

Heurística aplicada en inteligencia artificial, una revisión sistemática

Heuristics applied in artificial intelligence, a systematic review

Fabricio Marcillo¹, Marlon Sidney Castillo Anzules¹, Lucía Begnini²

¹Instituto Superior Universitario Japón, Santo Domingo de los Tsáchilas, Ecuador

fmarcillo@itsjapon.edu.ec

<https://orcid.org/0000-0003-2628-9167>

mcastilloa@itsjapon.edu.ec

<https://orcid.org/0009-0008-8031-3918>

²Instituto Superior Universitario Japón, Quito, Ecuador

lbegini@itsjapon.edu.ec

<https://orcid.org/0000-0001-5578-3004>

Correspondencia: fmarcillo@itsjapon.edu.ec

Recibido: 18/05/2024

Acceptado: 08/07/2024

Publicado: 07/08/2024

Resumen

Este estudio revisa sistemáticamente la aplicación de heurísticas en la inteligencia artificial (IA), destacando su impacto en la eficiencia, precisión y calidad del desarrollo de software. La revisión se centró en herramientas de automatización, análisis de código y chatbots de IA. Los resultados muestran que la IA, cuando se aplica mediante heurísticas, ha mejorado significativamente la productividad de los desarrolladores y ha impulsado la evolución de nuevas habilidades y roles profesionales. Sin embargo, también se identificaron desafíos éticos y de seguridad, incluyendo preocupaciones sobre la privacidad de los datos y la necesidad de transparencia en las decisiones de la IA. Además, se discute cómo las heurísticas afectan diversas etapas del ciclo de vida del desarrollo de software, desde la planificación hasta el mantenimiento. Se concluye que, aunque las heurísticas en IA ofrecen numerosas ventajas, es crucial abordar los desafíos para maximizar su efectividad y garantizar un uso responsable y ético de estas tecnologías avanzadas.

Palabras clave: Inteligencia artificial; heurística; desarrollo de software; automatización; análisis de código.

Abstract

This study systematically reviews the application of heuristics in artificial intelligence (AI), highlighting its impact on efficiency, accuracy, and software development quality. The review focused on automation tools, code analysis, and AI chatbots. The results show that AI, when applied through heuristics, has significantly improved developer productivity and driven the evolution of new skills and professional roles. However, ethical and security challenges, including data privacy concerns and the need for transparency in AI decisions, were also identified. Additionally, the study discusses how heuristics affect various stages of the software development lifecycle, from planning to maintenance. It concludes that while heuristics in AI offer numerous advantages, addressing the challenges is crucial to maximizing their effectiveness and ensuring responsible and ethical use of these advanced technologies.

Keywords: Artificial intelligence; heuristics; software development; automation; code analysis.

Introducción

La inteligencia artificial (IA) ha emergido como una tecnología transformadora en múltiples campos, y el desarrollo de software no es una excepción (Sabir et al., 2023). La capacidad de la IA para automatizar tareas, analizar grandes volúmenes de datos y aprender de manera autónoma ha cambiado la forma en que los desarrolladores crean, prueban y mantienen el software (Barjuei et al., 2024). Esta revisión sistemática tiene como objetivo analizar cómo las heurísticas se han integrado en la IA y su impacto en el desarrollo de software, evaluando su influencia en la productividad, las habilidades requeridas y los desafíos éticos y de seguridad (Barke et al., 2022).

Las heurísticas son técnicas que guían la resolución de problemas y la toma de decisiones mediante reglas prácticas y no óptimas, pero eficientes. En el contexto de la IA, las heurísticas se utilizan para mejorar la eficiencia de los algoritmos y reducir el

tiempo de procesamiento (Ogunsakin & Mehandjiev, 2022). Por ejemplo, en los algoritmos de búsqueda, las heurísticas ayudan a encontrar soluciones aceptables sin tener que explorar exhaustivamente todas las posibilidades (Chen et al., 2021).

La IA permite la automatización de tareas repetitivas y tediosas, lo que libera tiempo para que los desarrolladores se concentren en aspectos más creativos y complejos del desarrollo de software. Esto incluye el uso de asistentes de programación y sistemas de generación automática de código (Samsonovich et al., 2022). La automatización no solo reduce errores humanos, sino que también acelera los procesos de desarrollo, lo que resulta en una mayor eficiencia y productividad (Haass, 2022).

