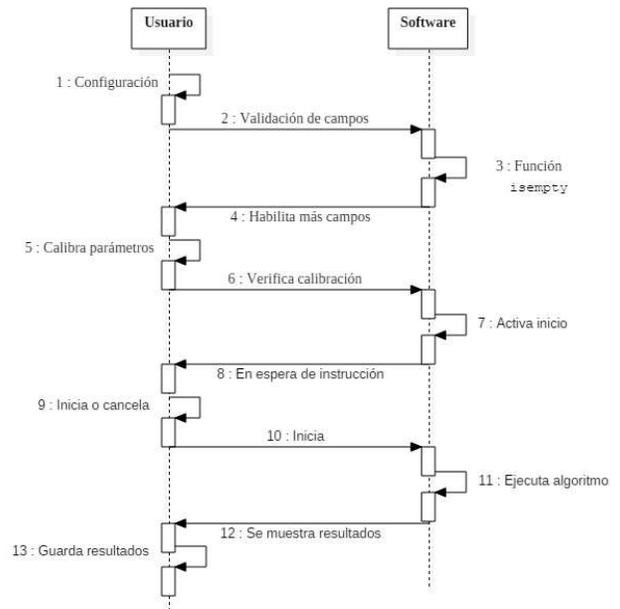


Figura 5. Diagrama de secuencia.

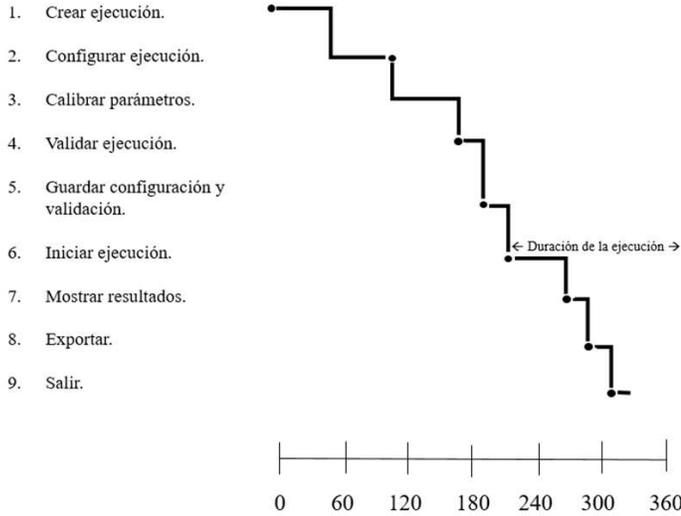


Figura 6. Diagrama de tiempo.



Figura 8. Interfaz de configuración y calibración de TS-MBFOA.

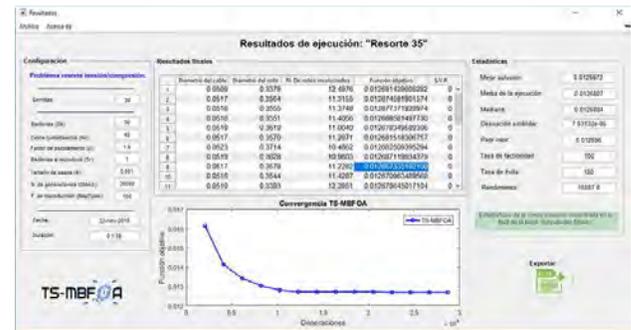


Figura 9. Interfaz de resultados.

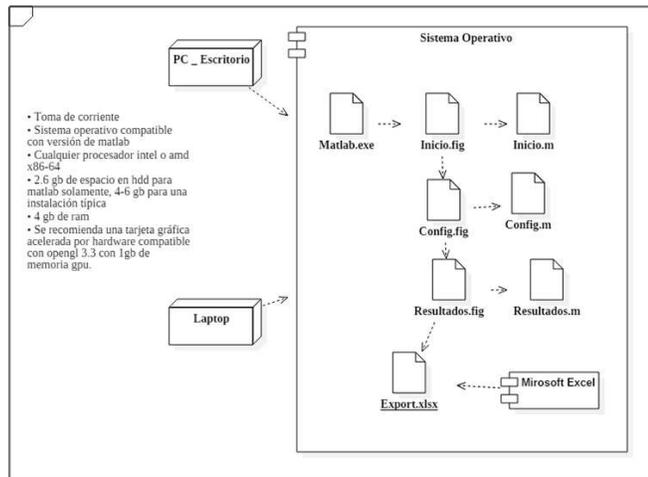


Figura 7. Diagrama de despliegue.

Diseño e implementación de la interfaz gráfica de TS-MBFOA

Las interfaces fueron desarrolladas en Matlab R2018b en una PC con procesador Core 2 Duo 3.5, 4GB de RAM, y 64 bits del Sistema Operativo Windows 7. La primera interfaz permite la configuración de la ejecución y la calibración de los parámetros del algoritmo (ver Figura 8). En la segunda interfaz (ver Figura 9), se presentan los resultados obtenidos.

Para probar la interfaz desarrollada, se hace uso del PONR conocido como el resorte de tensión/compresión; el valor óptimo conocido en el estado del arte de este problema de minimización es 0.012665, de un algoritmo híbrido entre un algoritmo genético y un algoritmo seno coseno [10]. Para la ejecución de TS-MBFOA, la configuración de prueba es llamada Resorte 35 y se seleccionó el problema: 1. Resorte de tensión/compresión. Además, se indicó el número de ejecuciones en 30 debido a que es el número común de ejecuciones independientes en el estado del arte. Los valores asignados para cada parámetro del algoritmo TS-MBFOA fueron: bacterias (S_b) = 50, tamaño de paso (R) = 0.001, factor de escalamiento (β)=1.9, ciclo quimiotáctico (N_c)= 40, bacterias a reproducir (S_r) = 1, frecuencias del ciclo de reproducción (RepCycle) = 100 y número de evaluaciones = 30,000. La interfaz fue configurada y ejecutada con una duración de 1:59 minutos. La mejor solución encontrada y los valores estadísticos básicos de las 30 ejecuciones independientes realizadas fue 0.0126673, la cual es extraída de la tabla Resultados finales de la interfaz resultados (ver Figura 9, “Estadísticas”). Además, la desviación estándar obtiene un valor muy cercano a cero, lo cual indica que todas las soluciones obtenidas en las 30 ejecuciones son similares al óptimo global o mejor solución conocida. Algo que permite confirmar esta observación es el porcentaje de la tasa de factibilidad y tasa de éxito, donde las 30 ejecuciones independientes obtuvieron soluciones factibles e

iguales o mejores que el óptimo global conocido en el estado del arte (100 %). Finalmente, se requieren alrededor de 15587 evaluaciones para obtener una solución factible y óptima para este problema con este algoritmo.

La convergencia de un algoritmo ocurre cuando una solución se conserva sin variaciones hasta el final de una ejecución. Esta solución puede ser: factible o no factible, un óptimo local o global. En la Figura 9 se muestra la convergencia de TS-MBFOA en el problema del resorte con los datos de la ejecución número 15, la cual es la mediana de las 30 ejecuciones. Para este problema de prueba, TS-MBFOA antes de las 10,000 generaciones tiene una convergencia rápida progresiva y continua, después de estas generaciones el algoritmo se mantiene sin variación. La convergencia en una solución de buena calidad para el problema del resorte se obtiene entre la generación 10,000 a la 15,000 aproximadamente con este algoritmo, de acuerdo a la Figura 9. Finalmente, la interfaz del algoritmo TS-MBFOA fue implementada de manera satisfactoria.

Conclusión

En este trabajo se desarrolló una interfaz gráfica para el algoritmo basado en el forrajeo de bacterias llamado TS-MBFOA, siguiendo los diagramas UML diseñados previamente. Dos ventanas fueron diseñadas a la interfaz, en la primera, el usuario final puede configurar la ejecución y calibrar los parámetros del algoritmo. En la segunda pantalla se pueden observar los resultados obtenidos por el algoritmo desplegados en gráfica y tablas, además de poder exportar los resultados a una hoja de cálculo para su posterior uso.

La interfaz gráfica fue probada para solucionar el problema conocido como resorte de tensión/compresión en 30 ejecuciones independientes, la cual funcionó sin ningún inconveniente. Se pudieron observar los resultados en las tablas y gráfica de las interfaces desarrolladas. Dentro de las ventajas o beneficios para el usuario final se encuentran el ahorro de tiempo y esfuerzo para la calibración de parámetros, control y orden de las ejecuciones a realizar, visualización clara y bien distribuida de los resultados, lo cual permite una mejor toma de decisiones para el usuario final.

Como trabajo futuro se pretende migrar a un software *open source*, además de incluir un conjunto de pro-

blemas del *benchmark* y modificar el software para que pueda aceptar problemas definidos por el usuario final. *

REFERENCIAS

1. Eiben, A., y Smith, J.E. (2003). Introduction to Evolutionary Computing. *Natural Computing Series*, Springer-Verlag, 53, 1-18.
 2. Passino, K.M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems Magazine*, 22(3), 52-67.
 3. Mezura-Montes, E., y Hernández-Ocaña, B. (2009). Modified bacterial foraging optimization for engineering design. En *Cihan H. Dagli and et al., editors, Proceedings of the Artificial Neural Networks in Engineering Conference of Intelligent Engineering Systems Through Artificial Neural Networks*, ASME Press, 19, 357-364.
 4. Hernández-Ocaña, B., Pozos-Parra, P., Mezura-Montes, E., Portilla-Flores, E. A., Vega-Alvarado, E., y Calva-Yáñez, M.B. (2016). Two-swim operators in the modified bacterial foraging algorithm for the optimal synthesis of four-bar mechanisms. *Computational Intelligence and Neuroscience*, 2016, 1-17.
 5. Hernández-Ocaña, B., Hernández-Torruco, J., Chávez-Bosquez, O., Calva-Yáñez, M., y Portilla-Flores, E. (2019). Bacterial Foraging-Based Algorithm for Optimizing the Power Generation of an Isolated Microgrid. *Applied Sciences*, 9(6), 1261.
 6. Expósito, C.M. (2006). Interfaz gráfica de usuario: Aproximación semiótica y cognitiva. Diseño gráfico y comunicación visual. Recuperado el 15 de Febrero, de <https://www.clubensayos.com/Temas-Variados/Interfaz-gr%C3%A1fica-de-usuario-Aproximaci%C3%B3n-semi%C3%B3tica/305876.html>.
 7. Hongdan, L., Sheng, L., y Lanyong, Z. (2015). Ship collision avoidance path planning strategy based on quantum bacterial foraging algorithm. En *2nd International Conference on Electrical, Computer Engineering and Electronics*, Jinan, P.R. China, 2015, 612-621.
 8. Kumar, A., Veeranna, V., Durgaprasad, B., y Sarma, B. (2013). A MATLAB GUI tool for optimization of FMS scheduling using conventional and evolutionary approach. *International Journal of Current Engineering and Technology*, 3(5), 1739-1744.
 9. Booch, G., Rumbaugh, J., y Jacobson, I. (2010). El Lenguaje Unificado de Modelado. Guía del usuario. 2ª edición. Madrid, Pearson Educación.
 10. El-Shorbagy, M.A., Farag, M.A., Mousa, A.A., El-Desoky, I.M. (2020). A Hybridization of Sine Cosine Algorithm with Steady State Genetic Algorithm for Engineering Design Problems. En *Hassanien A., Azar A., Gaber T., Bhatnagar R., F. Tolba M. (eds) The International Conference on Advanced Machine Learning Technologies and Applications (AMLTA2019)*, Advances in Intelligent Systems and Computing, Springer, Cham, Cairo, Egipto, 921.
-

SOBRE LOS AUTORES



J. Adrian García-López es un estudiante de la maestría en Ciencias de la Computación en la Universidad Juárez Autónoma de Tabasco. Sus intereses de investigación incluyen optimización global con restricciones usando metaheurísticas y desarrollo de software.



Betania Hernández-Ocaña obtuvo el grado de doctor en Ciencias de la Computación en la Universidad Juárez Autónoma de Tabasco en 2016, de la cual es profesora investigadora. Actualmente es miembro del Sistema Nacional de Investigadores Nivel I. Sus intereses científicos incluyen algoritmos de inteligencia colectiva y evolutivos, optimización global y manejo de restricciones.



Oscar Chávez-Bosquez obtuvo el grado de doctor en Ciencias de la Computación en la Universidad Juárez Autónoma de Tabasco en 2016, de la cual es profesor investigador. Actualmente es miembro del Sistema Nacional de Investigadores Nivel I. Sus intereses científicos incluyen desarrollo de software con código libre y la solución de problemas de la vida real usando inteligencia artificial.



José Hernández-Torruco obtuvo el grado de doctor en Ciencias de la Computación en la Universidad Juárez Autónoma de Tabasco en 2016, de la cual es profesor investigador. Actualmente es miembro del Sistema Nacional de Investigadores Nivel Candidato. Sus intereses científicos incluyen aprendizaje automático y minería de datos, especialmente su aplicación en Medicina.

Los algoritmos evolutivos son los algoritmos que se basan en la evolución de las especies; en general se basan en la teoría evolutiva de Charles Darwin. Se caracterizan por la existencia de una población de individuos expuestos a la presión ambiental, lo que conduce a la selección natural, es decir, a la supervivencia del más apto y, a su vez, al aumento de la aptitud media de la población. La aptitud es la medida del grado de adaptación de un organismo a su entorno; cuanto mayor es la aptitud, más apto y adaptado está el organismo al medio ambiente. En general, los algoritmos evolutivos se centran sólo en un subconjunto de mecanismos definidos sobre el proceso evolutivo biológico.

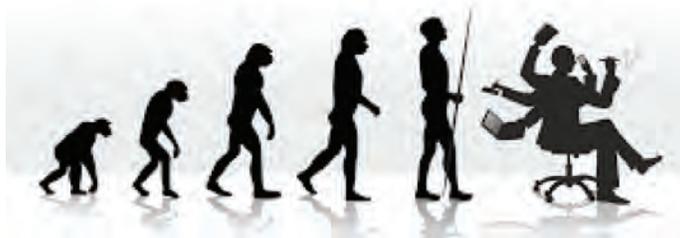


Imagen tomada de: <https://esa.github.io>.

ARTÍCULO ACEPTADO

El uso malicioso de la Inteligencia Artificial en Twitter

Israel Morales Trigueros y Carlos Adolfo Piña García

La Tuitósfera en México

Twitter se ha convertido en un canal de comunicación que diariamente genera diversas tendencias derivadas de eventos, hechos y noticias que ocurren en el territorio nacional. Dichas tendencias, etiquetadas con un ‘#’ (hashtag), promueven o atraen la atención de los usuarios para ser incluidos en la conversación o en una discusión polarizada.

La polarización se ha vuelto un elemento fundamental en todas las tendencias que emergen en Twitter; generando temas sesgados que probablemente obedecen a intereses particulares. El objetivo principal de este artículo es explorar y demostrar el uso de campañas coordinadas por automatismos en México. En este sentido, planteamos las siguientes preguntas de investigación que nos permitirán conocer el uso malicioso en torno a la conversación en Twitter.

1. ¿Qué tipo de cuentas ensucian o polarizan la conversación en México?
2. ¿Qué mecanismos o estrategias utilizan dichas cuentas para controlar el flujo de información?

Agentes computacionales o bots en Twitter

Ataques coordinados, acoso selectivo, repuestas automáticas, son algunas de las técnicas utilizadas por las granjas de automatismos o cibertropas intentando disciplinar, silenciar e intimidar para imponer una conversación. Este tipo de estrategia tiene sus bases en la teoría tradicional de los sistemas multi agente, donde se percibe a un agente como un sistema computacional capaz de actuar de forma semiautónoma para poder cumplir con su tarea o meta asignada.

Estos agentes o programas computacionales se han ido adaptando a la evolución digital en las redes sociales. Hoy los podemos encontrar como *chatbots* auxiliares en muchas páginas web, cuya tarea es interactuar respondiendo preguntas frecuentes a los usuarios. Este mismo principio es utilizado de forma maliciosa en Twitter donde los agentes, en este caso bots, presentan un comportamiento automatizado que les permite reaccionar ante alguna palabra o frase. La reactividad del agente se basa en términos de *likes*, retuits o ataque (*spam*). La inteligencia artificial detrás de estos bots está basada en la

definición de lo que es un agente inteligente, es decir, dicho bot está situado en un entorno, por ejemplo Twitter, que posee habilidades como la percepción y la recolección de información (tuits). A partir de lo anterior, el agente es capaz de ejercer alguna acción reactiva (como respuesta automática) ante un estímulo digital.

En el año 2016 se llevó a cabo un estudio donde se recolectó información acerca de la conversación en Twitter relativa al Brexit. Resultó que el 1 % de las cuentas eran automatizadas, pero este pequeño grupo fue el responsable de generar al menos el 32 % de todos los tuits[1].

