0

ALGORITMOS EVOLUTIVOS I (CURSADO)

DOCENTE DEL CURSO

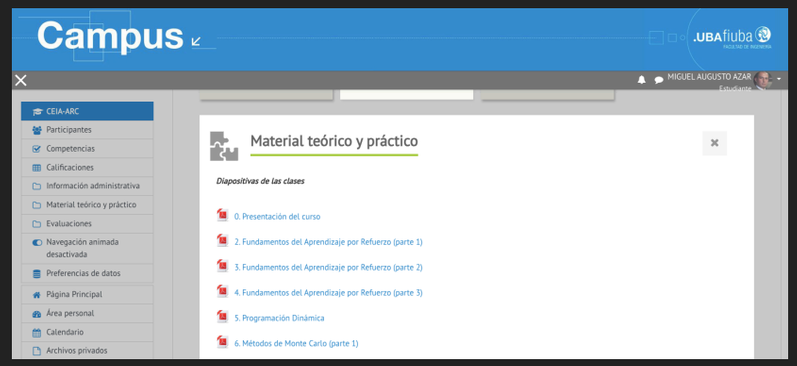
- Miguel Augusto Azar
- Ingeniero en Informática
- Especialista en Docencia Superior
- Docente e investigador
- Email: augusto.azar@gmail.com

RECURSOS VIRTUALES

- Campus virtual

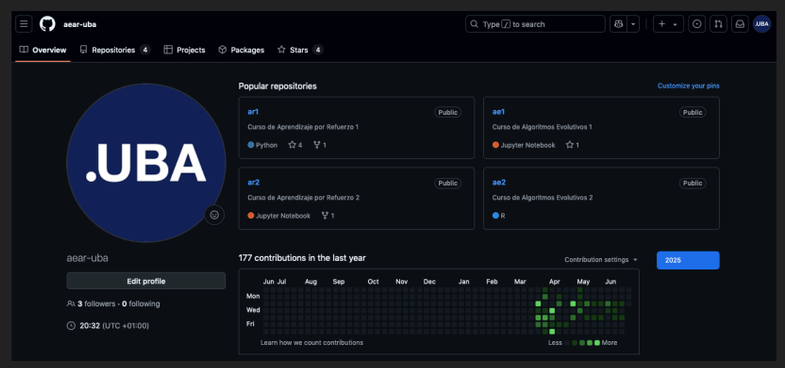
  https://campusposgrado.fi.uba.ar/course/view.php?id=420

  Automatriculación: ae1==2025



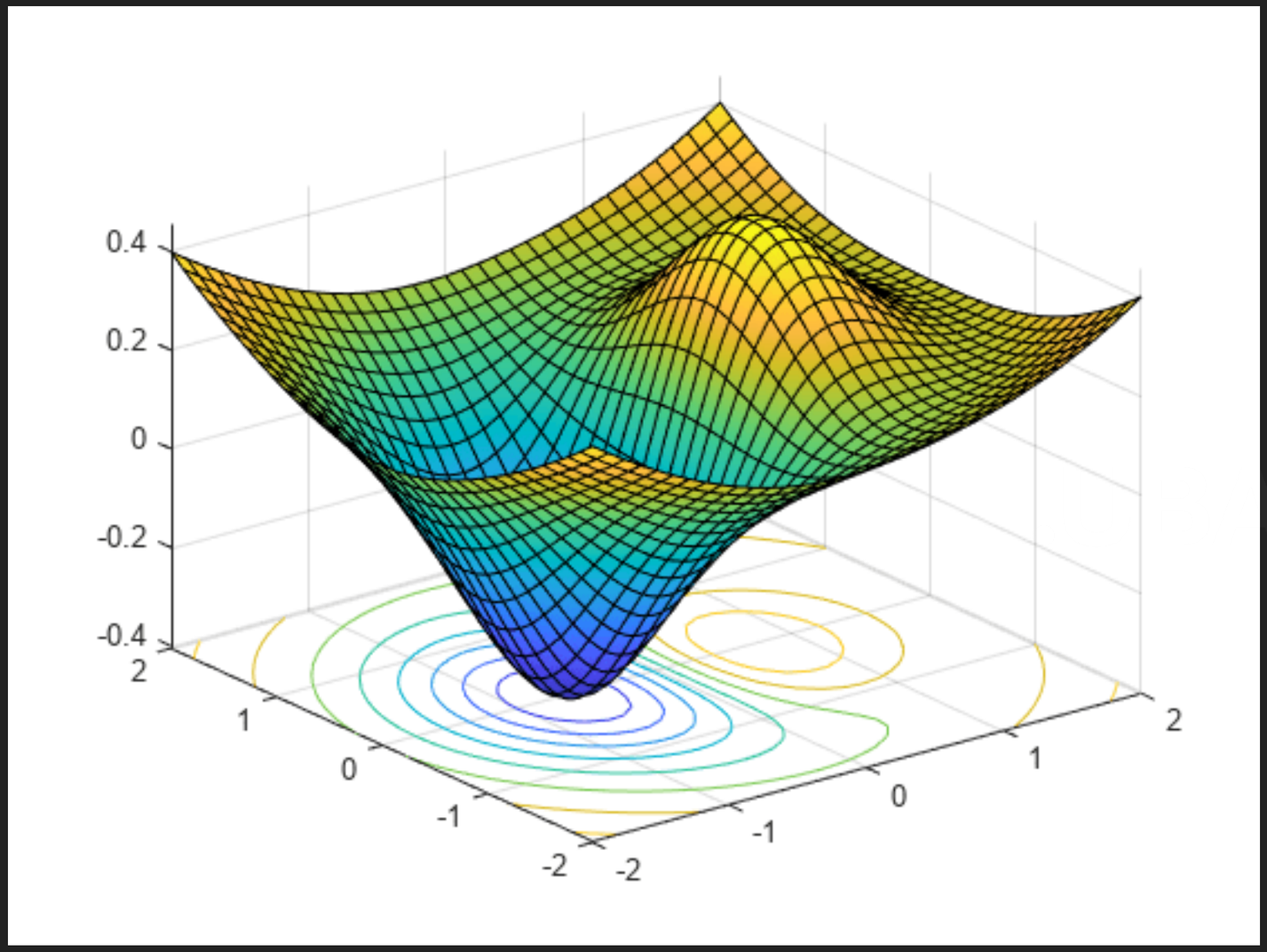
- Repositorio

  https://github.com/aear-uba/ae1/



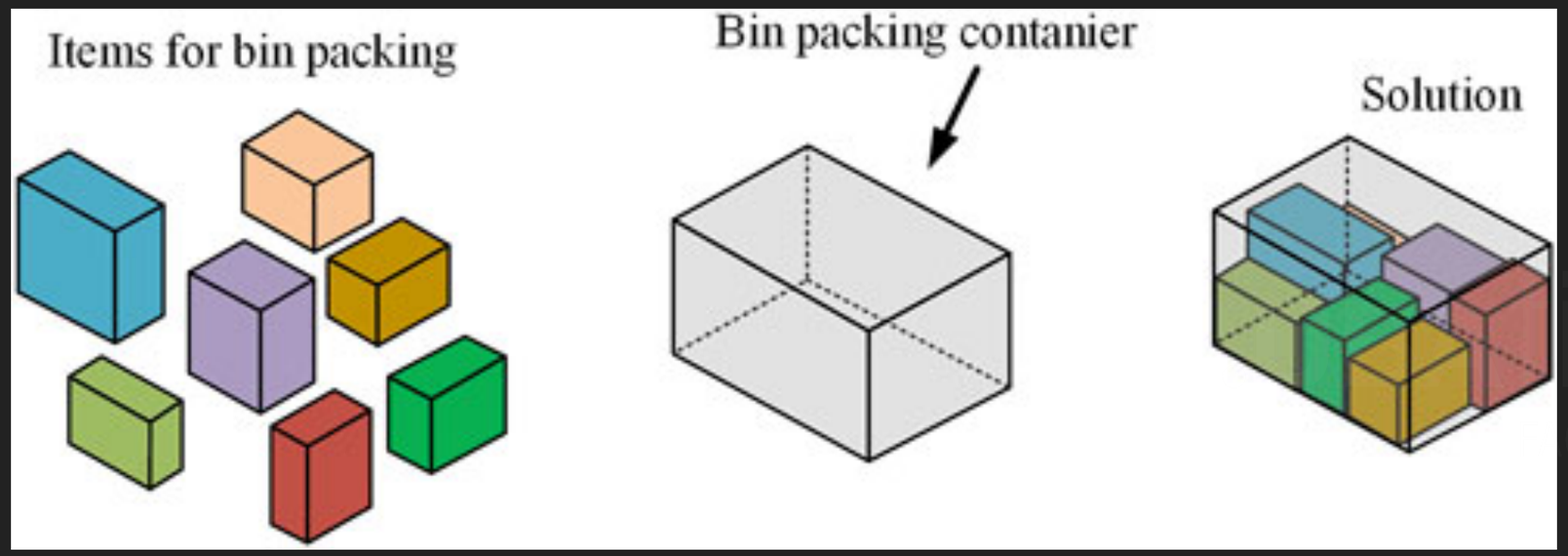
¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?



¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

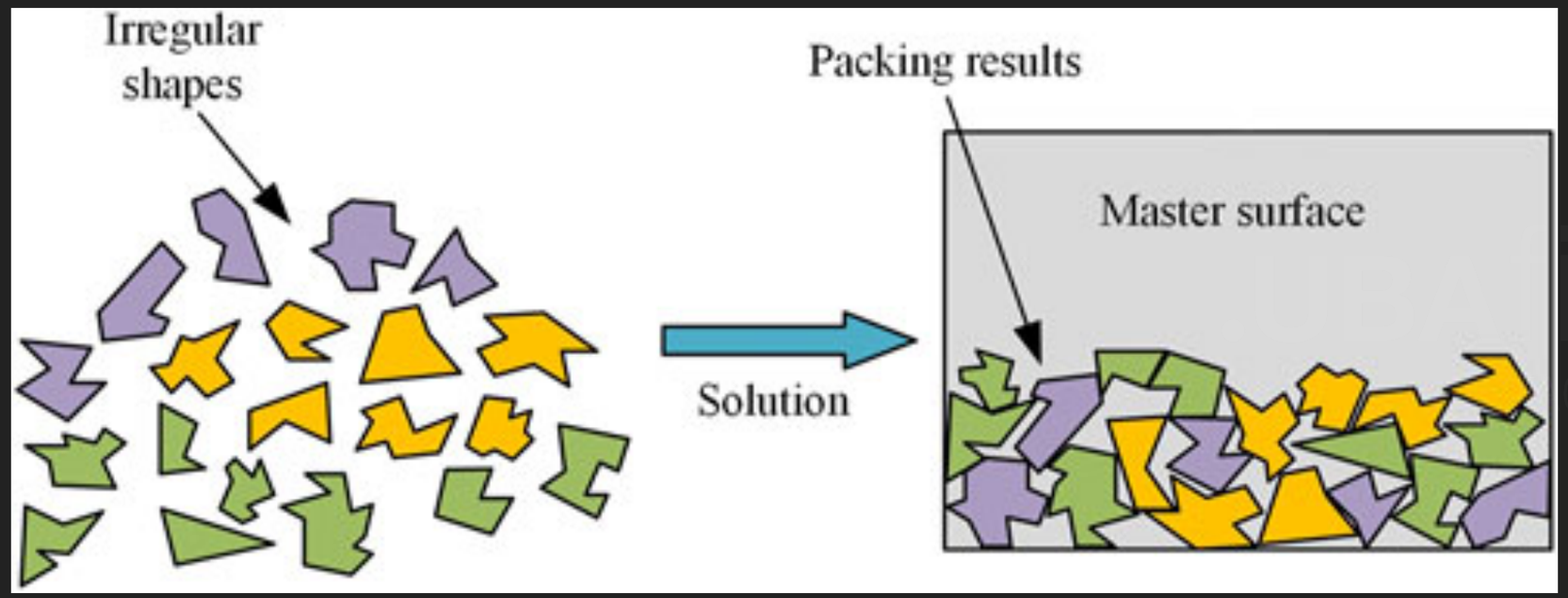
- Artículos a embalar en un contenedor



- 3D bin packing problem (set packing)

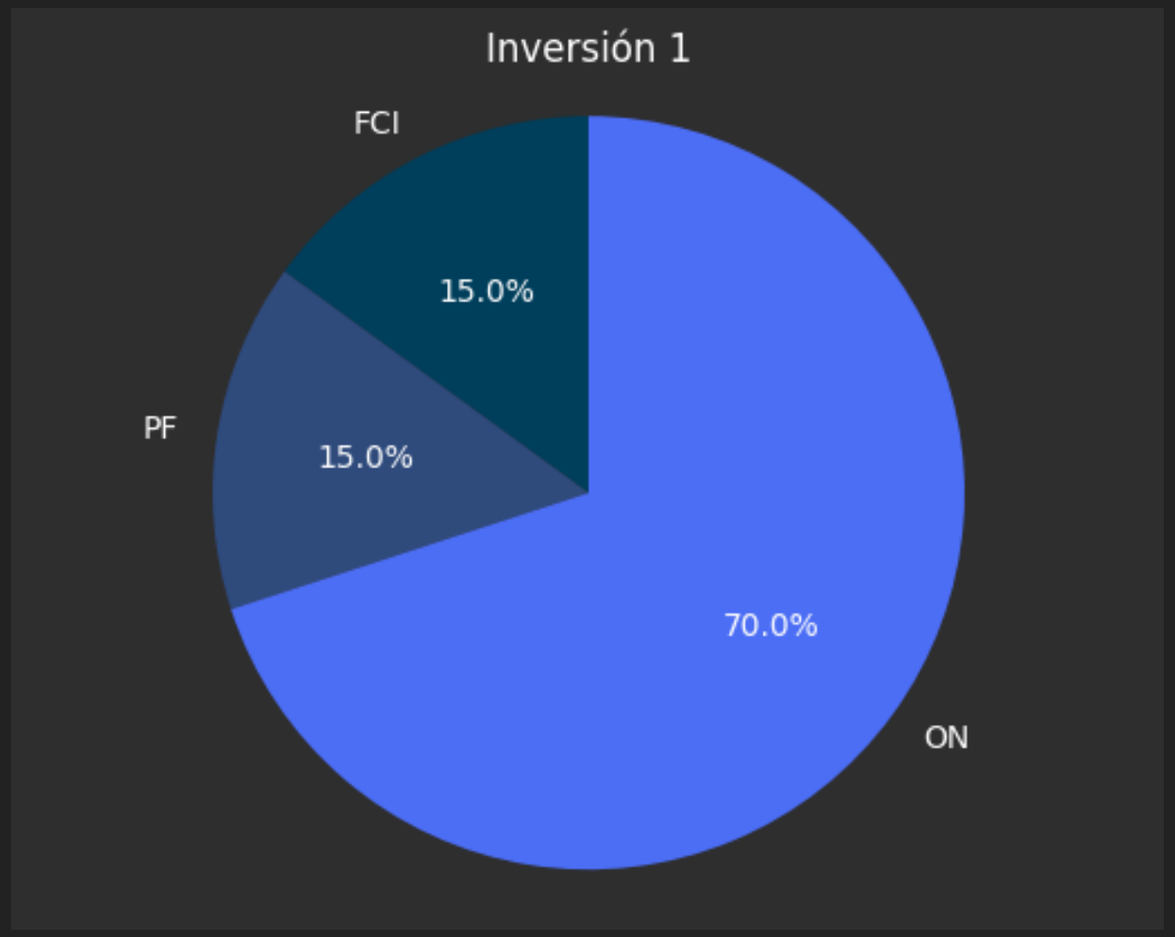
¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