Las herramientas de análisis de código basadas en IA utilizan algoritmos de machine learning y heurísticas para analizar el código en busca de errores, vulnerabilidades y problemas de calidad (Son et al., 2021). Estas herramientas pueden aprender de grandes volúmenes de datos de código para proporcionar recomendaciones precisas y mejorar la calidad del software. El análisis de código automatizado puede identificar patrones complejos que podrían pasar desapercibidos para los desarrolladores humanos, mejorando así la robustez y seguridad del software (Bida & Aouat, 2021).

Los chatbots de IA, como los desarrollados por IBM Watson y Microsoft Azure, son utilizados para asistir a los desarrolladores en la resolución de problemas técnicos (Khan et al., 2024). Estos chatbots pueden responder preguntas frecuentes, proporcionar documentación relevante y sugerir soluciones basadas en problemas similares resueltos anteriormente (Verma et al., 2021). Las heurísticas juegan un papel crucial en la programación de estos chatbots, permitiendo respuestas más rápidas y precisas (Lagos & Pereira, 2024).

La incorporación de la IA en el desarrollo de software plantea importantes desafíos éticos y de seguridad (Shabbir & Anwer, 2018). Es esencial considerar la privacidad y seguridad de los datos, la transparencia y aplicabilidad de las decisiones de la IA, y las implicaciones de la IA en la fuerza laboral y las habilidades requeridas. Las heurísticas deben diseñarse de manera que se minimicen los sesgos y se garantice la equidad en la toma de decisiones automatizadas (Akashi & Yamaguchi, 2022).

Métodos

Este estudio se llevó a cabo mediante una revisión sistemática utilizando la declaración PRISMA (*Preferred Reporting Items for Systematic reviews and Meta-Analyses*). Se realizaron búsquedas exhaustivas en bases de datos académicas como *IEEE*, *Web of Science* y *Science Direct*, abarcando el periodo de 2021 a 2024. Las palabras clave utilizadas incluyeron: "inteligencia artificial", "heurística", "desarrollo de software", "machine learning" y "chatbots".

Criterios de inclusión y exclusión

Inclusión

- Estudios que analizan la implementación de la heurística en la IA aplicada al desarrollo de software.
- Filtro de tiempo: 2021-2024.
- Documentos en inglés y español.

Exclusión

- Artículos que no se centren en el desarrollo de software.
- Artículos que no proporcionen datos empíricos o análisis detallados.
- Documentos duplicados o no revisador por pares.

Se seleccionaron un total de 40 estudios relevantes que cumplieran con los criterios de inclusión. La información relevante fue extraída y sintetizada para analizar los principales hallazgos, beneficios, desafíos y el impacto de las heurísticas en la IA aplicada a diversas etapas del ciclo de vida del desarrollo de software.

Desarrollo y discusión

Los resultados mostraron que la implementación de herramientas de IA, como Copilot, mejoró significativamente la productividad en el desarrollo de software. La heurística aplicada permitió a los desarrolladores completar tareas un 55.8% más rápido, beneficiando especialmente a los desarrolladores con menos experiencia y mayores de 25 años. Esto indicó que la heurística no solo mejoraba la eficiencia sino también la

accesibilidad y la adaptabilidad del desarrollo de software a diferentes grupos demográficos (Toya et al., 2021).

El uso de estrategias educativas combinadas con herramientas de IA demostró ser efectivo en el aprendizaje de programación. La heurística aplicada en estos entornos educativos facilitó la enseñanza y permitió a los educadores tomar decisiones más informadas y oportunas. La combinación de aula invertida y aprendizaje colaborativo con apoyo de IA mostró cómo estas estrategias heurísticas podían mejorar la gestión y el proceso de aprendizaje (Reb et al., 2024).

El procesamiento del lenguaje natural, el aprendizaje automático y la visión por computadora fueron campos en los que la IA podía acelerar los procesos de desarrollo, reducir costos y mejorar la eficiencia mediante la automatización de tareas rutinarias. Sin embargo, la dependencia de la IA en estructuras humanas y su naturaleza repetitiva sugirieron que, aunque las heurísticas eran útiles, aún requerían supervisión y adaptación humana (Gandomi, 2024; Nejjarou et al., 2023).