Básicamente, al usar cuentas automatizadas lo que se busca es crear un efecto de *cibercascada* donde un grupo de cuentas detonan una tendencia artificialmente coordinada a través de tuits, los cuales carecen de sentido. Dichos tuits funcionan en sincronía con el algoritmo de Twitter para inflar alguna tendencia. Así, una vez que ésta ha sido posicionada funciona como un atractor de cuentas no artificiales u orgánicas, las cuales participan en la conversación y terminan por consolidar el hashtag. La misión de los bots es sostener la tendencia en los primeros lugares.

Entre las comunidades de bots es muy importante el entorno en los sistemas multi agente. Lo anterior se debe a que representa una cámara de eco que solamente apoya ideas o intereses comunes. Las cámaras de eco permiten que el usuario común consuma contenido que reafirme sus creencias y opiniones, llevándolo a una radicalización o polarización.

Históricamente, el uso más evidente de bots fue en el año 2015 donde se presenció el uso del hashtag: #Ya-MeCanse. Sirviendo como reclamo a la ineficiencia del gobierno del expresidente Enrique Peña Nieto, el hashtag fue tan popular que tuvo un volumen aproximado de 4.4 millones de tuits. En ese caso se utilizó una estrategia disruptiva para mitigar o ensuciar la conversación en contra del gobierno de México. Esta situación dio como resultado uno de los términos coloquiales más populares de ese sexenio, los denominados *peñabots*.

Taxonomía de bots

Hasta ahora se ha dado una idea general de qué son los agentes computacionales o bots. Sin embargo, es necesario clarificar este tipo de automatismos. Chu et al. [2]

propusieron tres tipos de usuarios generales para Twitter:

- **Bot:** es un programa computacional autónomo que es desplegado en Twitter y que tiene capacidad de interactuar con otros usuarios dentro de esta red social.
- **Ciborg:** es una cuenta híbrida, la cual es manejada por un usuario humano y un bot. Dicha cuenta presenta algún grado de automatismo en ciertos periodos de tiempo.
- **Humano:** aquella cuenta que presenta un nivel bajo de automatización, y por lo tanto se le considera una cuenta promedio u orgánica.

Por otro lado, en [3] se propuso la siguiente clasificación:

- **Bot transmisor:** diseminan información a una audiencia general, compartiendo URLs benignos a sitios de noticias, blogs, entre otros. Suelen ser administrados por organizaciones o un grupo de personas.
- **Bot consumidor:** agregan contenido desde diversas fuentes y/o proveen servicios de actualización como reportes de clima, tráfico, etc.
- **Bot de spam:** postean contenido malicioso engañando a otros usuarios pretendiendo ser otra persona o redireccionándolos a sitios no seguros. Además, promueven contenido incierto, inválido o irrelevante (desinformación).

La detección de bots

Cuando los bots comenzaron a ser un problema real en la influencia de elecciones a nivel mundial, el Observatorio de Redes Sociales de la Universidad de Indiana desarrolló la herramienta *Botometer* (ver [https://botometer.iuni.iu.edu/#/](https://botometer.iuni.iu.edu/#!/) para más información). *Botometer* detecta automatismos y detecta mediante el uso de algoritmos de aprendizaje automático que utilizan variables asociadas a: tiempo de actividad, tasa de retuits, red de amistad, entre otras. Estas variables se utilizan para generar un sistema de puntos que va de 0 hasta 5, y que permite determinar si el usuario tiene algún grado de automatización. En la Figura 1 se muestra un ejemplo de bot con un valor de 4.6/5, a través de la interfaz web de *Botometer*. El tacómetro en la esquina superior derecha indica el *bot-score* de la cuenta analizada. Luego existen tres apartados, el primero para características de cuentas de habla inglesa (no se toma en cuenta para este estudio, el segundo las 4 variables determinantes del *bot-score* y finalmente el puntaje en forma de número decimal así como en porcentaje de probabilidad.



Figura 1. Interfaz de Botometer y los puntajes de sus respectivas variables.

Botometer se ha convertido en un elemento interesante en términos de la detección temprana de automatismos. No obstante, se debe ser cauteloso con su uso ya que existe la posibilidad de encontrar falsos positivos, es decir, cuentas que son clasificadas erróneamente como bots a pesar de ser controladas por un usuario humano. Es importante señalar que el campo disciplinar de la detección de automatismo en Twitter se encuentra en constante crecimiento puesto que la programación de bots o el uso de automatismos cada día es más sofisticada.

Caso de estudio: #PrensaProstituida

Examinaremos a continuación un caso de estudio para evidenciar si existió la presencia de bots, así como para detectar coordinación entre ellos para inflar la tendencia. Una vez obtenidos los resultados, fue posible concluir que, aunque en una proporción muy pequeña, sí existió presencia de bots en la conversación de este hashtag.

El pasado 4 de noviembre del 2019, la Unidad de Información, Infraestructura Informática y Vinculación Tecnológica presentó por medio de *La Mañanera* (conferencias de prensa realizadas por el presidente todas las mañanas) un informe sobre la conversación que sucedió en Twitter con los hashtags #PrensaSicaria, #PrensaProstituida y #PrensaCorrupta. Estos fueron principalmente usados en México del 31 de octubre al 3 de noviembre del 2019. Para este caso, hemos decidido descartar los hashtags #PrensaSicaria y #PrensaCorrupta pues en esos días también fueron populares en otras partes de Latinoamérica.

La metodología de exploración se basa principalmente en la descarga masiva de tuits para obtener información que nos ayude a encontrar evidencias de automatización y coordinación en las conversaciones. El primer paso consistió en recolectar aquellos tuits relativos a #PrensaProstituida; esta recolección se hizo a través de la API (interfaz de programación de aplicaciones) de Twitter. En total, se obtuvieron 104 mil 972 tuits. La proporción de los tuits se muestra en la Figura 2, donde cada cuadro es equivalente a 100 unidades.



Figura 2. Proporción de tuits para #PrensaProstituida.

Una forma de percibir las opiniones en Twitter es a través de la carga emotiva de cada tuit. Por lo tanto, se llevó a cabo un análisis de sentimientos, el cual fue realizado por medio de la API de procesamiento de lenguaje natural MeaningCloud. La Figura 3 muestra la polarización en las cargas sentimentales; la carga en los tuits es principalmente negativa y es posible argumentar que las palabras utilizadas están relacionadas con el odio, repulsión, enojo, desaprobación, entre otras.



Figura 3. Análisis de sentimientos de los tuits de #PrensaProstituida.

Ahora, examinaremos la influencia de los agentes inteligentes en esta tendencia utilizando *Botometer*. Debido a las limitaciones de la API, se tomó una muestra de 3000 usuarios para el análisis. La Figura 4 muestra la proporción entre bots y humanos. Dicha proporción nos indica que existió participación de cuentas automatizadas en la conversación.

En este sentido, es importante identificar aquellas cuentas que ensuciaron o amplificaron la tendencia. La Figura 5 muestra el top 50 de las cuentas que tuvieron mayor puntaje dentro de la muestra. Se debe mencionar que a pesar de sólo haber detectado 3 cuentas bots, las

cuentas que les preceden tienen un valor que podría hacerlas cuentas anómalas. Estas cuentas son manejadas por humanos que pueden mostrar un comportamiento automatizado y persistente sin ser en un bot.

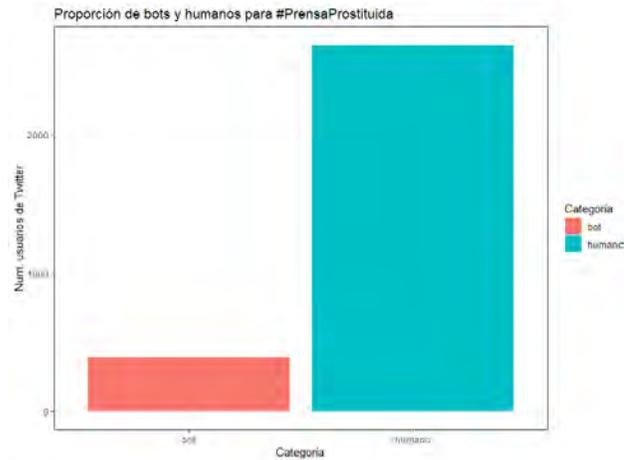


Figura 4. Proporción de usuarios orgánicos y automatismos.

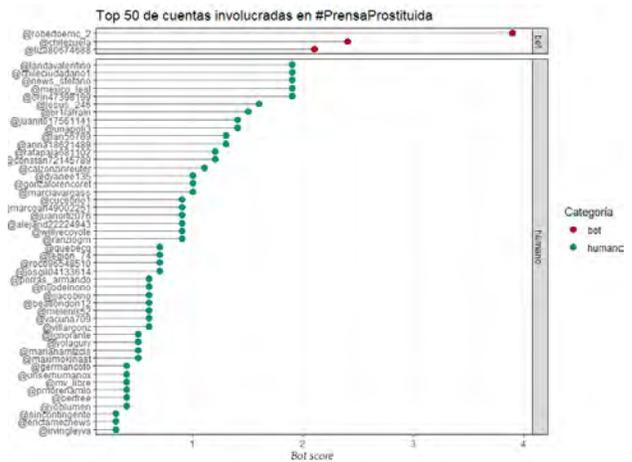


Figura 5. Proporción de usuarios orgánicos y automatismos.

Por otro lado, fue posible identificar variables como: *temporal* y *user*; las cuales están relacionadas con el comportamiento de cada cuenta. En este caso, la variable *temporal* expone información de qué tan activa es una cuenta con respecto a periodos de tiempo. La variable *user* precisa información relativa a los metadatos del usuario. Se calculó el coeficiente de correlación de Pearson para medir el grado de relación entre dichas variables. La Fig. 6 muestra el coeficiente de correlación obtenido; el gradiente hacia la derecha en el eje temporal indica una actividad inusual, y por lo tanto es evidencia de posible actividad automatizada en la tendencia #PrensaProstituida.

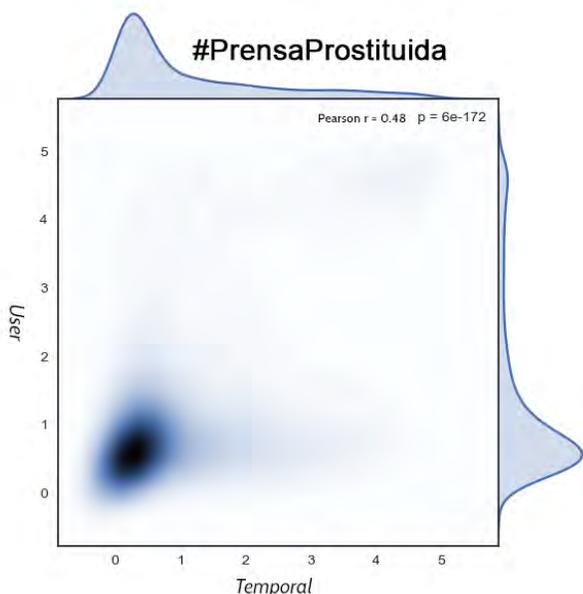


Figura 6. Coeficiente de Pearson para las variables temporal (eje x) y *user* (eje y).

Una de las variables más significativas de *Botometer* es el puntaje bot (*bot-score*), esta variable es utilizada como un umbral donde si *bot-score* es mayor o igual que dos, entonces se considera una actividad automatizada. La Figura 7 ilustra una gráfica de dispersión de automatismos donde el eje x representa el número de usuarios y el eje y el bot-score obtenido. Es posible apreciar la densidad de dispersión determinada por el puntaje de las cuentas de tipo bot.

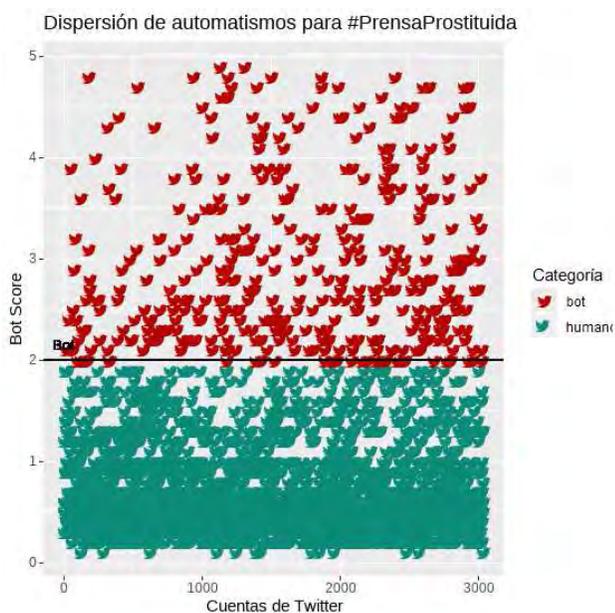


Figura 7. Dispersión de automatismos.

El umbral para la clasificación de los automatismos es resultado de la experiencia adquirida en investigaciones previas relacionadas a este tipo de fenómenos digitales; nos hemos basado en las recomendaciones de otros investigadores de este mismo campo, donde se indica ajustar el umbral de clasificación según las necesidades del estudio [4]. La validación de estas cuentas se realiza de manera manual, en donde se identifican aquellas cualidades que caracterizan a los bots [5].

Uno de los aspectos más importantes en las redes sociales es el análisis desde la teoría de grafos, donde cada nodo (ilustrados como círculos en la gráfica) representa un usuario y cada arco (ilustrados como líneas curvas) representa la correspondencia entre los nodos. Esta estructura nos permite conocer la relación entre los usuarios y las comunidades o grupos más cercanos. En la Figura 8 se presenta el grafo de relación entre los siguientes tipos de nodos: orgánicos (azul), automatizados o bots (rojo) y *hashtags* (verde).

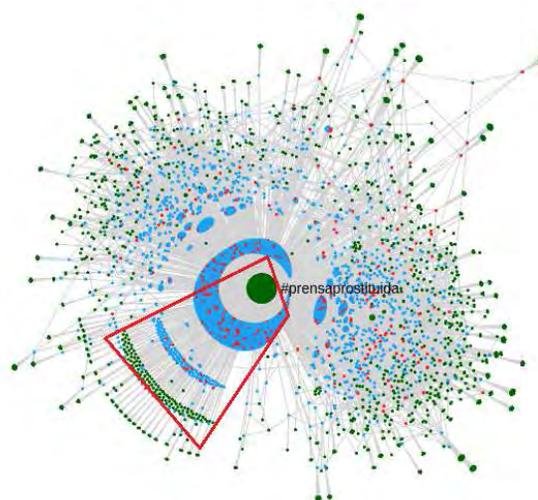


Figura 8. Grafo de relación entre nodos (usuarios) participantes en #PrensaProstituida.

En la figura antes mencionada es posible apreciar un comportamiento coordinado debido a la proximidad de diversos cúmulos verdes en el ángulo inferior izquierdo respecto al nodo central (#PrensaProstituida). Ese cuadrante inferior izquierdo es una representación gráfica de las cámaras de eco.

Otra forma de analizar la red es mediante un grafo de interacción de tipo usuario-hashtag. En la Figura 9 podemos encontrar comunidades más grandes dentro de la conversación, en este caso, sin distinguir lo automatizado de lo orgánico. La comunidad más grande se puede apreciar aglomerada alrededor de #PrensaProstituida (morado). Además, podemos observar una segunda comunidad compartida por #PrensaCorrupta y #PrensaSicaria (verde).

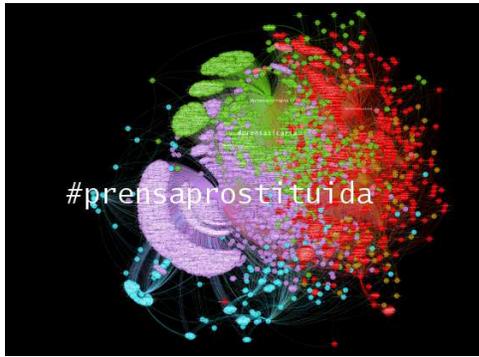


Figura 9. Grafo interacciones usuario-*hashtags* de #Prensa-Prostituida.

Conclusiones

En este artículo hemos hecho un análisis acerca del uso malicioso de la inteligencia artificial a través de cuentas automatizadas en Twitter. En general, encontramos que el tipo de cuentas que polarizan o ensucian la conversación son de tipo bot. Estas cuentas son controladas de forma grupal con la finalidad de crear una cámara de eco la cual amplifica la señal en Twitter, aunque sea una señal falsa. Además, las cuentas anómalas limitan y sesgan nuestros puntos de vista o alteran nuestra percepción acerca de un tema en particular.

Por otro lado, el análisis de sentimientos nos indicó la existencia de una polarización entre lo negativo y lo positivo; este tipo de cuentas anómalas construyen una comunicación emocional poco inteligente y de malos modales, de ahí la suspensión de muchas de las cuentas. Estar expuestos a estas cámaras de eco incrementa la probabilidad de encontrarse con contenido negativo, malicioso o tendencioso [6].

En resumen, las estrategias utilizadas para controlar

la conversación en Twitter se basan en la coordinación de automatismos, cuya misión es amplificar una señal aunque sea una señal falsa.