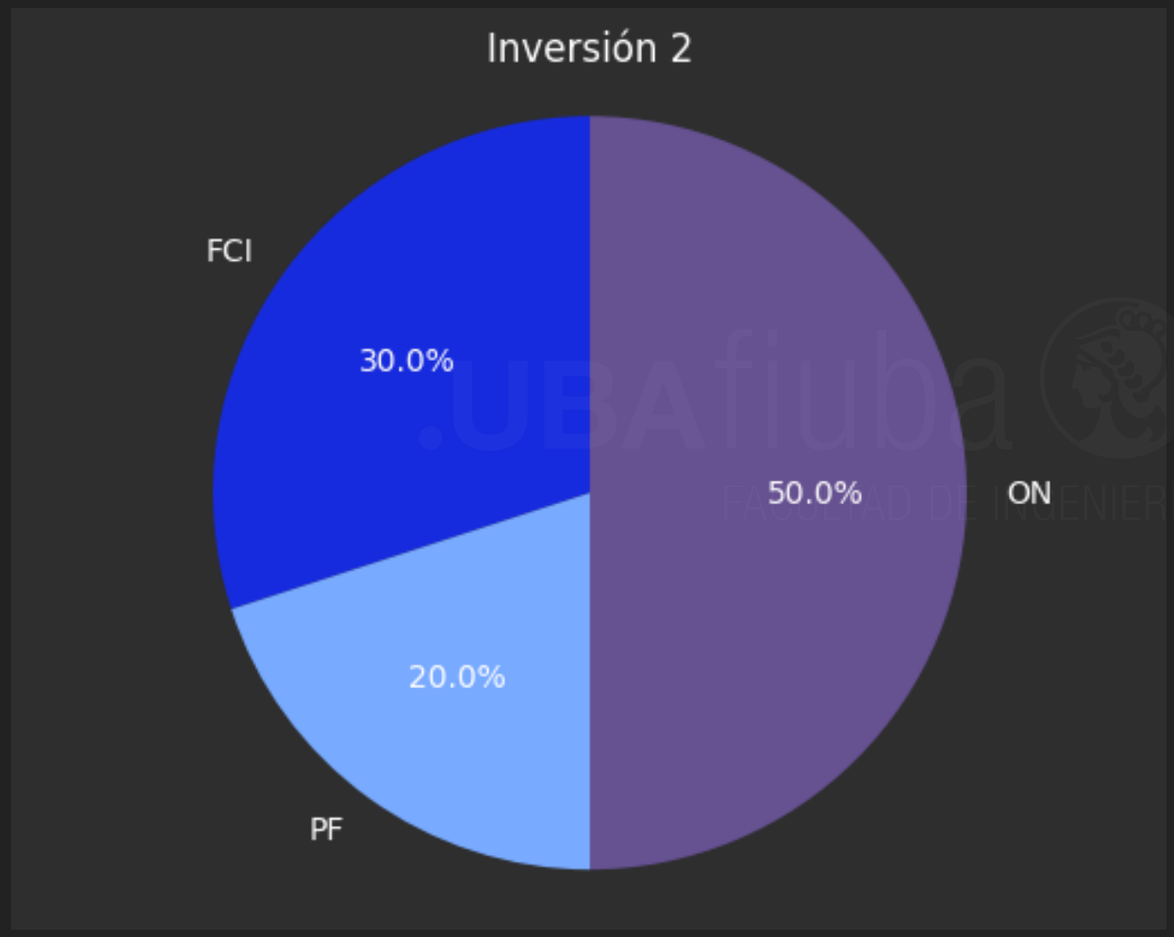
- Tela en una fábrica de prendas



¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

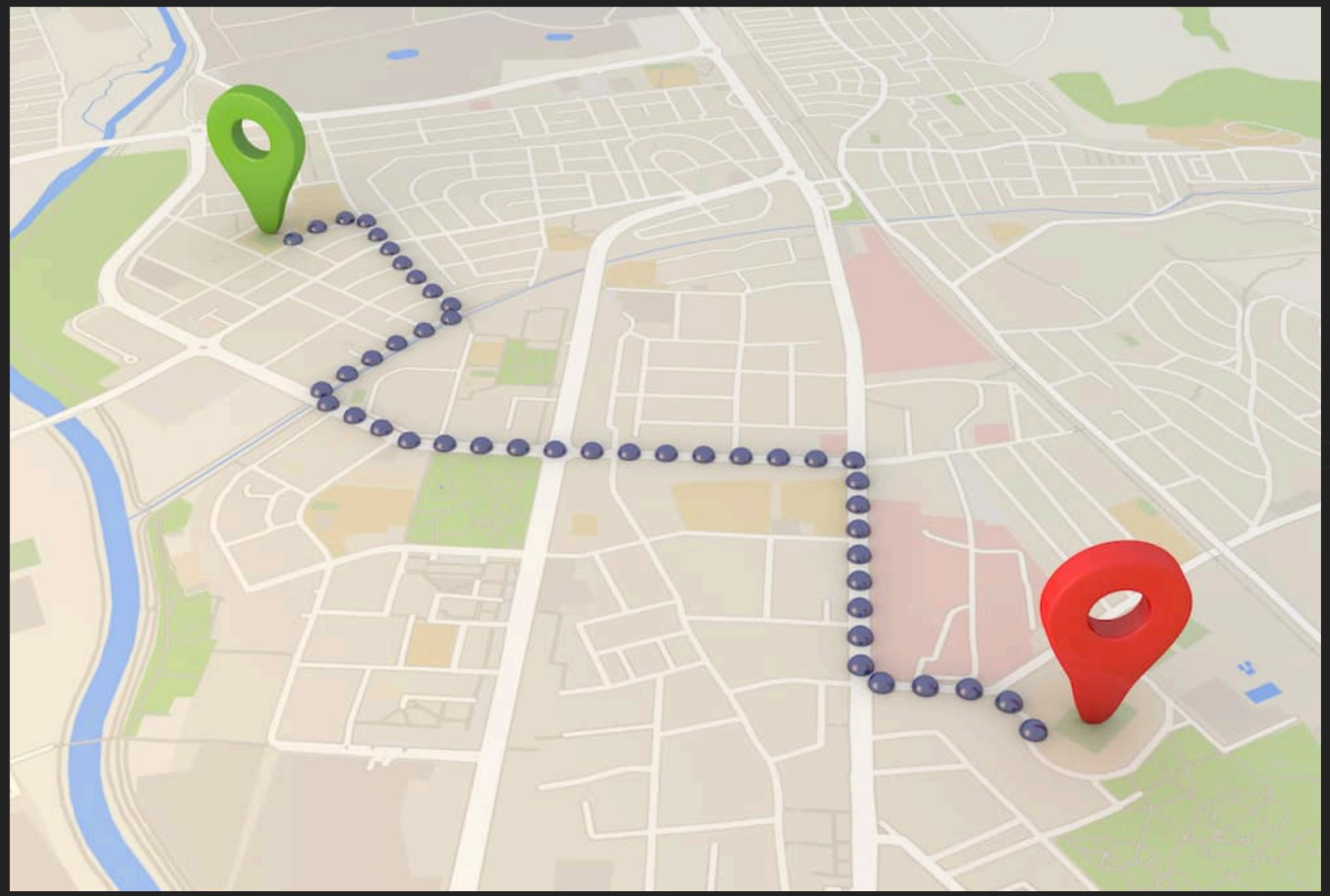
- Optimización de un portfolio de inversión