El análisis de GitHub Copilot reveló que, aunque podía generar código eficazmente, planteaba preocupaciones sobre la calidad y seguridad del código, especialmente para usuarios no experimentados (Alexa et al., 2023). Estas limitaciones indicaron que la heurística implementada en Copilot era útil pero no suficiente para reemplazar completamente a los programadores humanos (Rajput et al., 2021).

La investigación mostró que ChatGPT mejoraba el rendimiento en la generación de código, superando incluso a ingenieros de software en ciertas tareas. La heurística aquí se observó en la eficiencia de ChatGPT para resolver problemas y generar soluciones más rápido. Sin embargo, la variabilidad en el tiempo de ejecución y la precisión sugirió que, aunque las heurísticas de ChatGPT eran potentes, aún necesitaban mejoras para igualar la competencia de programadores experimentados (Schramm et al., 2023).

La comparación entre ChatGPT y StackOverflow reveló que las respuestas de ChatGPT eran preferidas por su claridad y detalle. La heurística en este contexto se refirió a la capacidad de ChatGPT para proporcionar soluciones más comprensibles y completas. Sin embargo, en tareas de optimización, StackOverflow superó a ChatGPT debido a evaluaciones de rendimiento más rigurosas, lo que sugirió una limitación en la heurística de ChatGPT para tareas altamente específicas (Wu et al., 2024).

La implementación cuidadosa de agentes GPT podía cambiar el ciclo de desarrollo de software, adaptándose continuamente a las necesidades del proyecto. La heurística en la identificación de errores, comprensión contextual y generación de informes automatizados permitió una mejora en la eficiencia y precisión del desarrollo de software. No obstante, la necesidad de reevaluar roles y estrategias de colaboración sugirió que estas heurísticas aún requerían integración con la supervisión humana (Chuan et al., 2024).

La presentación de patrones de desarrollo de software rápidos mediante modelos de lenguaje grandes (LLM) como ChatGPT subrayó la utilidad de estas heurísticas en diversas áreas del desarrollo de software. Sin embargo, la dependencia de participación humana significativa para el uso efectivo de LLMs indicó que, aunque las heurísticas eran avanzadas, aún necesitaban la experiencia humana para maximizar su potencial (Górriz et al., 2023).

La aplicación de LLMs en tareas de SE, mejorando significativamente la revisión y retroalimentación del código, demostró la eficacia de estas heurísticas. Sin embargo, la necesidad de datos bien seleccionados y estrategias de optimización sugirió que estas heurísticas requerían una configuración precisa para ser efectivas (Kletzander & Musliu, 2024).

La generación de métodos aislados por modelos GPT-4 y GPT-3.5 fue eficaz, pero su precisión disminuyó en la generación de clases completas. Esta limitación heurística indicó que los modelos aún tenían desafíos en la gestión de tareas más complejas y dependientes del contexto.

La implementación de ChatGPT en el desarrollo de software mostró un impacto positivo en la eficiencia y el desarrollo de habilidades, aunque también se identificaron limitaciones en requisitos complejos y riesgos de dependencia. Estas observaciones sugirieron que las heurísticas de ChatGPT eran valiosas, pero aún necesitaban refinamiento para superar barreras creativas y prácticas (Hajipour et al., 2023).

ChatGPT pudo generar código funcional sin necesidad de muchos cambios, aunque los estudiantes tenían opiniones encontradas sobre su efectividad. La heurística aquí fue útil en el contexto educativo, pero su aplicación en situaciones prácticas requería más refinamiento (Cai et al., 2023).

La interacción con Copilot reveló dos modos: aceleración y exploración. Los programadores usaron diferentes heurísticas según su nivel de confianza con la tarea. Esto sugirió que la eficacia de Copilot dependía de la adaptabilidad de las heurísticas a las necesidades del usuario, destacando la importancia de mejorar el control y la precisión de la herramienta (Bilgin et al., 2006).

La revisión mostró que las heurísticas se han implementado en la IA para el desarrollo de software a través de diversas herramientas y tecnologías. Las herramientas de automatización, como los asistentes de programación y los sistemas de generación automática de código, han reducido significativamente el tiempo necesario para completar tareas repetitivas (Lissovoi et al., 2023).