Las comunidades de automatismos suelen estar compuestas de usuarios que funcionan como concentradores o *hubs*. Los concentradores tienen mayor influencia sobre otros usuarios, y son los encargados de generar el contenido. El resto de ellos amplifican la tendencia usando retuits.

Finalmente, el contenido que nos presenta la red social Twitter representa un beneficio propagandista para aquellos que logran controlar el flujo de información mediante el uso de automatismos. *

REFERENCIAS

1. Suarez-Serrato, P., Roberts, M.E, Davis, C. y Menczer, F. (2016). On The Influence of Social Bots in Online Protests: Preliminary findings on a mexican study. En *Spiro E., Ahn YY. (Eds) Social Informatics. Lectures Notes In Computer Science, Springer*. 10047-1, 269-278.
2. Chu, Z., Gianvencchio, S., Wang, H. y Jajodia, S. (2012). Detecting Automation of Twitter Accounts: Are You a Human, Bot or Cyborg?. *IEEE Transactions on Dependable Secure Computing* 9(6), 811-824.
3. Oentaryo, R.J., Murdopo, A., Praseyto, P.K. y Lim, E. (2016). "On profiling bots in social media". En *Spiro E., Ahn YY. (Eds) Social Informatics. Lecture Notes In Computer Science*, 10046, 92-109.
4. Gramlinch, J. (2018). "Q&A: How Pew Research Center Identified Bots on Twitter". Recuperado el 14 de mayo de 2020, de <https://www.pewresearch.org/fact-tank/2018/04/19/qa-how-pew-research-center-identified-bots-on-twitter/>.
5. Symantec (2018). "How To Spot a Twitter Bot". Recuperado el 14 de mayo de 2020, de <https://symantec-enterprise-blogs.security.com/blogs/election-security/spot-twitter-bot>.
6. Stella, M., Ferrara, E. y De Domenico, M. (2018). Bots increase exposure to negative and inflammatory content in online social systems. En *Proceedings of the National Academy of Sciences*, 115(49), 12435-12440.

SOBRE LOS AUTORES



Israel Morales Trigueros es actualmente colaborador del Laboratorio para el Análisis de Información Generada a través de Redes Sociales en Internet (LARSI) de la Universidad Veracruzana. Obtuvo su Licenciatura en Sistemas Computacionales Administrativos en la Universidad Veracruzana. Su área de investigación es la Ciencia de Datos y el análisis de información en redes sociales.



Carlos Adolfo Piña García actualmente es coordinador del Laboratorio para el Análisis de Información Generada a través de Redes Sociales en Internet (LARSI) de la Universidad Veracruzana. Tiene una estancia postdoctoral en el Depto. de Ciencias de la Computación del Instituto de Investigaciones en Matemáticas Aplicadas y de Sistemas, de la UNAM. Obtuvo su doctorado en la Escuela de Ciencias de la Computación e Ingeniería Electrónica en la Universidad de Essex, Inglaterra. Obtuvo su grado de maestría en Inteligencia Artificial en la Universidad Veracruzana. Su área de investigación es la Ciencia de Datos y la Ciencia Social Computacional.

ARTÍCULO ACEPTADO

Algoritmos inspirados en la naturaleza para control de interferencia en una red de comunicaciones móviles celular

Jesús Alberto Gómez Avilés, Anabel Martínez Vargas y Ángel G. Andrade

Introducción

Un sistema celular se compone principalmente de estaciones base (antenas) y usuarios móviles (dispositivos móviles o teléfonos inteligentes). El medio de transmisión que se utiliza comúnmente es el aire a través de canales de radio frecuencia como se muestra en la Figura 1. El funcionamiento se asemeja al proceso de sintonizar una estación de radiodifusión comercial, ya sea de Amplitud Modulada (AM) o de Frecuencia Modulada (FM), en un radio receptor, y en el que cada comunicación entre usuarios se realiza en una banda de frecuencia distinta.

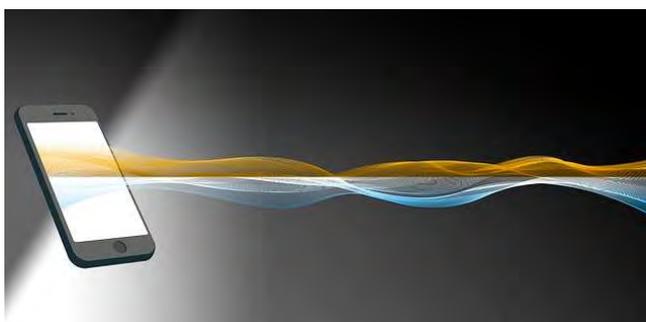


Figura 1. El aire es el medio de transmisión entre un usuario móvil y la estación base. Imagen de Nawhsad Jamil en Pixabay.

En un sistema de comunicación móvil celular, un usuario móvil hace una petición de llamada a su estación base y ésta le asigna una banda de frecuencia por medio de la cual intercambian información. Imagina unas vías de ferrocarril por las que circulan trenes, ¿no puede haber muchos trenes circulando al mismo tiempo!

En 1981 se lanzó comercialmente la primera generación (1G) de tecnología móvil celular que sólo ofrecía servicios de voz. Diez años después, la segunda generación (2G), además de servicios de voz, incluía servicio de mensajería. En este punto, la demanda de servicios y el incremento de usuarios móviles se volvió un problema por la poca cantidad de canales asignados a este servicio. Los organismos internacionales de telecomunicaciones asignaron más bandas de frecuencia al servicio de comunicaciones móviles terrestres, es decir, agregaron más vías para que más ferrocarriles pudieran circular al mismo tiempo [1].

A finales de los 90s, apareció la generación 2.5 (2.5G), que se caracterizó por el envío de información utilizando el protocolo de internet (información por paquetes). Este cambio fue análogo al tener asfalto en lugar de vías y en lugar de circular ferrocarriles enormes, circulaban muchos automóviles de diversos tamaños. Esto permitió que muchos más usuarios pudieran hacer uso de las líneas de comunicación simultáneamente [1].

Diez años después, los sistemas móviles de tercera generación (3G) transformaron el sistema celular de un sistema sólo de telefonía a una red de transporte de contenido multimedia con un gran impacto en las actividades de vida diaria. Siguiendo con la analogía de las carreteras y los automóviles, básicamente, es agregarle más carriles a la carretera. En esta generación surgieron nuevas tecnologías inalámbricas, logrando un incremento en las velocidades de transferencia [1]. Posteriormente en 2011, con el lanzamiento de los sistemas móviles de cuarta generación (4G), se tienen no sólo una carretera de cuatro carriles, sino ¡una autopista de alta velocidad con seis carriles!

Para el 2018, se desplegaron las primeras redes de quinta generación (5G), su objetivo principal fue atender, además de los usuarios, la interconexión de las cosas (conocido como el internet de las cosas). Por lo anterior, la velocidad de los datos debía ser 100 veces mayor a la de los sistemas 4G, el tiempo de respuesta 10 veces menor, y la transmisión de la información tendría que realizarse por cualquier medio disponible. Eso significaba que, habría posibilidad de utilizar un avión, un ferry o incluso una nave espacial para llegar a lugares, que, con medios de transporte convencionales, hubiera sido imposible.

Sin embargo, incremento en la velocidad de datos y en la cantidad de canales de comunicación pareciera no ser suficiente. Los usuarios móviles actualmente consumen más datos, ya que cuentan con una gran variedad de aplicaciones (apps) en sus teléfonos inteligentes (ver Figura 2), para reproducir video y música, acceso a redes sociales o para realizar pagos y trámites diversos; incluso aplicaciones de entretenimiento o para jugar remotamente con otros usuarios.

Aunado a lo anterior, cada vez hay más usuarios móviles con teléfonos inteligentes con mayor capacidad en memoria, resolución de pantalla, o en los procesadores. El consumo de más datos equivale a ser más exigentes

con la información que se envía o recibe, esto es, transferencia libre de errores o con una baja probabilidad de que ocurran. Entre más usuarios móviles se encuentren activos en una zona de servicio celular, mayor probabilidad existe de que se genere interferencia entre ellos. Si volvemos a la analogía de las carreteras podríamos decir que, a mayor tráfico, ¡mayor probabilidad de que los usuarios no lleguen a tiempo a su destino, o incluso que no lleguen! La interferencia es el factor principal que limita la capacidad y calidad en las redes inalámbricas emergentes [2].



Figura 2. El crecimiento de apps ha detonado la proliferación de teléfonos inteligentes. Imagen de Jan Vašek en Pixabay.

En este artículo, presentamos un ejercicio lúdico en el que aplicamos algunos algoritmos inspirados en la naturaleza para determinar la máxima relación señal-interferencia (por su sigla en inglés, C/I) en una red móvil celular.

El objetivo en este estudio de caso es que la estación base asigne los canales de comunicación a los usuarios móviles de tal manera que se logre un despliegue de la red con la mejor calidad de servicio (máximo C/I). El reto en este problema es que la estación base sólo cuenta con un canal de comunicación. Entonces, dicho canal debe asignarse a la mayor cantidad de usuarios móviles posible (lo que en la literatura se conoce como compartición de espectro) [3]. Entre mayor sea la interferencia en la red menor será su relación C/I. Los algoritmos que usamos para resolver el problema de asignación de canal y control de interferencia son: genético, cúmulo de partículas y abejas. El primero imita la evolución de especies. El segundo simula los movimientos coordinados de aves. El tercero imita la búsqueda que hacen las abejas por campos de flores para recolectar su néctar. Para mostrar la efectividad de los algoritmos inspirados en la naturaleza para asignar canales se comparan sus resultados con los obtenidos por simulación Monte Carlo.

Las fuentes de inspiración

Las ciencias computacionales estudian dos tipos de problemas, aquellos que las computadoras actuales pueden resolver en un tiempo razonable y aquellos para los que les tomaría años hallar una solución exacta (la mejor solución de todas las soluciones posibles) a un problema. La manera como las ciencias computacionales han solventado esto último, ha sido creando procedimientos o “recetas” que ofrecen una solución aproximada a la solución exacta del problema, tomándole a las computadoras menos tiempo (minutos) calcularla. Los procedimientos que sigue una computadora basados en comportamientos naturales para hallar en minutos una solución factible al problema se llaman algoritmos inspirados en la naturaleza (o algoritmos bio-inspirados). En los siguientes párrafos explicaremos los algoritmos bio-inspirados usados en el presente artículo.

Tal y como si se tratara de poblaciones de alguna especie, los algoritmos genéticos reproducen padres (dos soluciones candidatas) que heredan a sus hijos (dos nuevas soluciones candidatas) su material genético. Para la reproducción, entre los padres hay una competencia con base en su fuerza (aptitud). Cada nueva población de hijos (soluciones candidatas nuevas) sustituye a la población de padres (soluciones candidatas viejas) tal y como lo hacen los salmones (ver Figura 3). La solución al problema es la mejor solución a lo largo de las generaciones (iteraciones) [4].



Figura 3. Un algoritmo genético basa su funcionamiento en la sobrevivencia del más apto. Imagen de Cock-Robin en Pixabay.

El algoritmo por cúmulo de partículas [5] basa su funcionamiento en los movimientos coordinados de las parvadas (conjunto de soluciones candidatas) en su búsqueda por la mayor concentración de árboles para dormir (función objetivo). Si un ave (solución candidata) encuentra una mayor concentración de árboles a la que previamente había explorado, la guarda en su memoria

(experiencia del ave). La mayor concentración de árboles (mejor solución candidata) encontrada por la bandada se comparte con todas las aves (experiencia de la bandada). Entonces, cada ave modifica su velocidad con base en su propia experiencia y la experiencia del cúmulo. Enseguida actualiza su posición. La solución al problema es el ave que haya logrado llegar a la mayor concentración de árboles (ver Figura 4).



Figura 4. Un algoritmo por cúmulo de partículas basa su funcionamiento en los movimientos coordinados de las bandadas. Imagen de sipa en Pixabay.



Figura 5. Un algoritmo de abejas basa su funcionamiento en la búsqueda que hacen las abejas por los campos con la mayor concentración de flores. Imagen de Thomas Schiewer en Pixabay.

El algoritmo de las abejas [6] basa su funcionamiento en la inteligencia colectiva que emerge de los enjambres de abejas melíferas o abejas de miel. Hay tres tipos de abejas encargadas de encontrar fuentes de comida (soluciones candidatas): exploradoras, trabajadoras y observadoras. Las exploradoras vuelan sin guía, alguna de ellas en busca de zonas con alto potencial de encontrar

miel (soluciones potenciales). Las trabajadoras recorren las zonas marcadas por las exploradoras haciendo un doble papel: terminar de explorar la zona que marcó la abeja exploradora y explotar la zona (solución a optimizar). Si las trabajadoras encuentran una ubicación con suficiente miel (solución con una buena calidad), regresan al panal y vuelan en círculos convenciendo a las observadoras a unírseles y continuar explotando la zona con buena cantidad de miel. Si las observadoras determinan que la fuente de comida es buena, vuelan a la zona de recolección hasta terminar de explotar la solución candidata. Este proceso se repite por un determinado tiempo (número de iteraciones) o hasta encontrar la mejor solución al problema.

Fundamentos para calcular el factor C/I

La interferencia es la perturbación de las señales de radio causadas por señales no deseadas provenientes de otras fuentes, como equipos eléctricos sin blindaje u otras estaciones transmisoras de ondas de radio. Dicha perturbación resulta en una nueva onda que puede ser de mayor, menor o igual amplitud a la de las dos ondas que interfieren.

Para explicar cómo se determina la relación señal-interferencia (factor C/I), utilizaremos como ejemplo el escenario ilustrado en la Figura 6. En la imagen se puede observar al centro una estación base (BS) que recibe la señal de cada uno de los usuarios de la red móvil (identificados como C, e I_1 hasta I_5). En este caso, consideraremos a C como el usuario móvil que se está comunicando en este instante con la BS. Si suponemos que los seis usuarios utilizan simultáneamente el mismo canal de comunicación con la BS, entonces los usuarios $I_1 - I_5$ serán los usuarios que interfieren la comunicación del usuario C con la BS en ese mismo instante. La relación C/I es el cociente entre el nivel de potencia de la señal del usuario C y la suma de la potencia de cada señal correspondiente a los usuarios I_1 a I_5 respectivamente, llamada *interferencia total*.

El nivel de potencia de cada señal que recibe la BS se atenúa debido a la distancia que recorre cada señal (en la Figura 6 representada por d_c y $d_1 - d_5$) entre el usuario y la BS, y por la obstrucción de dicha señal debido a objetos grandes, como árboles o edificios, efecto conocido como *sombreo* de la señal. Por lo que el nivel de potencia de cada señal será distinto entre ellas.

Es un hecho que en cualquier sistema inalámbrico es imposible eliminar por completo la interferencia, pero estos sistemas se diseñan para ser tolerables a cierto nivel de interferencia. La relación C/I nos indica ese nivel de tolerancia y es utilizada por operadores de telefonía celular para garantizar calidad en los servicios de voz, datos y video que utilizamos diariamente. Por ejemplo, si la BS calcula una relación C/I mayor a un mínimo nivel de interferencia tolerable por la red, significa entonces que

existirá una comunicación exitosa entre el usuario C y la BS. Por el contrario, si la relación C/I es menor, entonces significa que el nivel de interferencia es mayor que la potencia de la señal del usuario C . En este caso, la comunicación entre C y la BS será ininteligible, esto significa que no habrá forma de que C comunique su información a la BS.

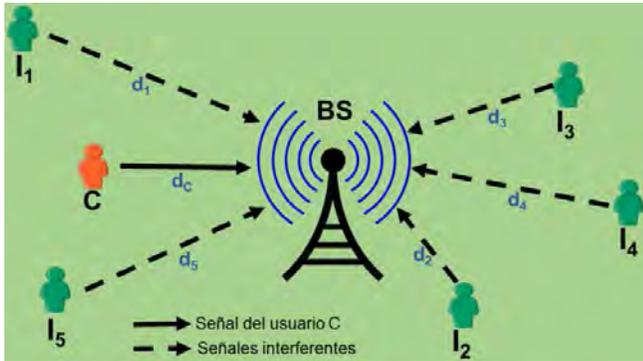


Figura 6. Escenario de simulación.

Aplicación de algoritmos bio-inspirados para maximizar la relación C/I

Para encontrar la mejor relación C/I en el escenario de la Figura 7 se aplicaron los algoritmos genético generacional con selección por torneo (AGT), binario por cúmulo de partículas (BPSO), y disimilar de colonia de abejas artificial (DisABC). Para fines de comparación se incluyó la evaluación por simulación Monte Carlo. La simulación Monte Carlo es un método de simulación clásico que, por medio de pruebas aleatorias repetidas, proporciona soluciones aproximadas a problemas que no se pueden evaluar con técnicas exactas. Esta estrategia calcula la media y varianza del valor final de una secuencia de sucesos aleatorios, como el rendimiento de un sistema digital de comunicaciones en presencia de ruido e interferencia.