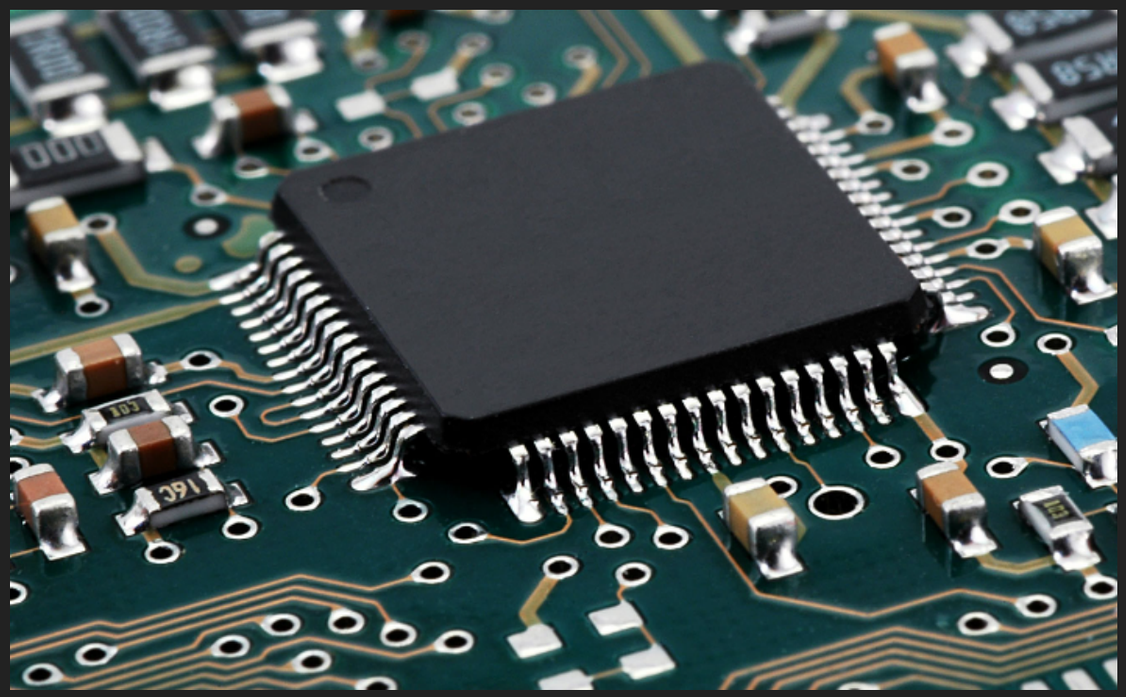
¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

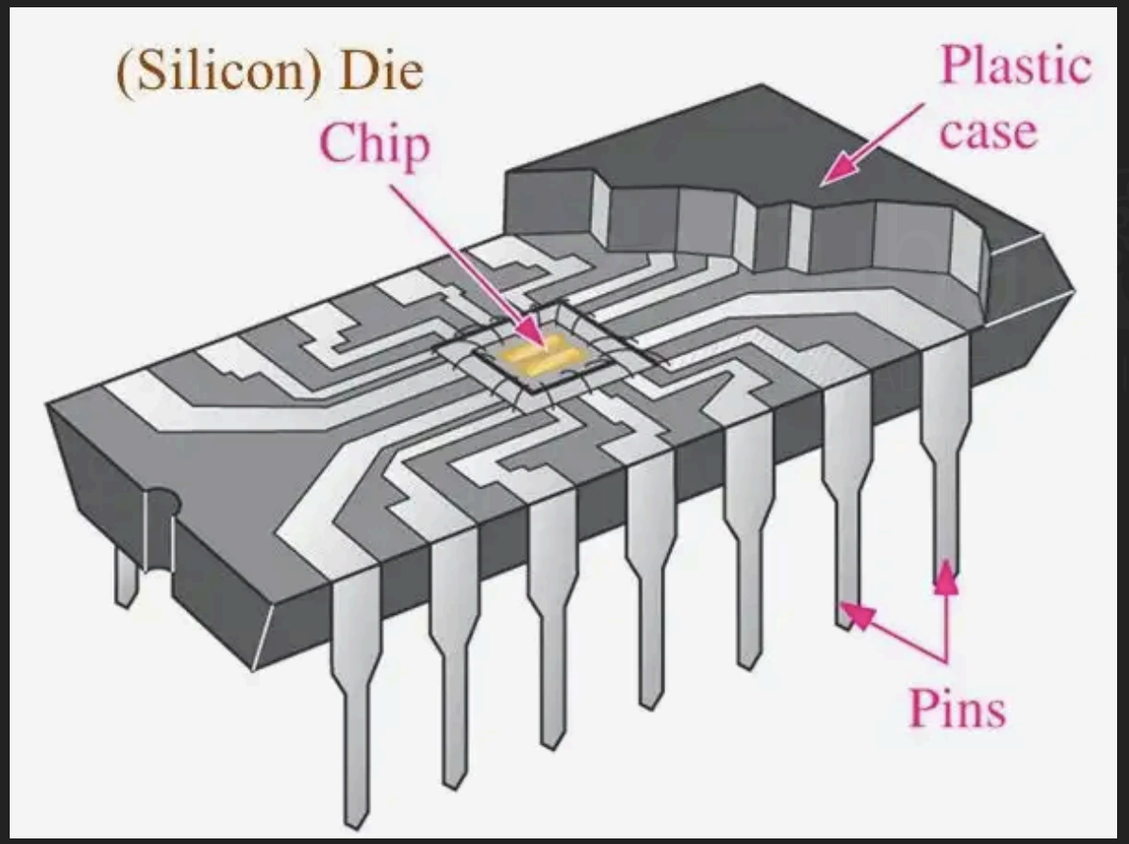
- Transporte



¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

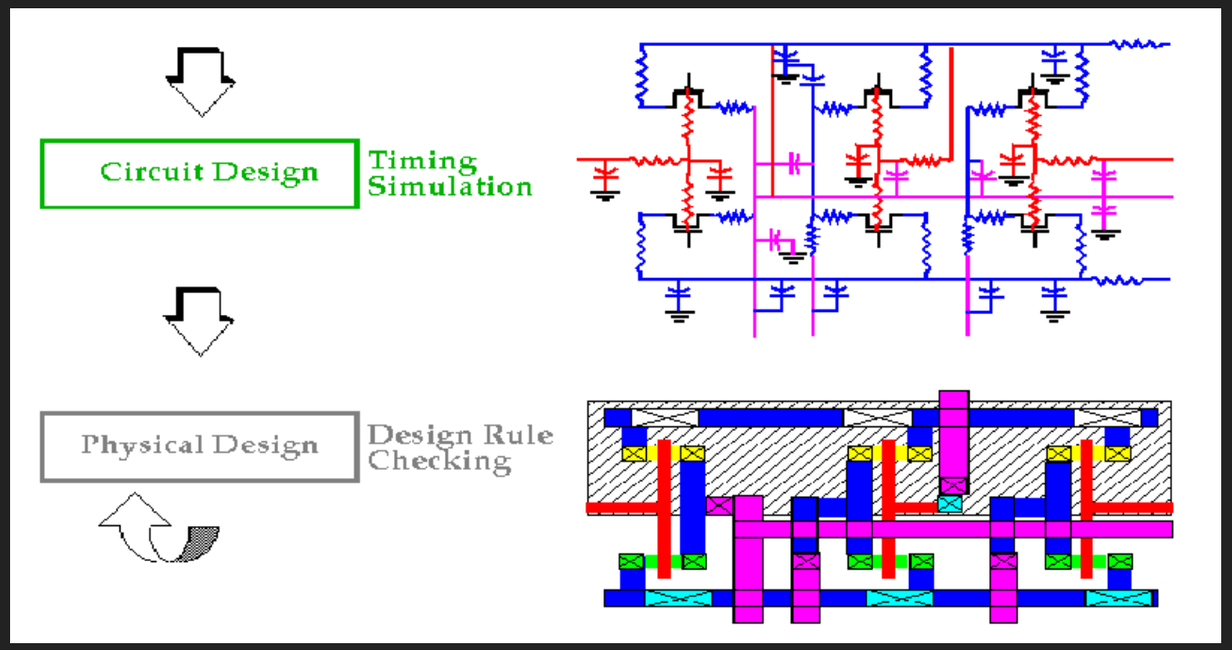
- Diseño de circuitos integrados VLSI (Very Large Scale Integration)