Los sistemas de análisis de código impulsados por IA y heurísticas han mejorado la calidad del software al identificar errores y vulnerabilidades de manera más efectiva. Además, los chatbots de IA, como los desarrollados por IBM Watson y Microsoft Azure, han demostrado ser útiles para asistir a los desarrolladores en la resolución de problemas técnicos (Fišer et al., 2024).

El análisis de GitHub Copilot revela que, aunque puede generar código eficazmente, plantea preocupaciones sobre la calidad y seguridad del código, especialmente para usuarios no experimentados. Estas limitaciones indican que la heurística implementada en Copilot es útil pero no suficiente para reemplazar completamente a los programadores humanos (Pitakaso et al., 2024).

Los resultados indican que la implementación de herramientas de IA con heurísticas ha mejorado significativamente la productividad de los desarrolladores. Las herramientas de IA permiten completar tareas de codificación más rápido y con mayor precisión, lo que ha llevado a un aumento en la eficiencia y la calidad del código (García-Peñalvo et al., 2021; Özlü et al., 2022). Sin embargo, también se identificó una potencial sobredependencia de estas herramientas, lo que podría disminuir la capacidad de los desarrolladores para realizar tareas de manera independiente (Ramamoorthy & Syrotiuk, 2024).

La adopción de la IA con heurísticas ha cambiado las habilidades necesarias para los desarrolladores de software. Ahora, los desarrolladores deben tener una comprensión básica de los algoritmos de machine learning y las técnicas heurísticas. Esto ha llevado a

una mayor demanda de formación y educación continua en estas áreas. Además, han surgido nuevos roles especializados, como los ingenieros de machine learning y los científicos de datos, que se centran en el desarrollo e implementación de soluciones de IA (Muriyatmoko et al., 2024).

La incorporación de la IA en el desarrollo de software plantea importantes desafíos éticos y de seguridad. La privacidad y seguridad de los datos son preocupaciones críticas, ya que la IA requiere grandes volúmenes de datos para entrenar sus modelos (Platt et al., 2021; Rose et al., 2023). Es esencial implementar medidas robustas para proteger la información sensible y cumplir con las regulaciones de privacidad. Además, la transparencia y explicabilidad de las decisiones de la IA son fundamentales para garantizar la confianza de los usuarios y evitar sesgos inadvertidos (Baniecki & Biecek, 2024).

La IA ha mostrado su potencial para mejorar diversas etapas del ciclo de vida del desarrollo de software. En la planificación y diseño, las herramientas de IA pueden anticipar problemas y optimizar recursos (Ferreira et al., 2023). Durante la implementación, las herramientas de generación de código y los asistentes de IA facilitan el proceso de codificación, permitiendo a los desarrolladores centrarse en tareas más complejas y creativas. En el mantenimiento, los sistemas de monitorización basados en IA pueden detectar anomalías y alertar a los desarrolladores sobre posibles problemas antes de que se conviertan en incidentes críticos (Buscaldi et al., 2024).

Conclusiones

La inteligencia artificial, cuando se aplica mediante heurísticas, está transformando significativamente la industria del desarrollo de software, proporcionando beneficios en términos de productividad, calidad del código y eficiencia operativa. Sin embargo, también plantea nuevos desafíos que deben ser abordados de manera adecuada. Es fundamental continuar investigando para optimizar el uso de la IA y abordar las preocupaciones éticas y de seguridad identificadas. Las futuras investigaciones deberían centrarse en mejorar la transparencia y explicabilidad de las decisiones de la IA, así como en desarrollar estrategias para mitigar la brecha de habilidades en el campo del desarrollo de software. La evolución continua y la adaptación son claves para el éxito en la integración de la IA y las heurísticas en el desarrollo de software.

La revisión de estos estudios mostró que las heurísticas aplicadas en inteligencia artificial habían demostrado ser altamente efectivas en mejorar la eficiencia y productividad en el desarrollo de software. Sin embargo, también revelaron limitaciones y desafíos que debían abordarse para maximizar su potencial. La integración de IA en diversas etapas del desarrollo de software y la educación mostró beneficios significativos, pero aún requería la supervisión y adaptación humana para ser verdaderamente efectiva. Estos hallazgos subrayaron la necesidad de continuar investigando y refinando las heurísticas utilizadas en IA para aprovechar al máximo sus capacidades y minimizar sus limitaciones.