Cada algoritmo procesa un conjunto de soluciones candidatas generadas al azar. Una solución candidata, está formada por una cadena de bits. Así, si se colocan 10 interferentes, la cadena binaria tendrá esa longitud, como se muestra en la Figura 7. Un bit igual a 1 representa a un interferente activo para transmitir al mismo tiempo que el usuario C . Como ejemplo, en la Figura 7 los interferentes 2, 3, 5, y 6 son seleccionados para transmitir simultáneamente con el usuario C . Cuando los algoritmos de asignación de canal siguen los procedimientos inspirados en la naturaleza seleccionan la solución candidata con el valor de C/I más alto.

Durante la evaluación de este ejercicio se consideró una frecuencia de operación del canal de 1920 MHz, 20 soluciones candidatas para los algoritmos AGT, BPSO y DisABC y 100 iteraciones. Para el algoritmo BPSO, los coeficientes social y cognitivo tienen un valor de 2. Para DisABC, ϕ_{max} es de 0.9 y ϕ_{min} tiene un valor de 0.5.

Para el AGT, la probabilidad de mutación es de 0.05 y la probabilidad de cruce es de 0.7.

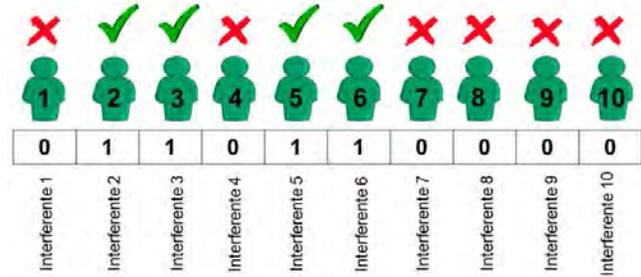


Figura 7. Solución candidata.

Se pueden observar en la Figura 8 los resultados correspondientes al máximo valor de C/I que cada algoritmo encontró durante su ejecución. Cada resultado estuvo en función de la cantidad de usuarios desplegados en la red. Conforme aumenta la cantidad de usuarios disminuyó el valor de C/I , esto se debió a que la interferencia producida por la compartición de espectro entre usuarios también aumenta. También se pudo observar que el algoritmo DisABC fue el que encontró la mejor relación C/I . Esto se debió a que su proceso de búsqueda y su algoritmo de búsqueda local son más eficientes.

La Figura 9 muestra el promedio de todos los valores de C/I encontrados para cada uno de los cuatro algoritmos. El algoritmo DisABC ofreció el mejor resultado mientras que el algoritmo AGT supera a Monte Carlo y BPSO para valores de interferentes menores a 10. Sin embargo, el método Monte Carlo encontró mejores soluciones que el algoritmo AGT y BPSO cuando se presentaron más de 10 posibles interferentes.

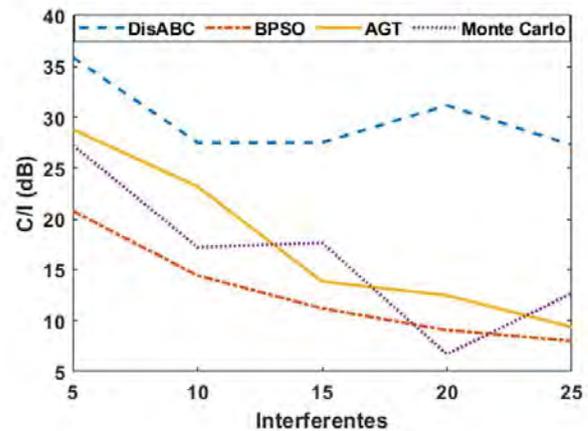


Figura 8. Valores máximos de C/I encontrados.

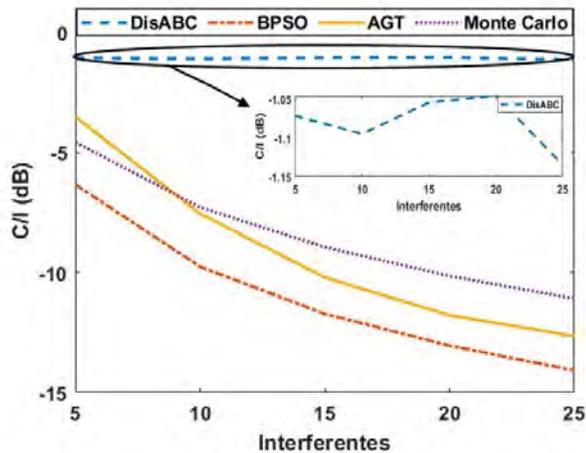


Figura 9. Valores promedio de C/I encontrados.

Conclusiones

Los algoritmos inspirados en la naturaleza se presentan como una alternativa eficaz para el análisis de interferencia en una red celular. También son menos costosos computacionalmente (en tiempo) para aplicarse en las redes celulares futuras, ya que estarán conformadas por un gran número de usuarios móviles. El crecimiento exponencial de usuarios móviles hará inviable el análisis de redes celulares con métodos de optimización exactos.

Actualmente, la academia empieza a explorar algoritmos de optimización inspirados en la naturaleza, pero con un enfoque cuántico. La llegada en los próximos años de la computación cuántica cambiará la forma como se

resuelven los problemas de optimización ya que se basa en la interacción y comportamiento de partículas subatómicas. Este enfoque toma ventaja de los fenómenos de física cuántica de entrelazamiento y superposición, lo que permite la realización de cálculos 108 veces más rápido [7]. Esto hará posible encontrar soluciones de mayor precisión a problemas en ciencias e ingeniería que con las computadoras que utilizamos actualmente no es posible obtener.*

REFERENCIAS

1. Taufique, A., Jaber, M., Imran, A., Dawy, Z., y Yacoub, E. (2017). Planning Wireless Cellular Networks of Future: Outlook, Challenges and Opportunities. *IEEE Access*, 5, 4821-4845.
2. Andrews, J.G., Ganti, R.K., Haenggi, M., Jindal, N., y Weber, S. (2010). A primer on spatial modeling and analysis in wireless networks. *IEEE Communications Magazine*, 48(11), 156-163.
3. Zhai, C., y Tian, J. (2020). Underlay spectrum sharing with adaptive interference cancelation at primary and secondary receivers. *Telecommunication Systems*, 73, 595-605.
4. Eiben, A.E., y Smith, J.E. (2003). Introduction to evolutionary computing. Berlin, Springer.
5. Kennedy, J., y Eberhart, R. (1995). Particle swarm optimization. En *Conferencia IEEE International Conference on Neural Networks*, 1942-1948.
6. Husseinzadeh, M., Nahavandi, K., Nahavandi, N., y Husseinzadeh, K.A. (2012). DisABC: A new artificial bee colony algorithm for binary optimization. *Applied Soft Computing*, 12(1), 342-352.
7. Montiel, O., Rubio, Y., Olvera, C., y Rivera, A. (2019). Quantum-Inspired Acromyrmex Evolutionary Algorithm. *Scientific Reports*, 9.

SOBRE LOS AUTORES



Jesús Alberto Gómez Avilés es ingeniero en software egresado de la Universidad Politécnica de Pachuca. Actualmente se encuentra cursando la Maestría en Tecnologías de la Información y Comunicaciones en la misma Universidad. Su principal interés se centra en algoritmos bio-inspirados, redes neuronales, sistemas embebidos y redes móviles celulares.



Anabel Martínez Vargas tiene Maestría en Computación con especialidad en Redes y Conectividad por la Universidad de Colima. Es Doctora en Ciencias por la Universidad Autónoma de Baja California Campus Mexicali, México. Realizó una estancia posdoctoral en el Centro de Investigación y Desarrollo de Tecnología Digital del Instituto Politécnico Nacional en Tijuana, México. Perteneció al Sistema Nacional de Investigadores de CONACYT. Sus líneas de interés son algoritmos bio-inspirados y optimización. Actualmente es profesora investigadora en la Universidad Politécnica de Pachuca.



Ángel G. Andrade es profesor-investigador en la Facultad de Ingeniería de la Universidad Autónoma de Baja California (UABC), México desde 1998. Su línea de investigación se centra en el desarrollo de Algoritmos de Asignación Dinámica y Compartición de Espectro para Redes Inalámbricas Cognoscitivas. Sus aportaciones principales se centran en la integración de algoritmos de inteligencia computacional y artificial para evaluar la capacidad y calidad en redes de comunicaciones móviles de nueva generación.

ARTÍCULO ACEPTADO

Conceptos básicos de los generadores de números aleatorios (RGN) y pseudoaleatorios (PRGN)

Elena Fabiola Ruiz Ledesma, Elizabeth Acosta Gonzaga y Tonahtiu Eduardo Ramírez Romero

Aspectos básicos de los Generadores de números aleatorios y pseudoaleatorios

Un Generador de Números Aleatorios o pseudoaleatorios, *Random numbers generator* (RNG por sus siglas en inglés), tiene como función principal el producir una secuencia de números enteros en un intervalo [mínimo, máximo], de manera que los valores no se puedan predecir y lo que busquen se expresa con las siguientes características [1], [2]:

- 1) Cada nuevo valor debe ser estadísticamente independiente del valor anterior. Es decir, que la ocurrencia de un valor no afecta para la ocurrencia del que le sigue, dada una secuencia de valores generada.
- 2) La distribución general de los números elegidos del intervalo tiene una distribución uniforme. En otras palabras, todos los números son igualmente probables y ninguno es más popular o aparece más frecuentemente dentro de la salida del RNG que otros.
- 3) La secuencia de números no se puede predecir. En caso de que hubiera una persona mal intencionada pretendiendo adivinarla, no podría hacerlo.

Como los sistemas informáticos son deterministas por naturaleza, producir números aleatorios de calidad que tengan estas propiedades (independencia estadística, distribución uniforme e imprevisibilidad) es mucho más difícil de lo que parece.

Más allá de estos requisitos, algunas otras propiedades deseables de los RNG incluyen:

- 1) El RNG es rápido en devolver un valor (es decir, bajo tiempo de respuesta) y puede atender un gran número de solicitudes dentro de un corto intervalo de tiempo (es decir, altamente escalable).
- 2) El RNG es seguro contra los atacantes, quienes buscan formas de cambiar su estado subyacente para predecir o influir en su salida o interferir de otro modo con su funcionamiento.

La Importancia de los Números Aleatorios

El ser humano durante años ha estado en contacto con sistemas aleatorios pertenecientes a la naturaleza y se

ha encontrado en la posición de tomar decisiones en condiciones de duda; por ello, desde la antigüedad se ha visto interesado por los juegos de azar, el destino y futuro, por lo que ha intentado realizar predicciones, ya sea por medio de la adivinación, el azar mismo, «la suerte», o mediante estudios formales como el estudio de fenómenos aleatorios que corresponde a Probabilidad y la Estadística. Esto último es el punto de partida del análisis de los números aleatorios (random numbers). El número aleatorio es aquel obtenido al azar, es decir, que todo número tenga la misma probabilidad de ser elegido y que la elección de uno no dependa de la elección del otro. Para abordar el tema de la generación de números aleatorios se requiere asentar las bases para su generación.

Evolución histórica de los métodos de generación de números aleatorios

Formalmente, se dice que los números aleatorios comenzaron a crearse artificialmente entre los años 1940 y 1949, y aunque no hay una forma de comprobarlo, en dichos años, surgió un método de generación de números aleatorios llamado *La simulación de Monte Carlo*, es un método que combina conceptos estadísticos con el muestreo aleatorio, con la generación de números aleatorios y la automatización de los cálculos. Por su parte Von Neumann, fue pionero en la investigación de los generadores de números aleatorios. En la década de los 70s. Neumann hizo un señalamiento que ahora se puede ver con claridad: “La computadora, ciertamente abrirá un nuevo enfoque para las estadísticas matemáticas y revolucionará el enfoque para el cálculo de experimentos” [3] .

Siguiendo la línea del tiempo se puede hacer una división en los generadores de números aleatorios, una primera etapa corresponde a antes de la aparición de las computadoras y una segunda abarca desde el surgimiento de las mismas.

Previo a las computadoras, existieron diversos artefactos que creaban números aleatorios, como en el caso de las ruletas, el lanzamiento de dados o de monedas (Figura 1).

Sin embargo, los métodos de generación de números aleatorios producidos por los artefactos como las ruletas, no pudieron ser usados en su totalidad en un modelo computacional, debido a la restricción de cómputo que había a sus inicios.



Figura 1. Imagen ruleta.

Aun así, hubo publicaciones como la de Babington Smith, que presentó 100,000 números aleatorios obtenidos mediante un disco giratorio que se iluminaba con un rayo.

En un inicio los primeros generadores de números aleatorios producían números sin seguir un patrón de comportamiento, posteriormente, al descubrir la factibilidad de seguir un patrón, es que fueron mejorados los RNG. Esto hizo que la investigación de los métodos para la generación de números aleatorios continuara avanzando a lo largo del tiempo. No es sino hasta 1951 cuando Lehmer [4] lanzó públicamente la propuesta de un generador de congruencia lineal, el cual recibió algunas propuestas de modificación por parte de Thomson y Tottenberg [5], consolidándose como el método por defecto para la generación de números aleatorios. Este generador es utilizado hasta la fecha para la generación de números aleatorios.

Hoy en día se han creado muchísimas formas de generar números aleatorios, por ejemplo, usando determinadas funciones en diferentes lenguajes de programación como JavaScript o Python.

Es importante mencionar la forma en que se involucraron los números aleatorios en la computación, hasta el grado de ser algo imprescindible al momento de dar soluciones no deterministas, es decir, soluciones que no se pueden predecir, o inclusive usar un pequeño proceso no determinista dentro de un programa determinista (que sería un programa que puede predecir los resultados).

¿Dónde se emplean los números aleatorios?

Los números aleatorios son un elemento clave en múltiples procesos digitales. Estos se utilizan no sólo en aplicaciones con gran componente de aleatoriedad como puede ser el juego en línea, sino que también tienen múltiples aplicaciones en el mundo de la ciberseguridad. En algunos sistemas criptográficos la seguridad depende de algo únicamente conocido por el personal autorizado, pero impredecible para los atacantes, tal y como pasa con los dispositivos de autenticación electrónicos de bancos, que generan números aleatorios en forma de clave de dígitos

y cambian cada minuto, para la seguridad de la transacción (Figura 2).



Figura 2. Imagen que muestra algunos dispositivos de autenticación.

En el caso de los casinos hay software que produce números aleatorios, estos programas son aleatorios y se relacionan con el algoritmo que utiliza el casino para funcionar. Permiten determinar el símbolo o los símbolos que van a aparecer a continuación, la carta que va a salir o cuándo los carretes dejarán de girar en los juegos de tragamonedas (Figura 3).



Figura 3. Máquinas tragamonedas.

Pero no solo se limitan al mundo de los casinos. De hecho, se pueden encontrar en diversas industrias en todo el mundo. Algunos se usan en videojuegos, otros en loterías.

Un ejemplo de software que puede generar números aleatorios es MATLAB.

Los números aleatorios permiten el desarrollo de algoritmos de análisis, control y modelado de sistemas. Por ejemplo, se tiene el caso de la red de cajeros automáticos ATM (por sus siglas en inglés Automated Teller Machine) donde cada transacción debe ser identificada con un número aleatorio de transacción; esto es de suma importancia, debido a que si los números utilizados para la transacción fueran predecibles, entonces podrían falsificarse transacciones y difícilmente el órgano regulador de

ellas podría ser capaz de identificar de dónde provino la falla, por dicho motivo, es estrictamente necesario hacer uso de una fuente de números aleatorios (Figura 4).



Figura 4. Imagen de un cajero automático.

Generadores de números pseudo-aleatorios (PRNGs)

Un enfoque ampliamente utilizado para lograr un buen comportamiento estadístico de RNG es aprovechar el modelado matemático en la creación de un generador de números pseudoaleatorios (PRNG) [13]. Un PRNG es un algoritmo determinístico, típicamente implementado en un software que calcula una secuencia de números que "parecen" al azar. Un PRNG requiere un valor semilla (valor inicial), que se utiliza para inicializar el estado del modelo subyacente. Una vez sembrados, puede generar una secuencia de números que muestren un buen comportamiento estadístico [6].

Los PRNG muestran una periodicidad que depende del tamaño de su modelo de estado interno. Es decir, después de generar una larga secuencia de números, todas las variaciones en el estado interno se agotarán y la secuencia de números a seguir repetirá una secuencia anterior [7]. Sin embargo, los mejores algoritmos de PRNG disponibles en la actualidad tienen un período que es tan grande que prácticamente se puede ignorar esta debilidad. Por ejemplo, el PRNG Mersenne - Twister MT19937 con longitud de palabra de 32 bits tiene una periodicidad de $2^{19937} - 1$ [8].