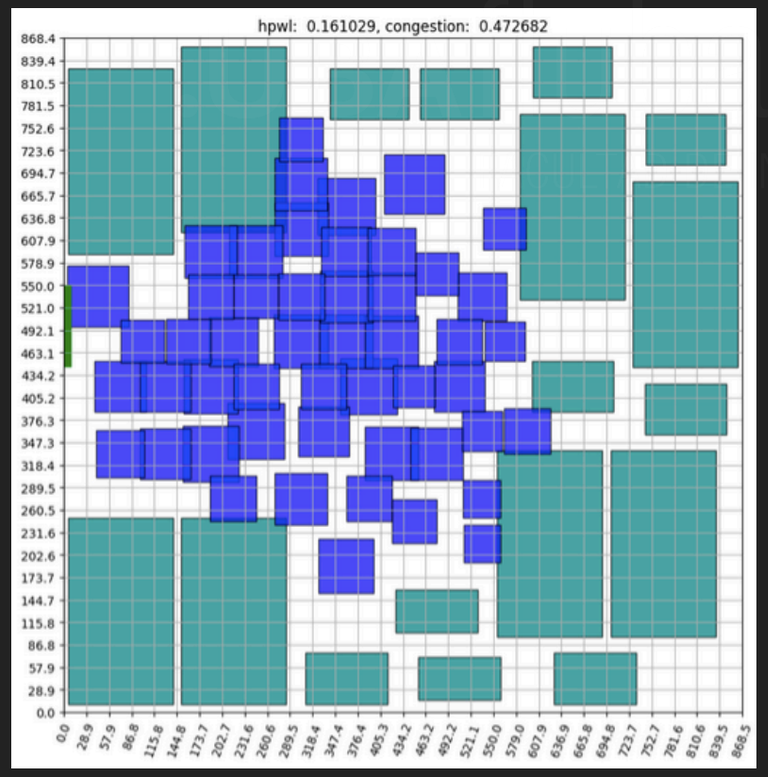




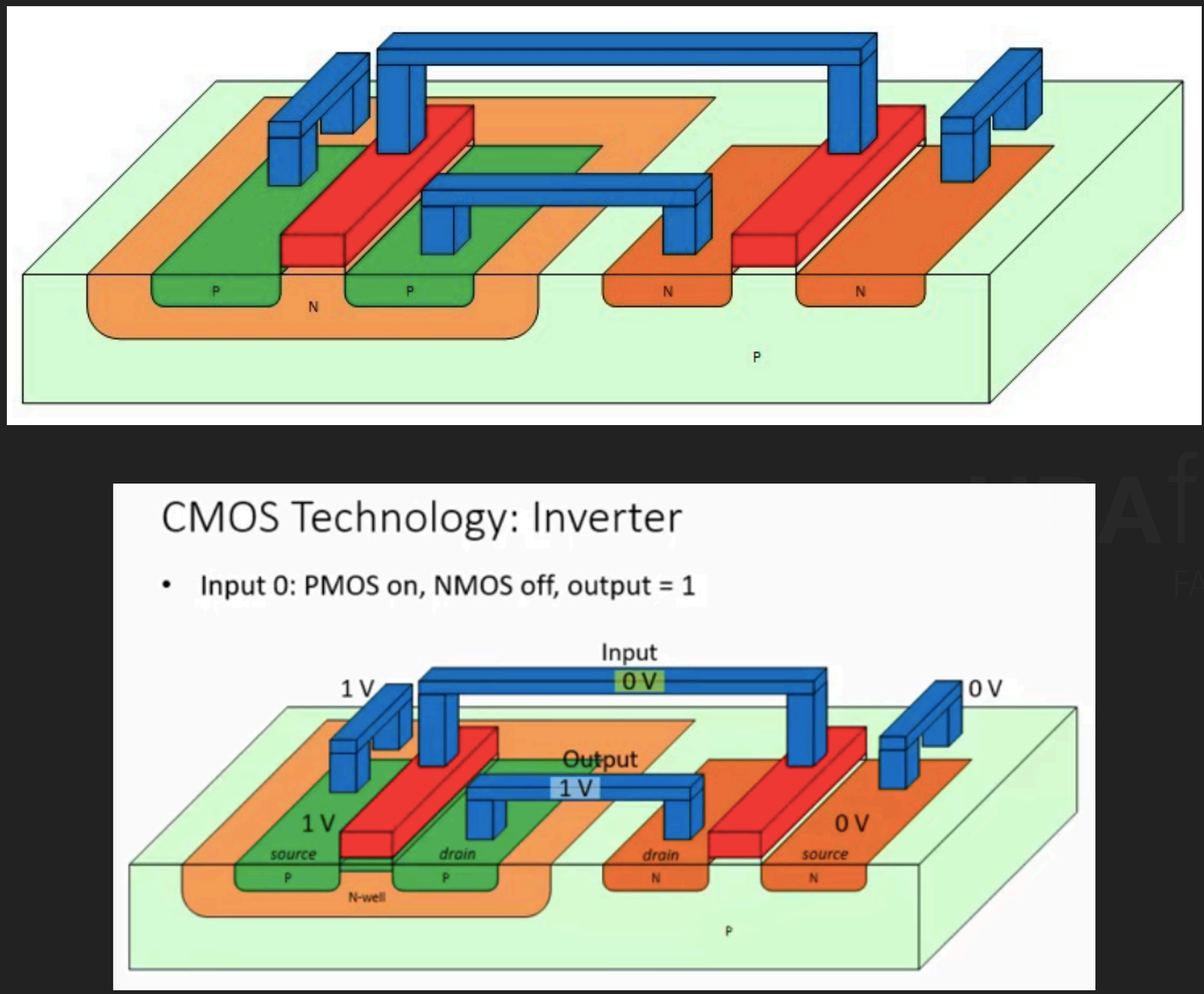
¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

- Diseño de circuitos integrados VLSI (Very Large Scale Integration):
  - ✓ Routing (enrutamiento de conexiones)
    - ➡ Interconexión, max. performance, min. Interferencia, min. longitud “cableado”.
  - ✓ Placement (colocación de componentes macros y clústers)
    - ➡ Floorplaning y PPA





¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?



¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

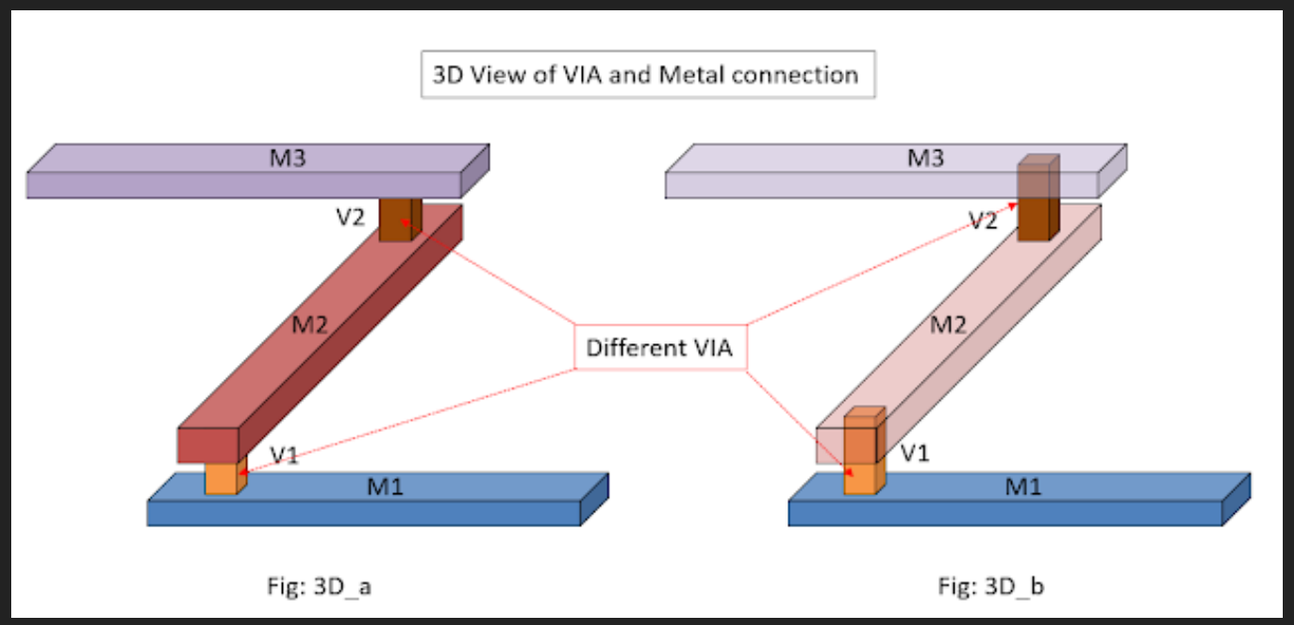
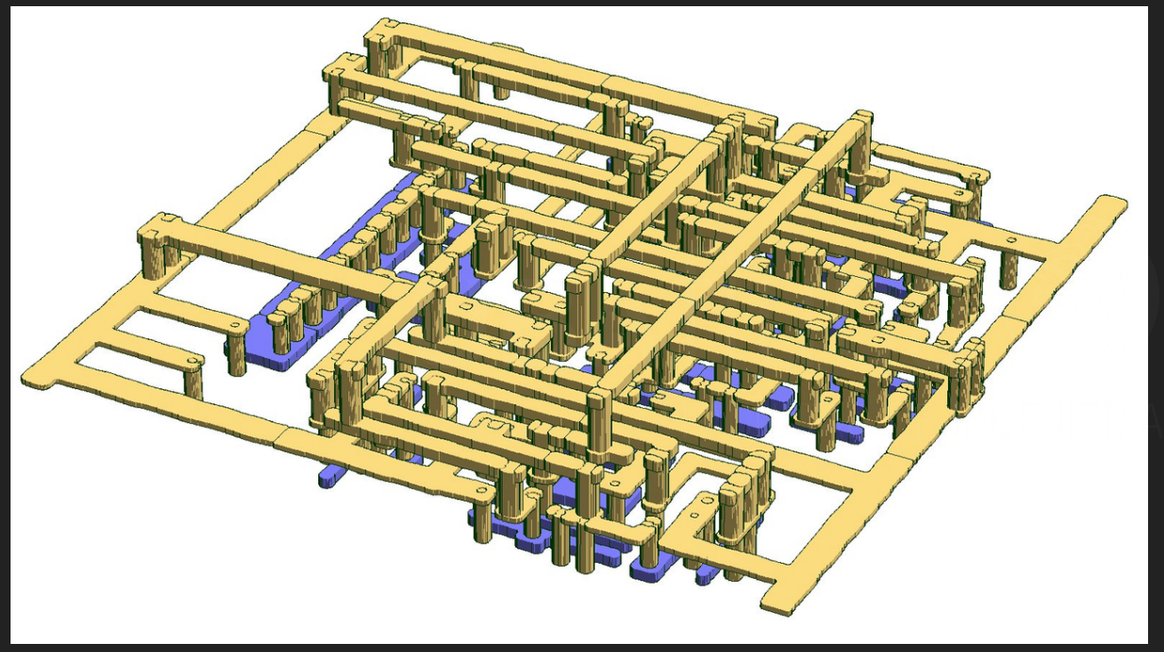
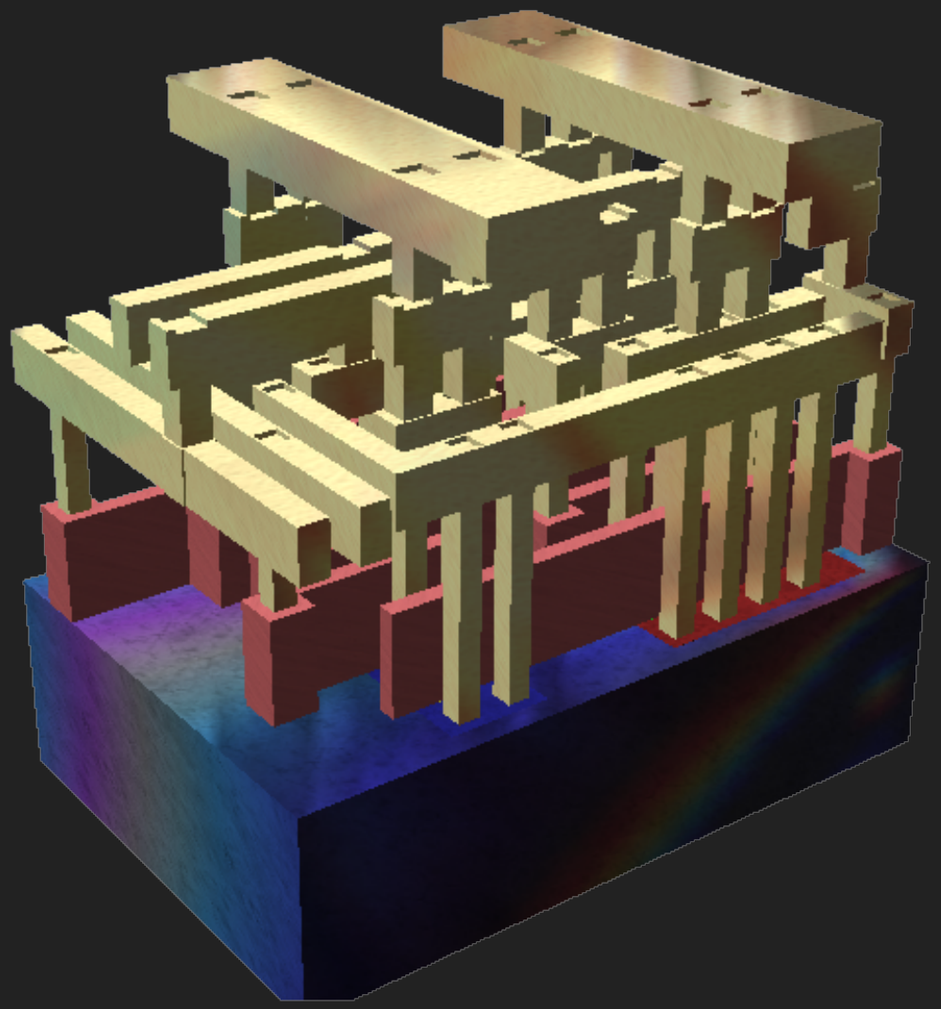