Referencias

- Akashi, K., & Yamaguchi, K. (2022). Independent Sets Extraction Graph Coloring Algorithm Using Beam Search. *Proceedings - 2022 IEEE/ACIS 24th International Winter Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, SNPD 2022*, 230–234. <https://doi.org/10.1109/SNPD54884.2022.10051806>
- Alexa, F., Otesteanu, M., Căleanu, C., Curiac, D.-I., Botez, R., Pasca, A.-G., Sferle, A.-T., Ivanciu, I.-A., & Dobrota, V. (2023). Efficient Network Slicing with SDN and Heuristic Algorithm for Low Latency Services in 5G/B5G Networks. *Sensors 2023, Vol. 23, Page 6053*, 23(13), 6053. <https://doi.org/10.3390/S23136053>
- Baniecki, H., & Biecek, P. (2024). Adversarial attacks and defenses in explainable artificial intelligence: A survey. *Information Fusion*, 107, 102303. <https://doi.org/10.1016/J.INFFUS.2024.102303>
- Barjuei, E. S., Capitanelli, A., Bertolucci, R., Courteille, E., Mastrogiovanni, F., & Maratea, M. (2024). Digital workflow for printability checking and prefabrication in robotic construction 3D printing based on Artificial Intelligence planning. *Engineering Applications of Artificial Intelligence*, 133, 108254. <https://doi.org/10.1016/J.ENGAPPAI.2024.108254>
- Barke, S., James, M. B., & Polikarpova, N. (2022). Grounded Copilot: How Programmers Interact with Code-Generating Models. *Proceedings of the ACM on Programming Languages*, 7(OOPSLA1). <https://doi.org/10.1145/3586030>

- Bida, I., & Aouat, S. (2021). Swarm Intelligence-based Decision Trees Induction for Classification - A Brief Analysis. *2020 2nd International Workshop on Human-Centric Smart Environments for Health and Well-Being, IHSH 2020*, 165–170. <https://doi.org/10.1109/IHSH51661.2021.9378746>
- Bilgin, B., Özcan, E., & Korkmaz, E. E. (2006). An experimental study on hyperheuristics and exam timetabling. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3867 LNCS, 394–412. https://doi.org/10.1007/978-3-540-77345-0_25
- Buscaldi, D., Dessí, D., Motta, E., Murgia, M., Osborne, F., & Reforgiato Recupero, D. (2024). Citation prediction by leveraging transformers and natural language processing heuristics. *Information Processing & Management*, 61(1), 103583. <https://doi.org/10.1016/J.IPM.2023.103583>
- Cai, W., Liu, Z., Zhang, M., & Wang, C. (2023). Cooperative Artificial Intelligence for underwater robotic swarm. *Robotics and Autonomous Systems*, 164, 104410. <https://doi.org/10.1016/J.ROBOT.2023.104410>
- Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H. P. de O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., ... Zaremba, W. (2021). *Evaluating Large Language Models Trained on Code*. <https://arxiv.org/abs/2107.03374v2>
- Chuan, C. H., Sun, R., Tian, S., & Tsai, W. H. S. (2024). EXplainable Artificial Intelligence (XAI) for facilitating recognition of algorithmic bias: An experiment from imposed users' perspectives. *Telematics and Informatics*, 91, 102135. <https://doi.org/10.1016/J.TELE.2024.102135>
- Ferreira, I. A., Bento, C., Mascarenhas, L., Pimentel, C., & Carvalho, H. (2023). A Heuristic Approach to Support Route Planning for Delivery and Installation of Furniture: A Case Study. *Applied Sciences 2023, Vol. 13, Page 3285, 13(5)*, 3285. <https://doi.org/10.3390/APP13053285>
- Fišer, D., Torralba, Á., & Hoffmann, J. (2024). Boosting optimal symbolic planning: Operator-potential heuristics. *Artificial Intelligence*, 104174. <https://doi.org/10.1016/J.ARTINT.2024.104174>