En algunos contextos, la naturaleza determinista de los PRNG es una ventaja. Por ejemplo, en algunos contextos experimentales y de simulación, a los investigadores les gustaría comparar el resultado de diferentes enfoques utilizando la misma secuencia de datos de entrada. Los PRNG proporcionan una forma de generar una larga secuencia de entradas de datos aleatorios que se pueden repetir utilizando el mismo PRNG, sembrado con el mismo valor.

En otros contextos, sin embargo, este determinismo es altamente indeseable. Considere una aplicación de ser-

vidor que genere números aleatorios para ser utilizados como claves criptográficas en intercambios de datos, con aplicaciones cliente a través de canales de comunicación seguros. Un atacante que conoce el PRNG en uso y también conoce el valor semilla (o el algoritmo utilizado para obtener un valor semilla) será capaz de predecir rápidamente cada clave (número aleatorio) a medida que se genera. Incluso con un algoritmo de siembra sofisticado y desconocido, un atacante que sabe (o puede adivinar) el PRNG en uso, puede deducir el estado del PRNG al observar la secuencia de los valores de salida. Después de un número sorprendentemente pequeño de observaciones (por ejemplo, 624 para Mersenne Twister MT19937), se pueden predecir todos y cada uno de los siguientes valores. Por esta razón, los PRNG se consideran criptográficamente inseguros [8].

Los investigadores de PRNG han trabajado para resolver este problema creando lo que se conoce como PRNG criptográficamente seguros (CSPRNG). Se han inventado varias técnicas en este dominio, por ejemplo, una de ellas consiste en aplicar un hash criptográfico, es decir, una función que convierta un número grande de datos en una longitud fija, para lo cual se emplean números de entre 0 y 9 y letras entre A y F, usando un cifrado de bloque o una unidad de código, que permite cifrar la secuencia de enteros generada. Además del cifrado por bloques está el cifrado de flujo que opera sobre dígitos individuales. Tales enfoques abordan el problema de evitar inferir un PRNG y su estado, al aumentar en gran medida su complejidad computacional, pero los valores resultantes pueden o no exhibir las propiedades estadísticas correctas (es decir, independencia, y distribución uniforme) necesarias para un generador robusto de números aleatorios. Además, un atacante podría descubrir cualquier algoritmo determinista por varios medios (por ejemplo, desensambladores, ataques de memoria sofisticados, un empleado descontento). Aún más común, los atacantes pueden descubrir o inferir el valor semilla (inicial) de PRNG al reducir su rango de valores posibles o la memoria de indagación de alguna manera. Una vez que se conoce el algoritmo determinista y su semilla, el atacante puede predecir cada número aleatorio generado, tanto pasado como futuro.

Comparación entre programas que generan PRNGs

En esta sección solo se revisan los programas que existen para generar números aleatorios. A lo largo del tiempo, muchos algoritmos han sido evoluciones de algoritmos anteriores, y la mayoría de los más recientes tienen una ventaja diferencial en relación a los anteriores, por lo que son pocos los que hoy en día se siguen usando, motivo por el que se descartaron los algoritmos no vigentes a la fecha. De esta forma, se comenzó la comparativa a partir

Tabla 1. Programas computacionales que generan PRNGs.

Algoritmo	Costo de implementación	Actuales Usos	Análisis
Mersenne Twister [8]	Es un código de aproximadamente 50. líneas escrito en C, que únicamente utiliza la librería standard <stdio.h>	Es utilizado para corregir la mayoría de los defectos encontrados en PRNG más antiguos	Este algoritmo es sencillo de implementar, basta tener 685 salidas para deducir la secuencia de números que dará, por lo que su uso se recomienda en sistemas cuya aleatoriedad sea de poca importancia (como desordenar una lista).
Xorshift [9]	Este algoritmo es de los más complicados de implementar, tiene fundamentos matemáticos matriciales	Ayuda a la rapidez de las. arquitecturas de computadoras modernas	Xorshift está preparado para su ejecución de forma continua sin que el algoritmo se desborde como puede pasar con otros.
Philox [10]	El código que implementa se encuentra en la biblioteca "Random123" está incluido en algunas versiones recientes de la biblioteca Intel Math Kernel	Uso en procesadores de computadoras	Philox está pensado para utilizarse en estadística, de modo que no solo contiene un algoritmo de números pseudoaleatorios, sino que además contiene una serie de herramientas muy útiles para pruebas y manipulación de los números y cómo se generan
Splix Mix [11]	Es un algoritmo orientado a objetos, su mayor uso reside en Java, porque Oracle decidió incluirlo en sus operaciones rand nativas	Está en casi todas las aplicaciones Android, ya que éste usa java para su funcionamiento.	Es un algoritmo que tiene una dispersión buena, y con un alto grado de periodicidad, pero tiene la debilidad de ser altamente predecible
Xoroshiro128+ [12]	Es un código complicado de codificar y de entender, este algoritmo fue creado durante el año 2018, siendo actualmente el algoritmo más avanzado por mantener el factor de impredecibilidad.	Es un algoritmo nuevo, aún se encuentra en evaluaciones	Mantiene una dispersión equitativa y el hecho de que cambia constantemente su comportamiento, evita la predictibilidad

del año 1998 con el algoritmo de Mersenne Twister, el cuál fue uno de los que ha trascendido hasta la actualidad.

La comparativa de los PRNGs, se presenta en la Tabla 1.

Análisis de los programas computacionales que generan PRNGs

Al comparar los algoritmos PRNGs se puede apreciar que Xoroshiro128+ [12], es el mejor algoritmo para generar números pseudoaleatorios con una gran entropía debido a su dispersión, sin embargo necesita altas cantidades de procesamiento rápido para poder almacenar su información ya que genera salidas muy grandes, de modo que el mejor modo de acceder a este algoritmo es mediante un Webservice donde un servidor se encargue de procesar y generar las salidas, y todos los clientes únicamente reciban un valor de alguno de los números generados. De esta manera no se satura el cliente procesando dicho algoritmo. Inclusive siendo el mejor de los algoritmos, no deja de ser un algoritmo de números pseudoaleatorios, de tal manera que la mejor forma de obtener números aleatorios es mediante un módulo de hardware como el que tiene desarrollado Intel en sus procesadores más recientes, y es un módulo como el que se acopla a los cajeros automáticos para crear un identificador único de transacción que se sincroniza día con día con el servidor central del banco, de este modo se impiden los fraudes bancarios.

Por otro lado, si se tiene el algoritmo Splix Mix [11], el cual es complejo pero eficiente y genera salidas mucho menos impredecibles pero suficiente para su campo de aplicación, motivo por el cual fue implementado dentro de java, y ahora se encuentra dentro de la mayoría de aplicaciones en sistemas Android, de forma que a diario es probable que se use este algoritmo para tareas que ni

quiera se podría imaginar que implican todo este desarrollo.

Finalmente, Philox [9] es uno de los algoritmos con mayor implementación en diversos programas para la investigación o en programas con entornos fijos como Windows, macOS y Linux. Por otro lado, el algoritmo de Xorshift [10] está preparado para procesar cantidades de datos masivos, motivo por el cual se usa en dispositivos de análisis de alto desempeño, como las unidades de cálculo para gráficos, ya que ahí se puede aprovechar su potencial porque este software es de los pocos que están preparados para generar más datos pseudoaleatorios.

Por último, está el Mersenne [8], el cual sería la recomendación ideal para ser implementado en la práctica por cualquier desarrollador con requerimientos comunes de software, ya que es el único que puede ser implementado en el código fuente de prácticamente cualquier sistema, y es bastante confiable. Aunque para cualquier asunto fuera del recreativo, no se recomienda usarlo.

Conclusiones

En este documento se revisó que la generación de números aleatorios y pseudoaleatorios, tiene un campo amplio de uso y la selección de un programa PRNGs depende de la aplicación que se le dará, así como de su costo y de sus características computacionales. Solo fueron revisados los generadores de números aleatorios con distribución uniforme, pero existen con otras distribuciones. *

REFERENCIAS

- Demirhan, H., y Bitirim, N. (2016). Statistical Testing of Cryptographic Randomness. *Journal of Statisticians: Statistics and Actuarial Sciences*, 9, 1-11.
- Park, S.K., y Miller, K.W. (1988). Random Number Generators: Good Ones Are Hard To Find. *Communications of the ACM*, 1192-1201.

3. Von Neumann, J. (1951). Various techniques used in connection with random digits Monte Carlo Method, National Bureau of Standards. *Applied Mathematics Series*, 12, 36-38.
4. Lehmer, D.H. (1951). Mathematical methods in large-scale computing units. En *Proc. 2nd Symposium on Large-Scale Digital Calculating Machinery*. 141-146.
5. Tausworthe, R.C. (1962). Random Numbers Generated by Linear Recurrence Modulo Two. *Mathematics of Computation*, 19, 201-209.
6. Dodis, Y., Pointcheval, D., Ruhault, S., y Vergnaud, D. (2013). Security analysis of pseudo-random number generators with input:/dev/random is not robust. En *Sadeghi A. R., Gligor V. D. y Yung M. (eds.) ACM CCS 13*, 647-658.
7. Unkašević, T., Banjac, Z., y Milosavljević, M. (2019). A Generic Model of the Pseudo-Random Generator Based on Permutations Suitable for Security Solutions in Computationally-Constrained Environments. *Sensors*, 19(2), 22-53.
8. Matsumoto, M., y Nishimura, T. (1998). MersenneTwister: A623-dimensionally Equidistributed Uniform Pseudo-Random Number Generator. *ACM Trans. on Modeling and Computer Simulation*, 3-30.
9. Marsaglia, G. (2003). Xorshift RNGs. *Journal of Statistical Software*, 8-14.
10. Almon, M., Moraes, R., y Shaw, D. (2011). Parallel random numbers: as easy as 1, 2, 3. En *Proc. International Conference for High Performance Computing, Networking, Storage and Analysis*, 1-12.
11. Steele, G.L.J., Lea, D., y Food, C.H. (2014). Fast splittable pseudorandom number generators. En *OOPSLA '14 Proc. of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications*, Portland, EEUU, 453-472.
12. Lemire, D., y O'Neill, M.E. (2019). Xorshift1024, xorshift1024+, xorshift128+ and xoroshiro128+ fail statistical tests for linearity. *Journal of Computational and Applied Mathematics*, 350, 139-142.

SOBRE LOS AUTORES



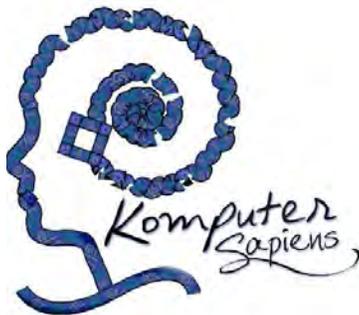
Elena Fabiola Ruiz Ledesma es docente-investigadora en la Escuela Superior de Cómputo del Instituto Politécnico Nacional (ESCOM-IPN), y en la UPIICSA del IPN. Es Licenciada en Matemáticas, con Maestría y Doctorado en Ciencias, especialidad Matemática Educativa, por el Centro de Investigación y Estudios Avanzados del IPN. Sus intereses de investigación incluyen cómputo educativo y matemática educativa.



Elizabeth Acosta Gonzaga es docente-investigadora en la SEPI-UPIICSA del Instituto Politécnico Nacional. Es Licenciada en Ciencias de la Informática por la UPIICSA del IPN, con Maestría en Ingeniería de Sistemas Empresariales por la Universidad Iberoamericana, y Doctorado en Investigación Educativa por la Universidad de Manchester, Reino Unido. Su investigación se enfoca en la tecnología educativa y administración de los sistemas de información. Es miembro del Sistema Nacional de Investigadores nivel I.



Tonhatiu Eduardo Ramírez Romero es profesor titular y Jefe de la Sección de estudios de posgrado e investigación en la Escuela Superior de Cómputo. Es Doctor en ingeniería de sistemas y Maestro en ciencias en ingeniería de sistemas por la ESIME Zacatenco IPN, Licenciado en ciencias de la informática por UPIICSA IPN, ha participado con 15 ponencias internacionales y 8 nacionales, ha publicado 5 artículos internacionales arbitrados y 3 artículos de difusión científica. Ha graduado alumnos de posgrado y de ingeniería.



INVITACIÓN A PUBLICAR EN KOMPUTER SAPIENS 2020

Revista indexada en:



ARTÍCULO ACEPTADO

Módulo de rutas para la entrega de comida a través de un algoritmo Optimizado por enjambre de partículas (PSO) y con tecnología de Google maps

Sergio Soto Rodríguez, Edmundo Bonilla Huerta, Alberto Portilla Flores, José Crispín Hernández Hernández y Oscar Atriano Ponce

Resumen

En este artículo se propone un módulo para la optimización de rutas de reparto para una aplicación móvil la cual está enfocada al sector restaurantero. Se utiliza un algoritmo PSO para encontrar la ruta óptima para la entrega de comida, minimizando así los costos de reparto y maximizando el número de entregas. En las experimentaciones se pone a prueba el algoritmo con Google maps (API) para generar la mejor ruta de entrega de los pedidos de los restaurantes. Los resultados obtenidos son prometedores, y permiten que el proceso de entrega se realice de forma eficiente.

Introducción

La entrega de alimentos se ha convertido en los últimos años en un servicio ampliamente utilizado por restaurantes. En general un restaurante recibe pedido de clientes desde diferentes lugares. El restaurante debe por su parte entregar los pedidos solicitados por los clientes lo más rápido posible. Par esto se requiere de una aplicación que determine la trayectoria más corta para hacer las entregas. Aunque la planificación de entrega de comida se puede abordar desde la perspectiva del problema del agente viajero (PSP), el optimizador del TSP se enfoca en obtener una única solución. En la vida real se deben incluir otras restricciones como el tráfico, el sentido de las calles y la experiencia del personal que realiza la entrega en algún tipo de vehículo (camioneta, automóvil, motocicleta o bicicleta). Por ello es conveniente contar con varias alternativas en tiempo real que permitan al vehículo efectuar las entregas en el tiempo establecido por la aplicación. En este artículo se propone implementar en una aplicación móvil el algoritmo PSO en combinación con la interfaz de Google maps (API) para la generación de rutas de entrega. Considerando los siguientes beneficios:

- Mejorar el rendimiento de entregas de productos.
- Proponer una mejor contratación de personal para la repartición.
- Reducir costos de entrega.
- Tener un mejor control de reparto.
- Ahorro de tiempo de entrega.

Este artículo presenta la implementación de un algoritmo PSO para la solución de la problemática de reparto de comida. Un factor de la lealtad de los clientes es el servicio de entrega, que contempla la rapidez, que el producto no esté maltratado, y la temperatura de la comida, los cuales son factores dependientes de la eficacia con la que se hace la entrega del producto preparado.

Este es un problema de planeación de rutas el cual se relaciona al problema del agente viajero (TSP). Para resolver esta problemática se toma en cuenta la geolocalización de cada persona que realiza un pedido a los diferentes restaurantes. La distancia del restaurante es un factor de importancia ya que un pedido de comida caliente no puede tardar tanto tiempo en entregarse.

Se realizó un módulo para la aplicación "Food express", una aplicación móvil enfocada para solicitar comida a domicilio y reservaciones en restaurantes, la cual fue desarrollada por Smartsoft America Bussines Applications S.A. de C.V. Food Express está disponible para sistemas operativos Android e IOS. El uso de la aplicación es simple, un usuario con la aplicación en su smartphone entra al restaurante de su agrado, ingresa al menú, selecciona su platillo, agrega una bebida si así lo desea o algún tipo de postre. Accede al carrito y procede al pago para que de esta manera el restaurante reciba la orden solicitada. Posteriormente un repartidor asociado a la aplicación recibe una notificación del restaurante donde se tiene que recoger el pedido a entregar. De esta manera el repartidor procede a entregar la comida al cliente que solicitó la comida desde la aplicación [1].

Nuestra propuesta asume que es necesario conocer la ubicación de cada usuario, ya que se toma un valor estático, es decir, si el pedido se realiza en movimiento el algoritmo toma la longitud y latitud en el momento que se hace el pedido. Estos parámetros se obtienen desde el dispositivo móvil de cada usuario. Una vez obtenida la longitud y latitud se agregan al mapa de Google maps por medio de una interfaz de programación de aplicaciones (API), para calcular la trayectoria de los diferentes puntos para aplicar el algoritmo de optimización, después generar la mejor trayectoria con las condiciones que impone un algoritmo (PSO).

El problema toma importancia ya que en la actualidad la mayoría de los restaurantes ofrecen el servicio de entrega a domicilio de manera gratuita, el vehículo en el cual hacen las entregas generalmente suele ser una motocicleta, la cual tiene un bajo consumo de gasolina, sin embargo, la implementación de este algoritmo puede economizar aún más los costos de gasolina y administrar de mejor manera el personal de entregas, cubriendo más pedidos una sola persona.

Google maps juega un papel importante para este proyecto, ya que es el que facilita el cálculo de las coordenadas de cada usuario que solicita un pedido y a su vez regresa una interfaz donde se encuentran las calles con direcciones, locales, por mencionar algunas herramientas de Google maps.

Estado del arte

En la literatura se ha abordado el problema de enrutamiento de vehículos (ERP) o Vehicule Routing Problem (VRP), para encontrar rutas de vehículos que visiten todas las ubicaciones de los clientes con una flota de vehículos, minimizando el tiempo y los costos.

Se propone en [2] un modelo híbrido el cual utiliza una Colonia de Abejas Artificiales (ABC) y un Algoritmo genético (GA) para resolver el VRP. En el primer paso, ABC se ha utilizado para encontrar una solución de optimización de rutas para cinco vehículos. En el segundo paso, el algoritmo genético ayuda a optimizar las soluciones obtenidas previamente por ABC. Los resultados muestran que las actividades de las abejas exploradoras son el factor más efectivo en las respuestas al problema y ayuda a converger rápidamente en la obtención de la ruta más óptima. En [3] los autores proponen el algoritmo de optimización de colonias de hormigas (ACO), el cual se emplea para resolver el VRP con el objetivo de determinar el número de sub-flotas de vehículos para realizar las entregas y las rutas óptimas para minimizar el costo total (ruta fija y costos de viaje). Una propuesta para resolver el VRP, se presenta en [4], con la extensión de enrutar vehículos ecológicos con múltiples depósitos (MDGVRP), para maximizar los ingresos y minimizar los costos, así como el tiempo de entrega. En esta propuesta se aplica una mejora al algoritmo ACO para resolver el problema VRP de manera eficiente. Una propuesta similar es reportada en [5], en ella se propone abordar las restricciones de tiempo y cooperación de depósitos de vehículos, para esto se aplica el algoritmo ABC para resolver el problema de múltiples depósitos (MDVRP). En una primera etapa, se utiliza la técnica de agrupamiento para reducir el MDVRP a VRP. En una segunda etapa se modifica el algoritmo ABC para obtener un VRP de depósito único y así generar soluciones para cada depósito. Finalmente se propone una estrategia de co-evolución para combinar los depósitos y generar una solución completa del MDVRP.

Modelo propuesto

Nuestro problema está asociado a funciones de Google maps que ofrece el servicio de calcular distancias de un punto A a un punto B, en un tiempo estimado de recorrido, pero este cálculo se hace mediante matrices de distancia, el cual utiliza los parámetros de latitud y longitud de cada punto, y lo proyecta en una interfaz de un mapa mundial digital, esto con el fin de proporcionar al usuario un tiempo estimado de tiempo de recorrido y distancia. El API de Google maps generalmente es utilizado en páginas web, o aplicaciones móviles y GPS (Sistema de Posicionamiento Global).

Una aplicación móvil comercial enfocada al sector restaurantero que hace uso de las herramientas de Google maps y el posicionamiento global es conocida como "Sin delantal" [6]. Utiliza una interfaz de Google con un concepto de mapas para poder seleccionar donde se está realizando el pedido para el reparto. De manera pública no se muestra el tipo de estrategia que tiene para la entrega de reparto para los pedidos de los diversos restaurantes, pero se busca hacer una entrega a la vez por cada repartidor. Sin delantal está desarrollado con lenguaje Android e IOS para utilizar recursos de WebServices REST., Google maps (API), Google Analytics, son algunos de sus herramientas utilizadas para su desarrollo. No es público si usan o no algún tipo de inteligencia para hacer la logística de reparto.

TSP

El problema del agente viajero conocido por la abreviación en inglés TSP (Travelling Salesman Problem), consiste básicamente en un viajero que quiere visitar n ciudades, solo una vez cada ciudad. Empezando por cualquiera de ellas y regresando al mismo lugar del que partió. Este problema puede resolver situaciones de la vida real que pueden formularse de diferente manera. El problema del agente viajero ayuda a solucionar problemas dentro de la robótica, mecánica, industria automotriz y logística principalmente enfocado a problemas de optimización [7].

Antecedentes

PSO

Dentro de los algoritmos bioinspirados existe uno muy frecuentemente utilizado en la literatura para la optimización de recorridos conocido como PSO (Particle Swarm Optimization) por sus siglas en inglés, que su traducción al español sería optimización con enjambre de partículas. Esta metodología surge al tratar de imitar el comportamiento de las parvadas o cardumen. El algoritmo de optimización por enjambre fue desarrollado por Kennedy y Eberhart [8]. Una de las hipótesis de cada partícula o agente que representa a las abejas, hormigas o pájaros o algún individuo que este en un grupo social con

una búsqueda guiada, la partícula que mejor solución ha encontrado hasta el momento cumple con la correcta función como líder. La idea del algoritmo propuesto es obtener una solución de partícula que evolucione para poder encontrar una mejor solución en su recorrido.

Dentro de la literatura, el fundamento teórico de este método es que el movimiento de cada una de las partículas existentes enfocadas a un objetivo en común está condicionado por dos factores: la primera es la nostalgia o memoria autobiográfica de la partícula y la influencia social del grupo o del enjambre.

Cada partícula tiene una posición instantánea de la población en el espacio de N-dimensional representa dentro del dominio de la función objetivo que se propone para una posible solución, N es el número de incógnitas del problema. Además, el proceso evolutivo se disminuye al mover cada partícula dentro del espacio de soluciones con una velocidad que se transformará de acuerdo a su velocidad actual, la memoria de la partícula e información global dentro de todo el enjambre.

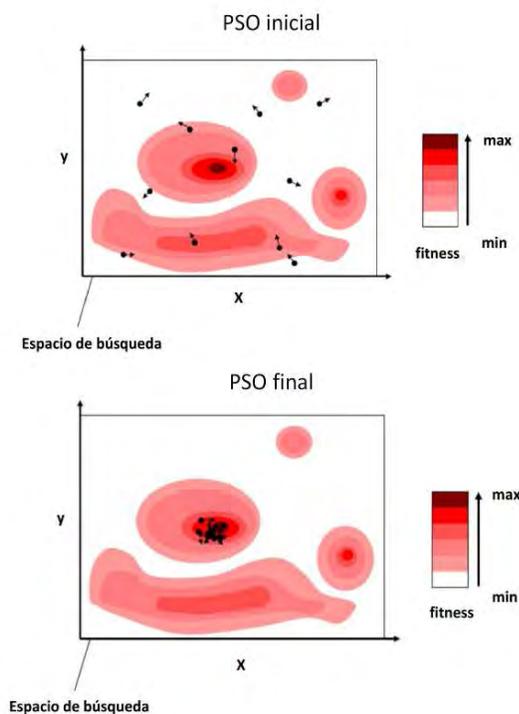


Figura 1. Ejemplo de algoritmo PSO inicio y final.¹

Dentro del espacio de búsqueda se optimizan las partículas mediante la solución fitness que es el mejor espacio de solución dentro del espacio de búsqueda (ver Figura 1). Al comenzar el algoritmo las partículas se mantienen distribuidas en el espacio, una vez que comienza

a ejecutarse el algoritmo las partículas van encontrando una solución fitness individualmente y grupalmente para poder encontrar un punto donde la solución sea la más óptima para las partículas.

El PSO consiste en un proceso iterativo y estocástico que opera sobre un cúmulo de partículas. La posición de cada partícula representa una solución potencial al problema que se esté resolviendo. Regularmente, una partícula ϕ_i está compuesta de tres vectores cada uno con dos valores de velocidad también conocido como fitness. Los siguientes valores componen el *fitness*.

- El vector $X_i = X_{i1}, X_{i2}, \dots, X_{in}$ almacena la posición actual de la partícula en el espacio. El tamaño de este vector depende del número de variables necesarias para resolver el problema.
- El vector $\phi Best_i = \phi_{i1}, \phi_{i2}, \dots, \phi_{in}$ almacena la posición de la mejor solución encontrada por la partícula hasta el momento.
- El vector de velocidad $V_i = V_{i1}, V_{i2}, \dots, V_{in}$ almacena la dirección según el cual se moverá la partícula.
- El valor de fitness $fitnessX_i$ almacena el valor de acuerdo de la solución actual (vector X_i).
- El valor fitness $fitness \phi Best_i$ almacena el valor de acuerdo de la mejor solución local encontrada hasta el momento (vector $\phi Best_i$).

Experimentos y pruebas

Se implementó un software, para la optimización de recorridos de entregas de comidas para el sector restaurantero, aplicando el algoritmo bioinspirado de PSO el cual beneficiará a las entregas de comidas optimizando tiempo, costos de gasolina y una mejor administración de reparto, además, dándole solución a la problemática del agente viajero la cual se funde con la incorporación de las herramientas de Google maps, para tener una interfaz visual agradable para el usuario final.

El algoritmo tiene la tarea de ser mejor que el cálculo de matrices del Google maps, trabajando de manera más eficiente y de manera inteligente [9]. Nuestra primera actividad es realizar un diseño conceptual para poder interpretar las necesidades del sector restaurantero, comprender de manera ilustrada el funcionamiento del algoritmo y como resolvería el problema del agente viajero dándole eficacia a la entrega.

Después se necesita la geolocalización del cliente para poder obtener su latitud y longitud, donde se realizará

¹Di Caro G.A. (2012) Collective and Swarm Intelligence. [Figura 3]. Recuperado de <https://sci2s.ugr.es/sites/default/files/files/Teaching/GraduatesCourses/Metaheuristics/Tema06-Metaheur%C3%ADsticas%20basadas%20en%20adaptaci%C3%B3n%20social%20-18-19.pdf>

la entrega de comida, esa información llega a la aplicación en la cual se generan los pedidos para la entrega de comida.

Una vez recibida la ubicación, se procede a activar el web service, donde obtendrá la petición de los diferentes puntos de entrega. Recabando los parámetros de geolocalización de los usuarios se activa el algoritmo PSO, donde encuentra la mejor ruta de entrega para el restaurante. El restaurante obtiene una recomendación del algoritmo dentro de la interfaz de Google maps, de manera que visualiza su posible trayectoria de reparto. En la Figura 2 se representa la solución usada para varias entregas de comida. El restaurante podrá recibir distintos pedidos, ubicaciones y diferentes consumos. Una vez obtenida la geolocalización del usuario más el pedido, se procesa a través de la aplicación y del web Service. Posteriormente se ejecuta el algoritmo con cierto número de iteraciones para encontrar la mejor solución a través de la matriz de distancias, para entrega el mayor número posible de entregas, con un solo repartidor en un solo viaje. La sugerencia del recorrido es regresada al restaurante para que decida la mejor entrega posible.

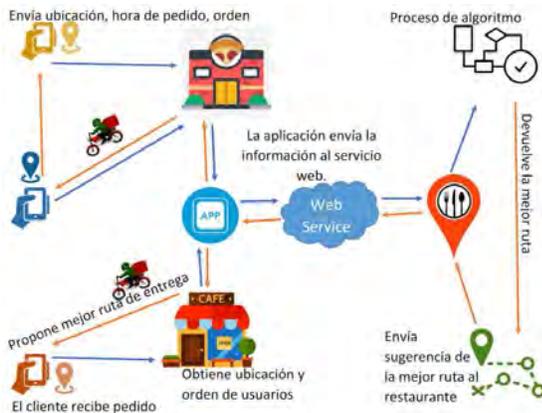


Figura 2. Diseño conceptual de resolución del problema en base a las entregas de comida.

Una vez establecido el diseño conceptual, el siguiente paso fue desarrollar la interfaz donde se muestre el mapa global de Google maps (API), mostrando la ubicación de los diferentes puntos de entrega.

Ejecutando el algoritmo PSO con los puntos de entrega arroja una recomendación para está calculando el tiempo de entrega, las iteraciones del algoritmo y los puntos de entrega. Algunos parámetros para el algoritmo se tienen establecidos como lo son: el tamaño de las partículas, la posición de las partículas, la velocidad, la partícula fitness para poder ejecutar el algoritmo de manera correcta. Una vez tomando los parámetros se plasma de manera gráfica en el mapa con la ruta propuesta por el algoritmo y el API. Mostrando diferentes resultados que son: los puntos de entrega, las iteraciones del algoritmo,

tiempo estimado de recorrido, orden de entrega con base a los pedidos por entregar, recorrido total en kilómetros y una ruta propuesta por los métodos de Google maps.

Inicialmente con el experimento se analizó una entrega con 5 puntos de entrega, utilizando una iteración de 50 para encontrar el mejor camino optimizado. El algoritmo maneja una velocidad de partícula que en este caso es de 0.1, con un espacio de solución de 50. Al aplicar el algoritmo en el mapa genera una recomendación de entrega al repartidor (ver Figura 3).

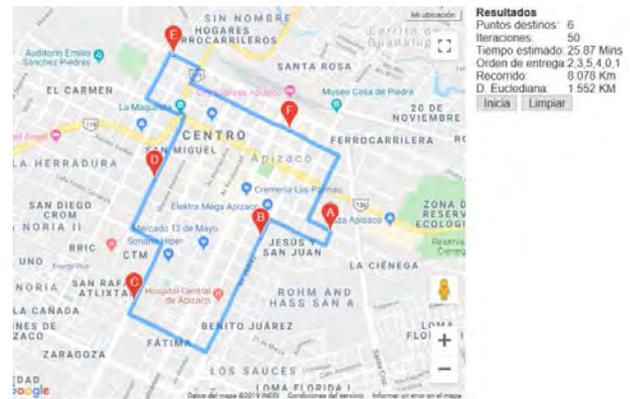


Figura 3. Ejecución del optimizador de rutas con algoritmo PSO (5 puntos de entrega).

Se muestran 6 puntos de entrega porque uno es de el punto de partida, donde se encuentra el restaurante, es por ello que tiene que haber un retorno al punto inicial. El camino es trazado mediante la librería de Google maps, las marcas dentro del mapa son utilizados por la misma librería.

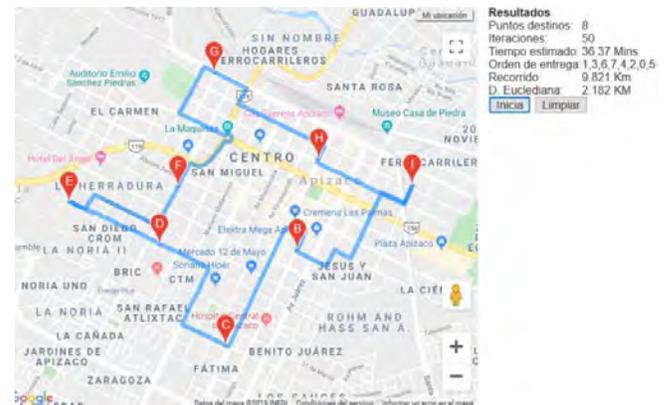


Figura 4. Ejecución del optimizador de rutas con algoritmo PSO (7 puntos de entrega).

La siguiente prueba fue realizada con 7 puntos de entrega, 50 iteraciones en el espacio de búsqueda para encontrar la mejor ruta óptima. El resultado que se visualiza en el mapa (ver Figura 4) es una sugerencia para los repartidores. Con base a que el algoritmo necesita

una velocidad de partícula que para esta prueba fue usada una velocidad de 0.1.

Uno de los requisitos a cubrir para este problema fue conocer la distancia del recorrido total, esta medición se calcula mediante la matriz de la API, que calcula el recorrido sobre calles o el trayecto del reparto entre las calles.

A continuación se presenta la prueba con una entrega de 9 pedidos, con 50 iteraciones a una velocidad de 0.1, ya que se es recomendado en la literatura. En esta prueba se visualiza un mayor número de entregas, lo cual se ve reflejado en el tiempo de entrega estimada (ver Figura 5), por lo que se recomienda esta cantidad de pedidos siempre y cuando los pedidos sean cerca del restaurante, contemplando que sean comidas que se sirvan a una temperatura caliente, de lo contrario pueden entregarse los diferentes pedidos en un rango un poco mayor que a la zona del restaurante.

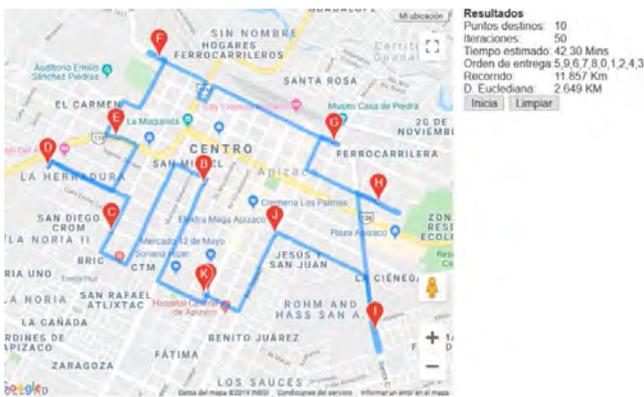


Figura 5. Ejecución del optimizador de rutas con algoritmo PSO (9 puntos de entrega).