- Gandomi, A. H. (2024). *Evolutionary Computation for Intelligent Data Analytics*. 000011–000012. <https://doi.org/10.1109/CINTI59972.2023.10382125>
- García-Peñalvo, F. J., Vázquez-Ingelmo, A., García-Holgado, A., Sampedro-Gómez, J., Sánchez-Puente, A., Vicente-Palacios, V., Dorado-Díaz, P. I., & Sánchez, P. L. (2021). Application of artificial intelligence algorithms within the medical context for non-specialized users: The cartier-ia platform. *International Journal of Interactive Multimedia and Artificial Intelligence*, 6(6), 46–53. <https://doi.org/10.9781/IJIMAI.2021.05.005>
- Górriz, J. M., Álvarez-Illán, I., Álvarez-Marquina, A., Arco, J. E., Atzmueller, M., Ballarini, F., Barakova, E., Bologna, G., Bonomini, P., Castellanos-Dominguez, G., Castillo-Barnes, D., Cho, S. B., Contreras, R., Cuadra, J. M., Domínguez, E., Domínguez-Mateos, F., Duro, R. J., Elizondo, D., Fernández-Caballero, A., ... Ferrández-Vicente, J. M. (2023). Computational approaches to Explainable Artificial Intelligence: Advances in theory, applications and trends. *Information Fusion*, 100, 101945. <https://doi.org/10.1016/J.INFFUS.2023.101945>
- Haass, J. C. (2022). Cyber Threat Intelligence and Machine Learning. *Proceedings - 2022 4th International Conference on Transdisciplinary AI, TransAI 2022*, 156–159. <https://doi.org/10.1109/TRANSAT54797.2022.00033>
- Hajipour, V., Niaki, S. T. A., Tavana, M., Santos-Arteaga, F. J., & Hosseinzadeh, S. (2023). A comparative performance analysis of intelligence-based algorithms for optimizing competitive facility location problems. *Machine Learning with Applications*, 11, 100443. <https://doi.org/10.1016/J.MLWA.2022.100443>
- Khan, Z. A., Aziz, I. A., Osman, N. A. B., & Nabi, S. (2024). Parallel Enhanced Whale Optimization Algorithm for Independent Tasks Scheduling on Cloud Computing. *IEEE Access*, 12, 23529–23548. <https://doi.org/10.1109/ACCESS.2024.3364700>
- Kletzander, L., & Musliu, N. (2024). Hyper-heuristics for personnel scheduling domains. *Artificial Intelligence*, 104172. <https://doi.org/10.1016/J.ARTINT.2024.104172>
- Lagos, F., & Pereira, J. (2024). Multi-armed bandit-based hyper-heuristics for combinatorial optimization problems. *European Journal of Operational Research*, 312(1), 70–91. <https://doi.org/10.1016/J.EJOR.2023.06.016>
- Lissovoi, A., Oliveto, P. S., & Warwicker, J. A. (2023). When move acceptance selection hyper-heuristics outperform Metropolis and elitist evolutionary algorithms and

- when not. *Artificial Intelligence*, 314, 103804.
<https://doi.org/10.1016/J.ARTINT.2022.103804>
- Muriyatmoko, D., Djunaidy, A., & Muklason, A. (2024). Heuristics and Metaheuristics for Solving Capacitated Vehicle Routing Problem: An Algorithm Comparison. *Procedia Computer Science*, 234, 494–501.
<https://doi.org/10.1016/J.PROCS.2024.03.032>
- Nejjarou, O., Aqil, S., & Lahby, M. (2023). Improvement artificial bee colony meta-heuristic for solving manufacturing flow-shop scheduling problem under blocking constraint. *Proceedings - SITA 2023: 2023 14th International Conference on Intelligent Systems: Theories and Applications*.
<https://doi.org/10.1109/SITA60746.2023.10373714>
- Ogunsakin, R., & Mehandjiev, N. (2022). Towards Autonomous Production: Enhanced Meta-heuristics Algorithm. *Procedia Computer Science*, 200, 1575–1581.
<https://doi.org/10.1016/J.PROCS.2022.01.358>
- Özlü, İ. A., Baimakhanov, O., Saukhimov, A., & Ceylan, O. (2022). A Heuristic Methods-Based Power Distribution System Optimization Toolbox. *Algorithms*, 15(1). <https://doi.org/10.3390/A15010014>
- Pitakaso, R., Srichok, T., Khonjun, S., Golinska-Dawson, P., Sethanan, K., Nanthasamroeng, N., Gonwirat, S., Luesak, P., & Boonmee, C. (2024). Optimization-driven artificial intelligence-enhanced municipal waste classification system for disaster waste management. *Engineering Applications of Artificial Intelligence*, 133, 108614. <https://doi.org/10.1016/J.ENGAPPAL.2024.108614>
- Platt, I. A., Kannangara, C., Carson, J., & Tytherleigh, M. (2021). Heuristic assessment of psychological interventions in schools (HAPI Schools). *Psychology in the Schools*, 58(7), 1399–1415. <https://doi.org/10.1002/PITS.22527>
- Rajput, N. S., Banerjee, R., Sanghi, D., Santhanam, G., & Singhal, K. (2021). Swarm intelligence inspired meta-heuristics for solving multi-constraint QoS path problem in vehicular ad hoc networks. *Ad Hoc Networks*, 123, 102633.
<https://doi.org/10.1016/J.ADHOC.2021.102633>
- Ramamoorthy, M., & Syrotyuk, V. R. (2024). Learning heuristics for arc routing problems. *Intelligent Systems with Applications*, 21, 200300.
<https://doi.org/10.1016/J.ISWA.2023.200300>