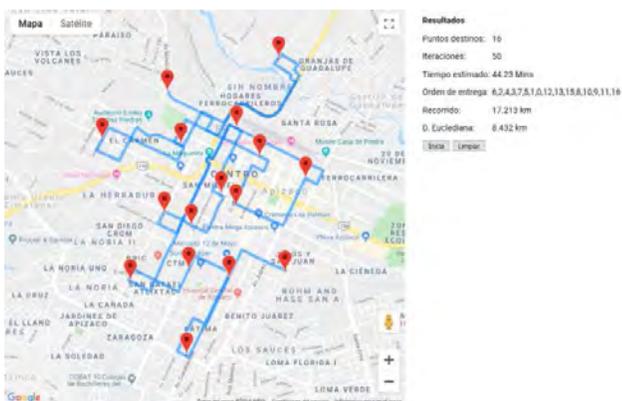


Figura 6. Ejecución del optimizador de rutas con algoritmo PSO (15 puntos de entrega).

Para poner a prueba la eficacia del algoritmo PSO se utilizó un ejemplo con 15 puntos de entrega en un radio no mayor a 8 kilómetros de donde se encuentre

ubicado el restaurante. En este caso no se rebasan los 8 kilómetros de radio. Recordemos que en el mapa se visualizan 16 puntos (ver Figura 6) ya que uno de ellos es el restaurante, donde tiene que partir y regresar. Una vez teniendo los resultados, se analizó el tiempo de entrega de reparto que supera los 40 minutos de entrega, con un recorrido de 17 km. De esta manera se tiene un aproximado de cuanto combustible gastará el repartidor, así se administran mejor los gastos y se reducen los costos de combustible. Al usar el API de Google se encontró que después de 15 puntos la herramienta deja de poner letras en los puntos para que sea más fácil de ubicar cada lugar. Es por eso que las figuras 8 y 9 no muestran alguna etiqueta encima del punto rojo.

Por último, se realizó un experimento con un número de 30 repartos. En este punto las entregas ya cumplen un tiempo prolongado, por el número de estas por un solo repartidor, en este caso al ser una optimización para servicio de comida es demasiado tiempo de entrega. No se recomienda este número de entregas para comida. Puede ser funcional para otro sector como el de mensajería, logística, por mencionar algunos. En esta prueba se tomaron los parámetros de 50 iteraciones con una velocidad al igual de las anteriores pruebas de 0.1, con un espacio de búsqueda de 30 partículas. La prueba se realizó con el fin de demostrar un tiempo de entrega, conocer un aproximado de recorrido para el repartidor. En el mapa se presenta (ver Figura 7) una posible entrega, pero esto puede variar dependiendo de las calles o bloqueos inesperados.

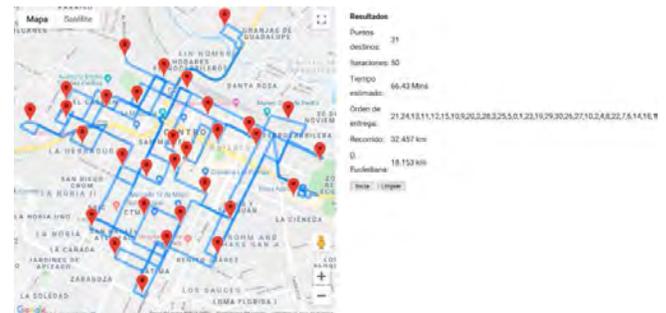


Figura 7. Ejecución del optimizador de rutas con algoritmo PSO (30 puntos de entrega).

Una vez realizadas las pruebas ejecutadas se realizó una tabla de resultados (ver Figura 8), con el fin de proporcionar un conjunto de datos para observar los diferentes tiempos de entrega. Obviamente entre más pedidos a entregar será más tardado el recorrido. Considerando el tiempo de ejecución puede variar por la velocidad de internet y los puntos de entrega. Con pocas entregas se ejecuta de manera considerablemente rápido, en esto también influyen mucho los destinos, las iteraciones y la velocidad de la partícula.

Datos obtenidos							
	Orden 1	Orden 2	Orden 3	Orden 4	Orden 5	Orden 6	Orden 7
Tiempo en ejecución(seg.)	5.07	6.96	8.18	10.01	11.38	13.45	16.1
Destinos	1	2	3	4	5	6	7
Iteraciones	50	50	50	50	50	50	50
Pbest	5.3636	0.00134	0.003673	0.054951	0.043359	0.0032	0.062027
Partículas	50	50	50	50	50	50	50
Velocidad	0.1	0.1	0.1	0.1	0.1	0.1	0.1
Tiempo de recorrido (Mins)	10.34	15.82	20.33	25.43	28.34	31.13	34.21

Figura 8. Tabla de resultados de 1 a 7 órdenes.

Todas las pruebas de la orden 1 a la 15 utilizaron el mismo número de iteraciones, la misma velocidad, con el mismo número de partículas.

En la Figura 9 se muestra la tabla de resultados de las órdenes 8 hasta la 30, destacando un tiempo de entrega exponencial. Aumentando la velocidad de la partícula en las órdenes 20 y 30 ya que se considera un mayor número de pedidos, también incrementando el número de iteraciones a 100.

Datos obtenidos						
	Orden 8	Orden 9	Orden 10	Orden 15	Orden 20	Orden 30
Tiempo en ejecución(seg.)	19.21	22.47	26.17	32.42	54.35	80.32
Destinos	8	9	10	15	20	30
Iteraciones	50	50	50	50	100	100
Pbest	0.14728	0.27539	0.14629	0.83444	1.8858	7.3331
Partículas	50	50	50	50	100	100
Velocidad	0.1	0.1	0.1	0.1	0.2	0.2
Tiempo de recorrido (Mins)	37.9	41.43	44.08	59.32	90.42	131.02

Figura 9. Tabla de resultados de 8 a 30 órdenes.

Cabe mencionar que puede variar el tiempo de ejecución si se modifican ciertos valores que hagan más robusto el proceso, por ejemplo, si se cambia el tamaño de partículas a una cantidad mayor puede demorar un poco más ya que este algoritmo suele ser exponencial.

Conclusiones

Este artículo propone el desarrollo de un módulo para una aplicación móvil con el fin de optimizar rutas de entrega de comida al sector restaurantero a través de un algoritmo de optimización con enjambre de partículas, utilizando el API de Google maps, con el fin de proponer a los restaurantes una ruta donde les permita hacer diferentes entregas de pedidos en un solo viaje por repartidor. Dentro de los resultados obtenidos se puede concluir la buena funcionabilidad del módulo. La eficacia y rapidez del algoritmo puede verse afectado en el dispositivo en el cual se realice la ejecución del PSO.

Para reducir la varianza y la naturaleza estocástica del PSO, y así acelerar el proceso de convergencia hacia una solución óptima, se puede adaptar el modelo Kriging al PSO como se propone en [10]. Sin embargo, el acoplamiento del modelo Kriging con el PSO puede fallar si se tienen dos o más distancias de entrega muy cercanas la una de la otra en la ruta del repartidor.

De igual manera, el beneficio de esta implementación busca la adaptabilidad de los diferentes restaurantes que ofrezcan el servicio de entregas a domicilio. El algoritmo PSO puede explotarse y superarse con implementaciones que busquen una mejora del algoritmo mediante las combinaciones de otros métodos como es la lógica difusa tipo I o tipo II. Como trabajo a futuro se pretende la implementación de lógica difusa y el algoritmo PSO aplicando un lenguaje natural, brindando a cada restaurante variables de que tan factible es la entrega a ciertos domicilios y a cuáles no es tan factible dependiendo de la distancia, puntos de entrega, pedidos, por mencionar algunos.*

REFERENCIAS

1. Food Express. (2019). Aplicacion para entrega de comida. Recuperado 14 de octubre de 2019, de <https://foodexpress.com.mx/index.xhtml>.
2. Choong, S. S., Wong, L.-P., y Lim, C. P. (2017). An artificial bee colony algorithm with a modified choice function for the Traveling Salesman Problem. *Swarm and Evolutionary Computation*, 44, 622-635.
3. Huang, Y.-H., Blazquez, C. A., Huang, S.-H., Paredes- Belmar, G., y Latorre-Nuñez, G. (2018). Solving the Feeder Vehicle Routing Problem Using Ant Colony Optimization. *Computers & Industrial Engineering*, 127, 520-235.
4. Li, Y., Soleimani, H., y Zohal, M. (2019). An improved ant colony optimization algorithm for the multi-depot green vehicle routing problem with multiple objectives. *Journal of Cleaner Production*, 227, 1161-1172.
5. Gu, Z., Zhu, Y., Wang, Y., Du, X., Guizani, M., y Tian, Z. (2020). Applying artificial bee colony algorithm to the multi-depot vehicle routing problem. *Software: Practice and Experience*.
6. SinDelantalMx. (2019). Aplicacion para entrega de comida. Recuperado 21 de septiembre de 2019, de <https://www.sindelantal.mx>.
7. Stockdale, M. (2011). El problema del viajante: un algoritmo heurístico y una aplicación. Tesis de licenciatura, Licenciada en ciencias exactas y naturales. Universidad de Buenos Aires, Argentina.
8. Kennedy, J. (1995). Particle swarm optimization. En *Proceedings of ICNN'95 - International Conference on Neural Net-works*. Perth, WA, Australia. 1942-1948.
9. Fontana, R. (2004). Recent system applications of short-pulse ultra-wideband (uwb) technology. *IEEE Transactions on Microwave Theory and Techniques*, 52, 2087-2104.
10. Zhang, Y., An, Z., y Zhou, J. (2010). Optimization Design of Flash Structure for Forging Die Based on Kriging-PSO Strategy. En *International Conference in Swarm Intelligence*, 373-381.

SOBRE LOS AUTORES



Sergio Soto Rodríguez es estudiante de posgrado en el Departamento de Estudios de Posgrado e Investigación en el Instituto Tecnológico de Apizaco. Sus intereses de investigación incluyen inteligencia artificial y algoritmos de optimización.



Edmundo Bonilla Huerta recibe el grado de licenciatura y Maestría en Ciencias Computacionales en el Instituto Tecnológico de Apizaco, Tlax. México, en 1998, y el grado de Doctor en Informática por la Université d'Angers. Campus Belle Beile, Francia en 2008. Sus principales áreas de investigación incluyen: Soft Computing, Visión Artificial, Minería de Datos y Aprendizaje Automático.



Alberto Portilla Flores es responsable del Área de Investigación e Innovación en SmartSoft, y Profesor en la FCBIyT-UATx, Doctor en Informática por la Universidad de Grenoble, Francia, Doctor en Ciencias de la Computación Cum Laude por la Fundación Universidad de las Américas-Puebla, México y PosDoc obtenido en el French Mexican Laboratory of Informatics and Automatic Control (LAFMIA UMI-3175), ha publicado artículos arbitrados a nivel nacional e internacional en el área de ingeniería de software, cómputo en la nube, sistemas transaccionales y cómputo orientado a servicios.



J. Crispín Hernández Hernández obtuvo el título de Lic. en Informática por el Instituto Tecnológico de Apizaco (ITA), Apizaco Tla., México, en 1993. El grado de M. en C. en Ciencias Computacionales por el ITA, Apizaco, Tlax., México, en 1998. Y el grado de Doctor en Informática por la Université d'Angers. Campus Belle Beile, Francia en 2008. Sus principales áreas de investigación incluyen: meta-heurística optimización, bioinformática, biomedicina, lógica difusa, visión por computadora y sistemas de recomendación.



Oscar Atriano Ponce es Director general de SmartSoft y presidente del Clúster de Tecnologías de Información de Tlaxcala, Maestro en Ciencias por la Universidad de las Américas-Puebla, Ingeniero en Computación por la Facultad de Ciencias Básicas, Ingeniería y Tecnología de la Universidad Autónoma de Tlaxcala. Tiene más de 20 años de experiencia en los campos docentes y productivos en el área de Ingeniería de Software e Innovación.

Swarm Intelligence es un paradigma inteligente distribuido que permite resolver problemas de optimización, originalmente fue inspirado en los ejemplos biológicos de fenómenos de enjambre, bandada y pastoreo en animales vertebrados.



Imagen tomada de: <https://www.stratio.com/>

IA & Educación

Julieta Noguez, Lucía Barrón y Yasmín Hernández
iaeducacion@komputersapiens.org

Los Ambientes Adaptativos para la enseñanza y la evaluación en línea como herramientas de soporte para la educación durante el distanciamiento social

Ante la pandemia actual por el brote del virus SARS-CoV-2, en nuestro país las instituciones educativas de todos los niveles adoptaron la estrategia de intercambiar los cursos presenciales por cursos en línea para enfrentar la crisis y acatar las medidas del distanciamiento social. A pesar de este esfuerzo, las clases sincrónicas, clases remotas prolongadas y los documentos compartidos, no han sido suficientes para cubrir las expectativas de las nuevas generaciones que han crecido en un entorno digital. De acuerdo al estudio de Simpson Scarborough, realizado en los meses de marzo y abril de 2020 en Estados Unidos a estudiantes de nivel licenciatura, el 70 % de los estudiantes dice que el aprendizaje en línea es deficiente con respecto a la instrucción presencial. Además, ante la opción de graduarse en línea o regresar a su campus y graduarse en persona, el 85 % prefiere la segunda opción (Simpson Scarborough, 2020).

Cuando hablamos de educación en línea, la idea central es utilizar de una manera creativa e ingeniosa las tecnologías digitales para así transformar la experiencia de aprendizaje. Sin embargo, durante este tiempo de crisis ha habido todo tipo de problemas, desde las capacidades de internet y ancho de banda disponible para las clases, hasta clases aburridas y con poca interacción, en parte, debido al poco tiempo que han tenido los profesores para preparar y adaptar sus materiales a esta modalidad de enseñanza tan retardadora.

Algunas instituciones han desarrollado y aplicado con éxito una gran cantidad de herramientas tecnológicas para lograr cursos estimulantes. Por ejemplo, el Tecnológico de Monterrey promueve el uso de Internet de las cosas, Laboratorios Remotos, Lectores biométricos, Realidad aumentada, Realidad mixta, Realidad virtual, Tele-presencia Robótica, Video o Recorridos 360, entre otros (Tecnológico de Monterrey, 2020). Adicionalmente, se han formado comunidades de aprendizaje y fondos de apoyo económico para la innovación de la enseñanza en línea como el Fondo Novus (Fondo Novus, 2020) cuyo objetivo es el de promover las mejores prácticas en el uso de la tecnología para este fin.

También es importante mencionar el impacto que ha significado para los profesores el esfuerzo de desarrollar y adaptar los materiales correspondientes. Antes de la pan-

demia, en enero de 2020, se publicó un estudio de Reino Unido donde encuestaron a 5888 académicos en todo el sector de Educación Superior y cuyos resultados se utilizaron para construir una tabla que mostrara la satisfacción del personal con la administración. Dicha tabla es una acusación severa al estado actual del sector de educación superior en este país con una puntuación media de satisfacción del 10.54 %. El estudio también recopiló datos cualitativos y se ampliaron los conocimientos utilizando estos datos. El análisis temático reveló siete temas principales: 1) dominio y la brutalidad de las métricas de evaluación de desempeño para los profesores; 2) carga de trabajo excesiva; 3) gobernanza y rendición de cuentas; 4) cambios continuos; 5) proyectos de vanidad; 6) el académico silenciado y 7) trabajo y salud mental (Erickson, Hanna & Walker, 2020). Actualmente, la carga de trabajo del profesor se ha incrementado debido a la necesidad de adoptar el modelo de enseñanza y evaluación en línea.

Para reducir la carga de trabajo del profesor al impartir sus clases en línea, es indudable que la Inteligencia Artificial y las tecnologías digitales para la enseñanza y evaluación pueden ser una solución importante al problema. Con estas herramientas se puede lograr la adaptación del contenido y los materiales educativos para los estudiantes de acuerdo con sus necesidades de aprendizaje, teniendo en cuenta sus rasgos cognitivos y afectivos y sus conocimientos previos. Los entornos de evaluación en línea ajustan las preguntas que se presentan al alumno para que su aprendizaje sea más eficiente y efectivo, omitiendo las preguntas que el alumno ya conoce y dirigiéndolo a las áreas que necesita reforzar.

Las pruebas adaptativas computarizadas (Computer Adaptive Test, CAT), se ajustan automáticamente al nivel de competencia de los examinados. Actualmente, casi todos los programas de prueba a gran escala en el mundo occidental ya son adaptativos o están en proceso de serlo (Yeng, 2012), (Rezaie & Golshan, 2015), (Martin & Lazendic, 2018).

Existen otras soluciones de sistemas comerciales y empresas que brindan sistemas de evaluación en línea llamados "Smart Books". Estos brindan ayuda al estudiante para que pueda revisar contenidos, realizar ejercicios en línea, ser evaluados y recibir asistencia inmediata de manera complementaria a la labor del profesor. Sin embargo, algunos de ellos presentan los siguientes problemas (Beer & Wagner, 2011): a) el contenido se define de acuerdo con un libro de texto que no necesariamente sigue el programa del curso, b) algunos de estos sistemas

no le permiten al maestro usar sus propios recursos, es decir, el conjunto de problemas disponibles está cerrado, c) los maestros pueden agregar sus ejercicios, pero no estarán disponibles si la licencia de usuario de la academia, los estudiantes o el maestro ya no se paga, y d) la navegación de los entornos de evaluación no siempre es adaptativa.

También existen otros ambientes desarrollados por profesores-investigadores cuyas soluciones adaptativas integran incluso el modelado emocional del estudiante (Muñoz et al, 2016), (Cabada et al, 2018) pero que por diferentes razones su aplicación se ha quedado sólo a nivel institucional.

La crisis actual que enfrentan las instituciones educativas debido a la emergencia sanitaria nos ha demostrado que es tiempo de retomar estos esfuerzos para mejorar la educación en línea y ayudar a que la experiencia de interacción de los estudiantes con los sistemas de evaluación adaptativa sea retardadora, facilitando la labor del profesor, utilizando la gran cantidad de datos que se recopilan en ambientes de aprendizaje en línea, modelando características y reconociendo emociones de los estudiantes, mejorando el desempeño del alumno durante la interacción y finalmente explotando el potencial de las herramientas de Inteligencia Artificial disponibles, especialmente aquéllas que pertenecen al área de aprendizaje de máquina (también llamado aprendizaje automático).*

REFERENCIAS

Beer, W. & Wagner, A. (2011). Smart books: adding context-awareness and interaction to electronic books. Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia. ACM New York. ISBN: 978-1-4503-0785-7. DOI: 10.1145/2095697.2095740. 218-222

Erickson M., Hanna P. & Walker C. (2020). The UK higher education senior management survey: a stactivist response to managerialist governance. *Studies in Higher Education*. DOI: 10.1080/03075079.2020.1712693

Fondo Novus (2020). Fondo NOVUS para la innovación educativa. Tecnológico de Monterrey. Consultado el 20 de mayo de 2020. <https://novus.itesm.mx/>

Martin, A. J. & Lazendic, G. (2018). Computer-adaptive testing: Implications for students' achievement, motivation, engagement, and subjective test experience. *Journal of Educational Psychology*, 110(1), 27.

Muñoz, K., Noguez, J., Neri, L., Kevitt, P. M., & Lunney, T. (2016). A Computational Model of Learners Achievement Emotions Using Control-Value Theory. *Journal of Educational Technology & Society*, 19 (2), 42–56. ISSN: 1436-4522. <http://www.ifets.info/issues.php?show=current>.

Rezaie, M. & Golshan, M. (2015). Computer adaptive test (CAT): Advantages and limitations. *International Journal of Educational Investigations*, 2, 128-137.

Scarborough, S. (2020). Higher Ed and COVID-19. April Replication of the National Student Survey. Consultado el 22 de mayo de 2020. <https://cdn2.hubspot.net/hubfs/4254080/The%20April%20Replication%20of%20the%20National%20Student%20Survey%20by%20SimpsonScarborough.pdf>

Tecnológico de Monterrey. (2020). Transforma tu curso a flexible y digital para la continuidad académica. Tecnológico de Monterrey. Informe del 14 de abril de 2020. Consultado el 26 de mayo de 2020. <https://innovacioneducativa.tec.mx/mosaicotec/2020/04/14/transforma-tu-curso-a-flexible-y-digital/>

Yeng Y. (2012). Book Review of Van der Linden, W. J., & Glas, C. A. W. (Eds.) (2010). *Elements of Adaptive Testing*. New York, NY. Springer. *Journal of Educational Measurement* Spring 2012, 49(1), 116–119. Consultado el 22 de mayo de 2020. <http://web.a.ebscohost.com/ehost/pdfviewer/pdfviewer?vid=2&sid=3d77beb2-ce4d-4c3c-9eb6-c869bdi03e41%40sessionmgr4007>.

Zatarain-Cabada R., Barrón-Estrada M.L., González Hernández F., Orama Bustillos R., Reyes-García C.A. (2018). An affective and Web 3.0-based learning environment for a programming language. *Telematics and Informatics* 35 (3), 611-628. <https://doi.org/10.1016/j.tele.2017.03.005>.



Las instituciones educativas, el reto de la educación en línea y la explotación de las herramientas de Inteligencia Artificial.

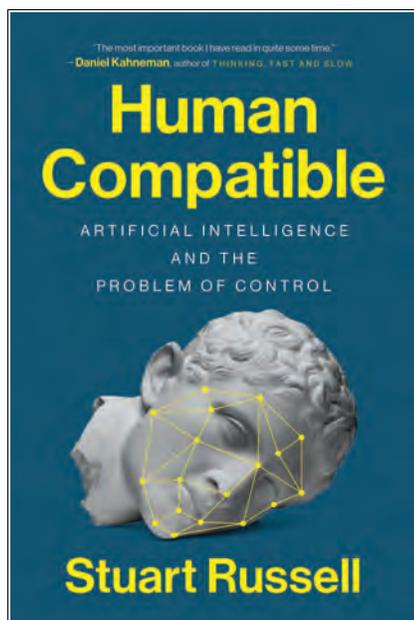
Deskubriendo Konocimiento

Alejandro Guerra Hernández y Leonardo Garrido
deskubriendokonocimiento@komputersapiens.org

Human Compatible: Artificial Intelligence and the problem of control

Dr. Alejandro Guerra-Hernández

Universidad Veracruzana
Centro de Investigación en Inteligencia Artificial



Portada de la revista.

Toda disciplina científica necesita una discusión seria y continua sobre su presente, pasado y futuro; y la Inteligencia Artificial (IA) no es la excepción. De hecho, Stuart Russell, autor del libro que reseñamos **, señala una doble justificación para ello —No solamente la IA se vuelve omnipresente a un ritmo acelerado en el presente, sino que es la tecnología dominante del futuro. La cuestión es que deberíamos tener un plan para contender con la posibilidad de desarrollar máquinas que excederán considerablemente nuestra capacidad para la toma de deci-

siones, y no tan solo no lo tenemos, sino que no discutimos esta cuestión con la seriedad y constancia que amerita. Aunque el tono de divulgación, llenó de anécdotas, referencias literarias y cinéfilas, se mantendrá a lo largo de todo el libro; no hay que caer en el engaño, se trata de un texto profundo que invita a los especialistas en IA a repensar los fundamentos de ésta y a reflexionar sobre ¿Qué sucederá si tenemos éxito?

El libro está organizado en tres partes. La primera, del primer capítulo al tercero, introduce el concepto de inteligencia en humanos y en computadoras. Se trata también de un ameno génesis de la IA para responder a la pregunta de ¿Cómo hemos llegamos hasta aquí? Aquí, en opinión de Russell, significa la adopción entusiasta de una concepción errónea: Las máquinas son inteligentes en la medida en que se puede esperar que sus acciones logren sus objetivos. Y como éstas no tienen objetivos, nosotros se los damos y listo, asunto arreglado. A esta idea presente no solo en IA, sino en economía, teoría del control, investigación de operaciones y la estadística, Russell le llama el modelo estándar.

El primer crítico del modelo estándar fue el mismísimo Norbert Wiener, quien impactado en 1960 por el programa jugador de damas

chinas de Arthur Samuel que aprendió a jugar mejor que su autor, escribió: Si usamos, para lograr nuestros propósitos, agentes mecánicos en cuya operación no podemos interferir efectivamente... mejor que estemos seguros de que el propósito puesto en la máquina es el que realmente deseamos. Russell llama a éste, el problema del Rey Midas y a su eventual solución agente benéfico. El capítulo dos discute el rol de la evolución en la inteligencia y el concepto de racionalidad para uno y dos agentes. Todo esto para profundizar en la crítica al modelo estándar y para preparar la presentación los agentes benéficos como una eventual alternativa.

Esta parte del libro cierra con una discusión sobre el futuro de la IA, desde sus aplicaciones eminentes en ecosistemas inteligentes con autos autónomos, asistentes personales, casas inteligentes y robots domésticos a nuestro servicio; hasta el eventual advenimiento de una IA superinteligente. En opinión de Russell, esta posibilidad es remota, pero Lord Rutherford estaba seguro de que la reacción nuclear en cadena era imposible y Szilard le demostró lo contrario unas cuantas horas más tarde. Un evento análogo en la IA podría llevarnos rápidamente a una superinteligencia. Para ello debe darse un rompimiento conceptual de esa talla. Algunos

**Russell, S. *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking. New York, NY, USA, 2019.

ejemplos considerados por Russell incluyen: resolver el problema del lenguaje y el sentido común; resolver cómo descubrimos acciones nuevas; y descubrir cómo administramos nuestra actividad mental. De igual importancia es tener claros los límites de tal superinteligencia, a saber: de ninguna manera una máquina así tendría un conocimiento completo y perfecto del presente y futuro. La física aplica también para tal máquina.

Como complemento a la primera parte del libro, se ofrecen cuatro apéndices en torno a los siguientes conceptos fundacionales de la IA: la búsqueda de soluciones; el conocimiento categórico y la lógica; la incertidumbre y la probabilidad; y el aprendizaje basado en la experiencia.

La segunda parte, del cuarto al sexto capítulo, aborda los problemas inherentes a la concepción de máquinas inteligentes, con énfasis en el problema del control ¿Cómo retendremos nuestro poder absoluto sobre máquinas que serán más poderosas que nosotros? Primero se revisan los malos usos que pueden dársele a la IA: vigilancia, persuasión y control de los ciudadanos; armas letales automáticas; la eliminación del trabajo tal y como lo co-

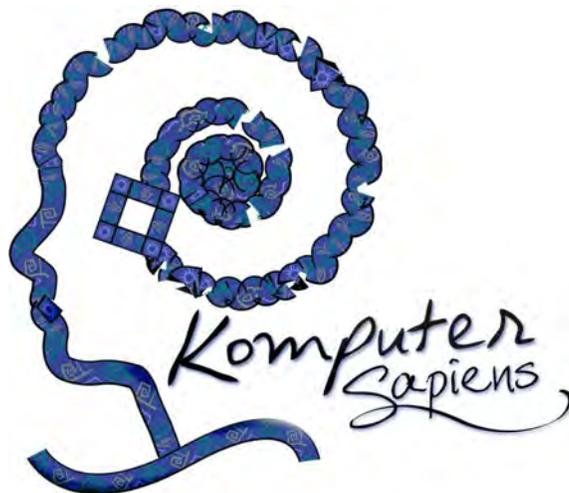
nocemos; y la usurpación de otros roles humanos.

El segundo problema con una IA demasiado inteligente es llamado por Russell el problema Gorila. Alrededor de hace unos diez millones de años los ancestros del gorila moderno dieron lugar (por accidente seguramente) al linaje genético de los modernos humanos ¿Cómo se sentirán los gorilas al respecto? ¿Cómo mantendremos nuestra supremacía y autonomía si creamos máquinas más inteligente que nosotros mismos? Por sorprendente que resulte, Russell llega a la conclusión de que el debate en IA en torno a esta cuestión es casi inexistente. O bien se niega que el problema exista: se le considera remoto, difícil de ocurrir y se llega a la conclusión de que es demasiado pronto para tal preocupación; o bien divergimos de su solución: no se puede controlar la investigación en IA; se puede, pero qué pasaría si...; o se prefiere guardar silencio. Este libro enfrenta precisamente este silencio.

La tercera parte, del séptimo capítulo al décimo, sugiere una posible solución al problema de control, máquinas benéficas que sigan estos tres principios: altruismo puro, el único objetivo de estas máquinas es maximizar la realización de

las preferencias humanas; modestia, inicialmente las máquinas tienen incertidumbre sobre tales preferencias; predicción, la última fuente de información para aprender sobre estas preferencias es la conducta humana. La propuesta en torno a estos principios será una IA benéfica demostrable. Una complicación de la propuesta, nosotros: heterogéneos, no siempre racionales, emocionales, envidiosos y multitudinarios humanos. ¿Cómo podría aprender una pobre IA acerca de nuestras preferencias? Una posibilidad es que el aprendizaje sea acerca de metapreferencias, preferencias acerca de las clases de cambio en las preferencias que son aceptables o inaceptables.

Si la IA benéfica demostrable puede solucionar el problema de la pérdida de control de nuestras máquinas superinteligentes, eso no significa que tengamos todo resuelto. El último capítulo discute la necesidad de contar con órganos de gobernanza para la IA a nivel mundial y la regulación para prevenir malos usos. El libro cierra con una reflexión abierta sobre ¿Qué sucederá cuando seamos tan dependientes de nuestras máquinas superinteligentes? El tiempo dirá.*





Durante la fase tres de la epidemia de COVID-19, el número de contagios crece de forma rápida, es por eso que debes quedarte en casa para reducir el número de casos que se pueden presentar.

El coronavirus es muy contagioso y, si hay más personas en las calles, seguramente habrá muchos enfermos que saturarían los hospitales.

¡No habría forma de ayudar a todos!

En cambio, si respetamos la sana distancia, habrá menos casos y más capacidad para que todos reciban atención.

Quédate en casa



Y no olvides:



Lavarte continuamente las manos con agua y jabón, al menos durante 20 segundos



Estornudar o toser en el ángulo interno del codo



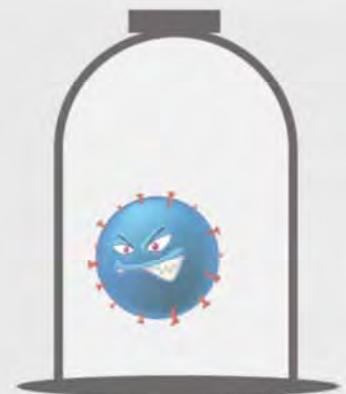
Mantener tu casa ventilada y limpia

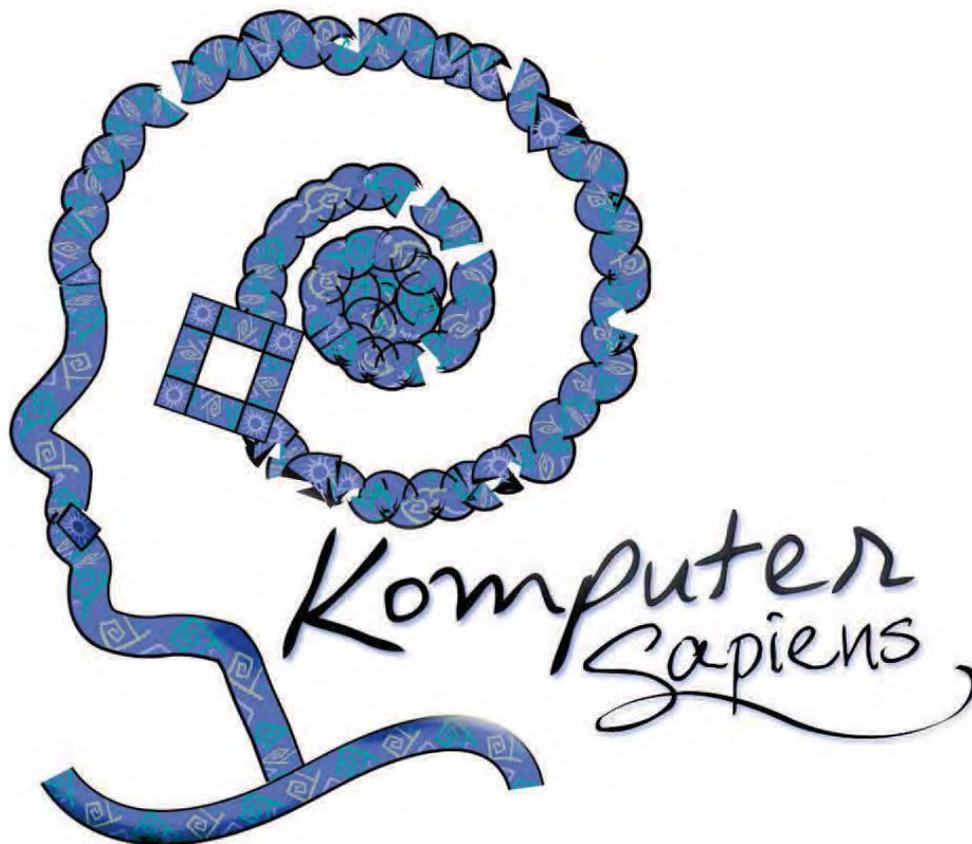


Desinfectar los utensilios y superficies de uso común

Quedándote en casa y cuidándonos unos a otros podremos salir adelante.

¡SI AISLAMOS AL VIRUS, LO VENCEREMOS!





¡Publique en Komputer Sapiens!



Komputer Sapiens solicita artículos de divulgación en todos los temas de Inteligencia Artificial, dirigidos a un amplio público conformado por estudiantes, académicos, empresarios, tomadores de decisiones y consultores. Komputer Sapiens es patrocinada por la SMIA, la Sociedad Mexicana de Inteligencia Artificial



www.smia.org.mx

Instrucciones para autores e información general: <http://www.komputersapiens.org>
Síguenos en las redes sociales: www.facebook.com/Komputer.Sapiens, twitter.com/KomputerSapiens