- Reb, J., Luan, S., & Gigerenzer, G. (2024). Smart Management. *Smart Management*.
<https://doi.org/10.7551/MITPRESS/14939.001.0001>
- Rose, W. J., Bell, J. E., & Griffis, S. E. (2023). Inductive research in last-mile delivery routing: Introducing the Re-Gifting heuristic. *Journal of Business Logistics*, 44(1), 109–140. <https://doi.org/10.1111/JBL.12318>
- Sabir, Z., Asmara, A., Dehraj, S., Raja, M. A. Z., Altamirano, G. C., Salahshour, S., Sadat, R., & Ali, M. R. (2023). A mathematical model of coronavirus transmission by using the heuristic computing neural networks. *Engineering Analysis with Boundary Elements*, 146, 473–482.
<https://doi.org/10.1016/J.ENGANABOUND.2022.10.033>
- Samsonovich, A. V., Shumsky, S. A., Karpov, V. E., Kotov, A. A., & Kolonin, A. G. (2022). Key Advanced Research Initiative: A Manifesto for the New-Generation Artificial Intelligence. *Procedia Computer Science*, 213(C), 824–831.
<https://doi.org/10.1016/J.PROCS.2022.11.140>
- Schramm, S., Wehner, C., & Schmid, U. (2023). Comprehensible Artificial Intelligence on Knowledge Graphs: A survey. *Journal of Web Semantics*, 79, 100806.
<https://doi.org/10.1016/J.WEBSEM.2023.100806>
- Shabbir, J., & Anwer, T. (2018). *Artificial Intelligence and its Role in Near Future*.
<http://arxiv.org/abs/1804.01396>
- Son, N. T., Jaafar, J., Aziz, I. A., & Anh, B. N. (2021). Meta-Heuristic Algorithms for Learning Path Recommender at MOOC. *IEEE Access*, 9, 59093–59107.
<https://doi.org/10.1109/ACCESS.2021.3072222>
- Toya, T., Mizuno, K., & Koike, S. (2021). Adaptive Ant Colony Optimization with Several Pheromone Updates for Constraint Satisfaction Problems. *Proceedings - 2021 International Conference on Technologies and Applications of Artificial Intelligence, TAAI 2021*, 19–24. <https://doi.org/10.1109/TAAI54685.2021.00013>
- Verma, S., Sharma, R., Deb, S., & Maitra, D. (2021). Artificial intelligence in marketing: Systematic review and future research direction. *International Journal of Information Management Data Insights*, 1(1), 100002.
<https://doi.org/10.1016/J.IJIMEI.2020.100002>
- Wu, L., Chen, Z. F., & Tao, W. (2024). Instilling warmth in artificial intelligence? Examining publics' responses to AI-applied corporate ability and corporate social

responsibility practices. *Public Relations Review*, 50(1), 102426.
<https://doi.org/10.1016/J.PUBREV.2024.102426>

Los autores no tienen conflicto de interés que declarar. La investigación fue financiada por el Instituto Superior Universitario Japón y los autores.

Copyright (2024) © Fabricio Marcillo, Marlon Sidney Castillo Anzules, Lucía Begnini

Este texto está protegido bajo una licencia
[Creative Commons de Atribución Internacional 4.0](https://creativecommons.org/licenses/by/4.0/)