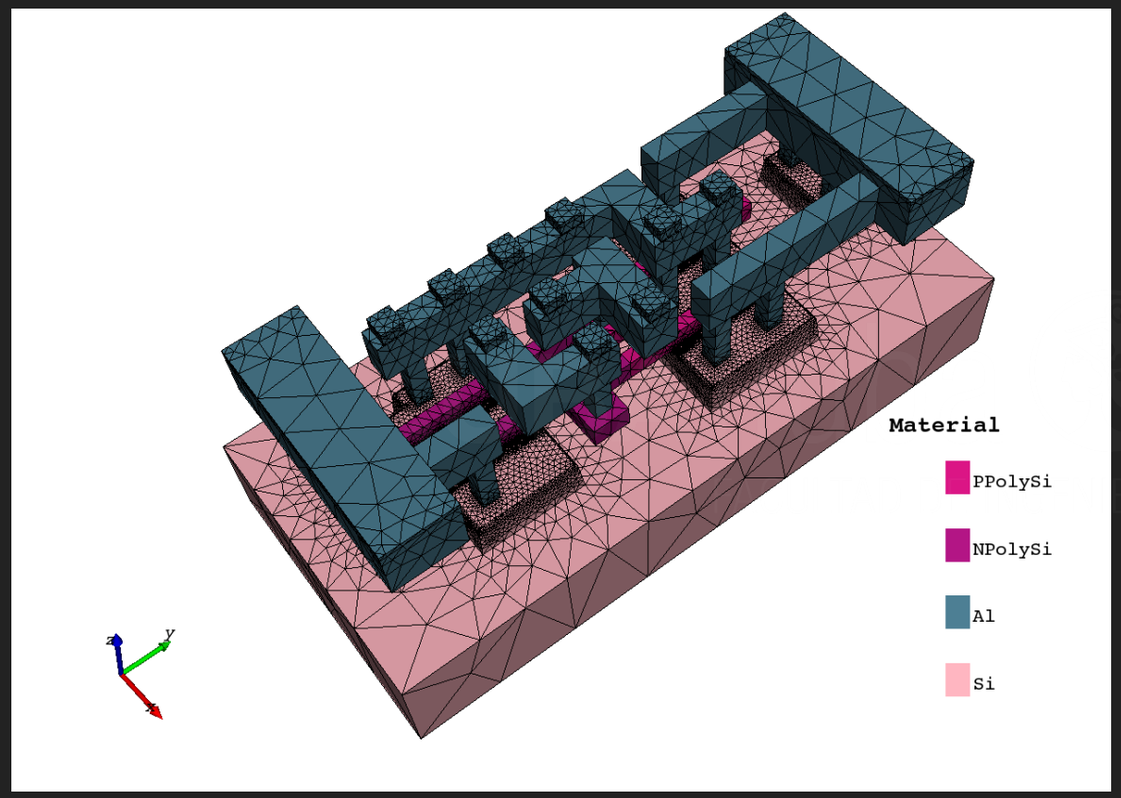
Fig: 3D_a

Fig: 3D_b



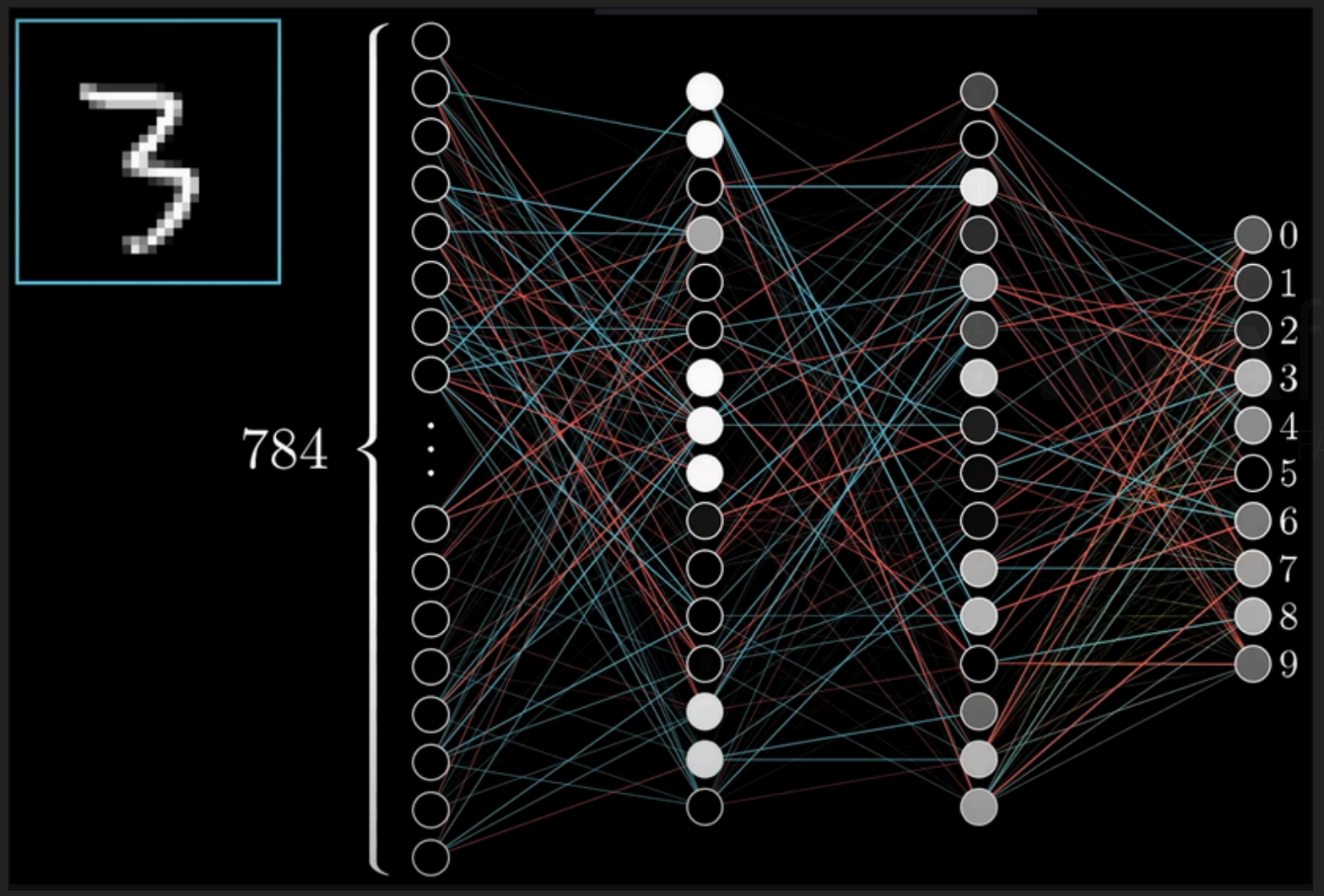
¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?





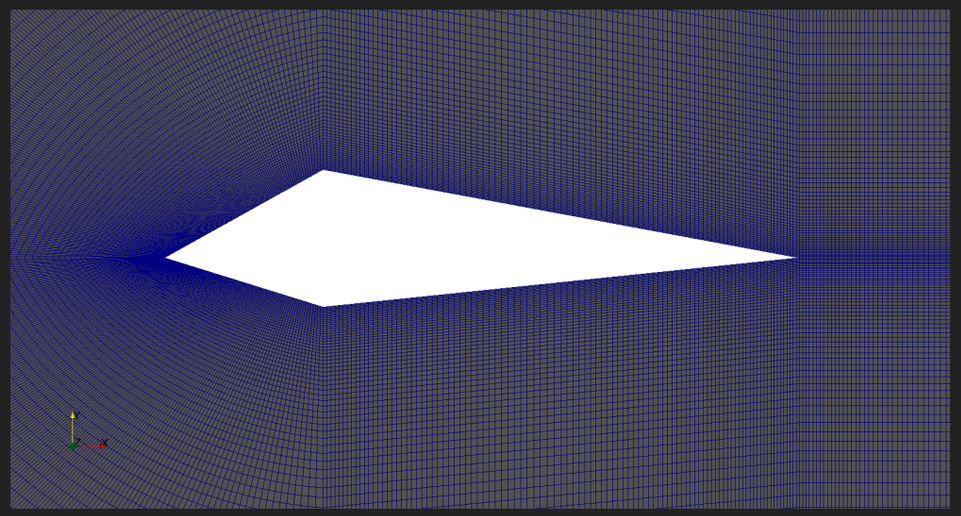
¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

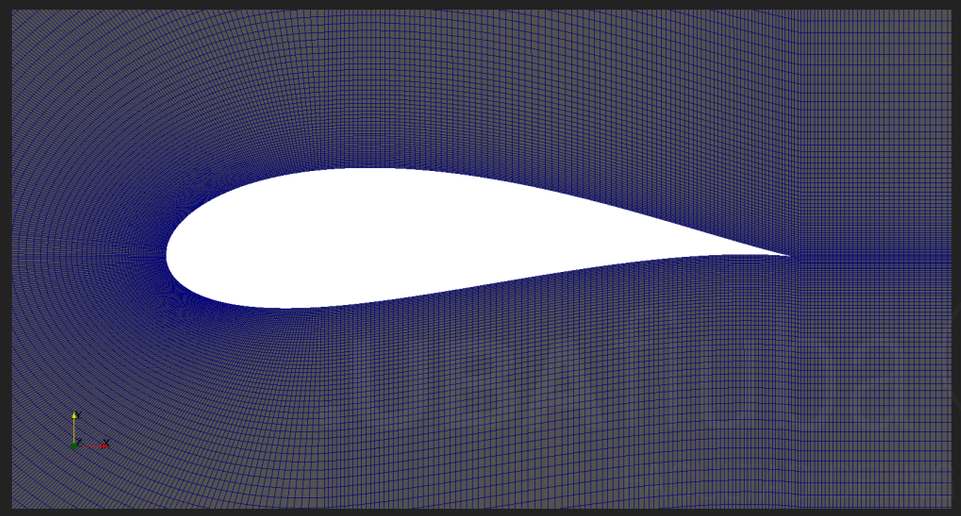
- Optimización de hiperparámetros en DNN

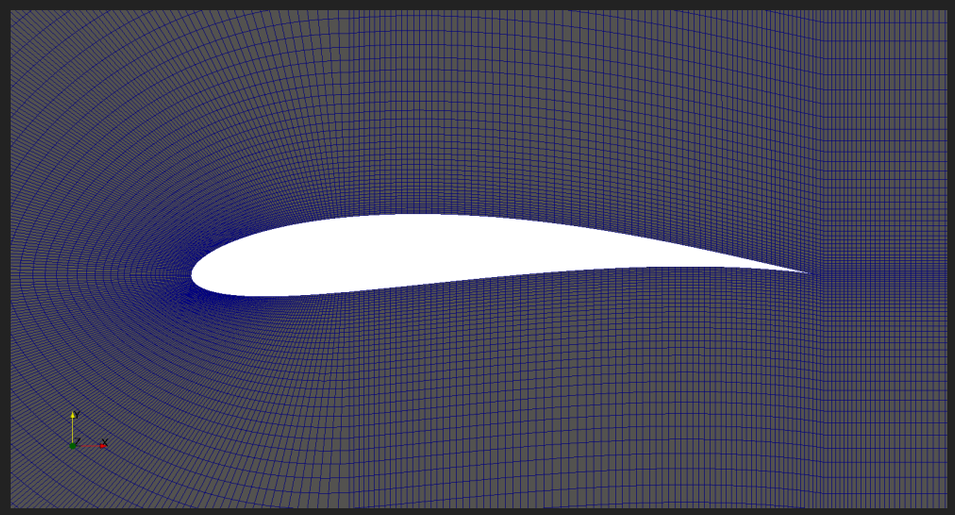


¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

- Diseño óptimo de aerodinámica en vehículos terrestres, aéreos o navales.







¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

- Generador evolutivo de nuevas variantes del mecanismo de atención
- ASI-ARCH (Framework)
- “Momento AlphaGo para el Descubrimiento de Arquitecturas de Modelos” (Liu et al., 2025)

ASI SII-GAIR

AlphaGo Moment for Model Architecture Discovery

Yixiu Liu[1,2,4*] **Yang Nan**[2,4 *] **Weixian Xu**[1,2,4 *] **Xiangkun Hu**[2,4]
Lyumanshan Ye[1,2,4] **Zhen Qin**[3] **Pengfei Liu**[1,2,4†]

[1]Shanghai Jiao Tong University [2]SII [3]Taptap [4]GAIR

SII-GAIR/ASI-Arch Model Gallery

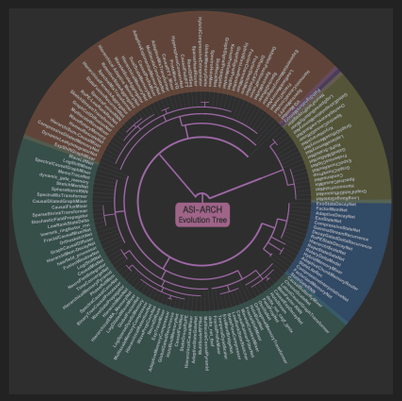
Abstract

While AI systems demonstrate exponentially improving capabilities, the pace of AI research itself remains linearly bounded by human cognitive capacity, creating an increasingly severe development bottleneck. We present ASI-ARCH, the first demonstration of **Artificial Superintelligence for AI research (ASI4AI)** in the critical domain of neural architecture discovery—a fully autonomous system that shatters this fundamental constraint by enabling AI to conduct its own architectural innovation. Moving beyond traditional Neural Architecture Search (NAS), which is fundamentally limited to exploring human-defined spaces, we introduce a paradigm shift from *automated optimization* to *automated innovation*. ASI-ARCH can conduct *end-to-end* scientific research in the challenging domain of architecture discovery, autonomously hypothesizing novel architectural concepts, implementing them as executable code, training and empirically validating their performance through rigorous experimentation and past human and AI experience. ASI-ARCH conducted **1,773** autonomous experiments over **20,000** GPU hours, culminating in the discovery of **106** innovative, **state-of-the-art** (SOTA) linear attention architectures. Like AlphaGo’s **Move 37** that revealed unexpected strategic insights invisible to human players, our AI-discovered architectures demonstrate emergent design principles that systematically surpass human-designed baselines and illuminate previously unknown pathways for architectural innovation (Fig. 2). Crucially, **we establish the first empirical scaling law for scientific discovery** itself—demonstrating that architectural breakthroughs can be scaled computationally, **transforming research progress from a human-limited to a computation-scalable process**. We provide comprehensive analysis of the emergent design patterns and autonomous research capabilities that enabled these breakthroughs, establishing a blueprint for self-accelerating AI systems. To democratize AI-driven research, we open-source the complete framework, discovered architectures, and cognitive traces.

¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

- Argumento:
  - ✓ La investigación en IA está limitada por la capacidad cognitiva humana –> cuello de botella en el desarrollo de nuevas arquitecturas de LLMs.
- Propuesta:

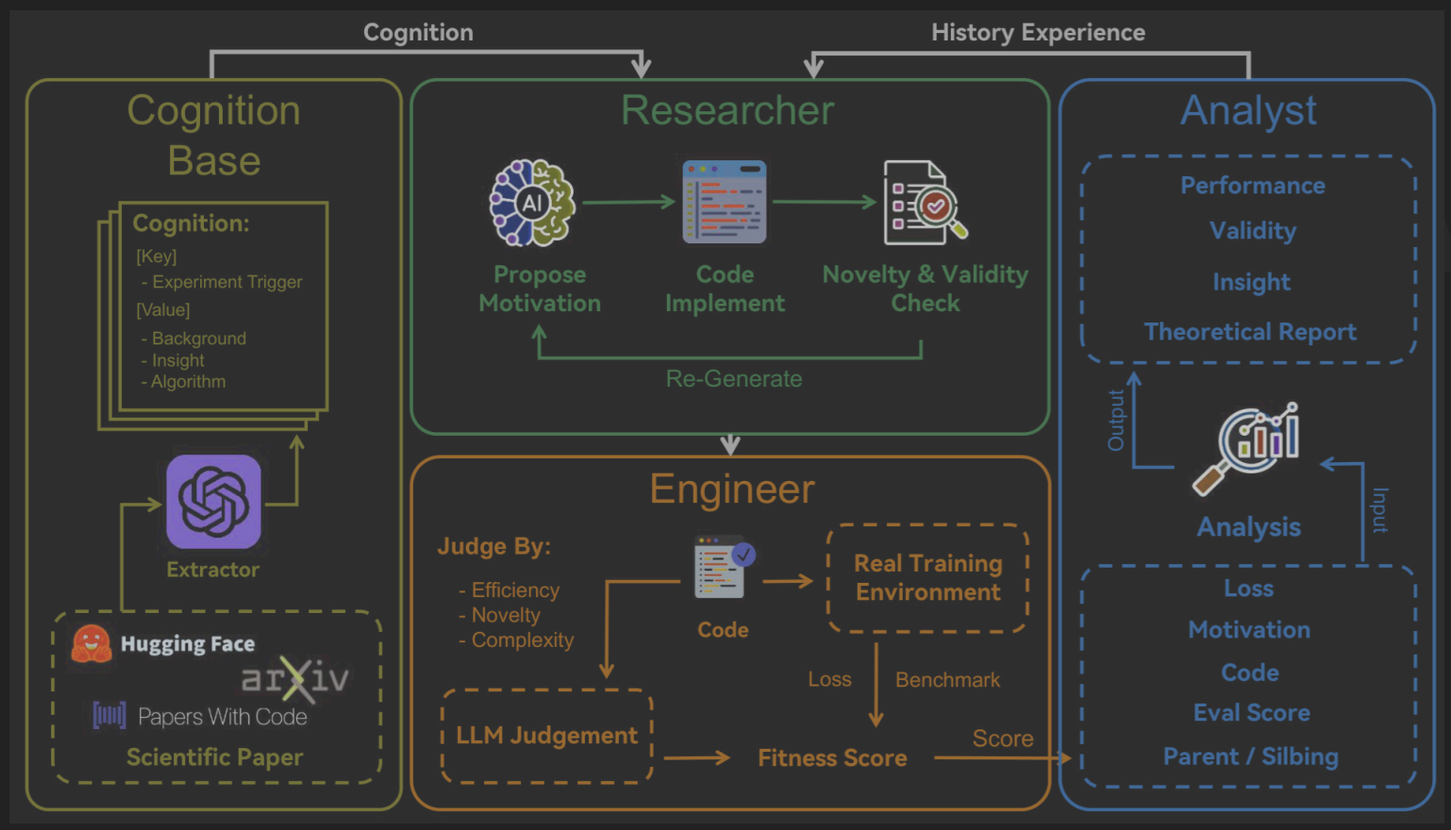
  Diseñar un framework que:
  - ✓ genera hipótesis de nuevas arquitecturas
  - ✓ las implementa en código
  - ✓ entrena los modelos y
  - ✓ valida su desempeño empíricamente de forma automática.
- Resultado:
  - ✓ Creó 106 arquitecturas de atención lineal.



¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

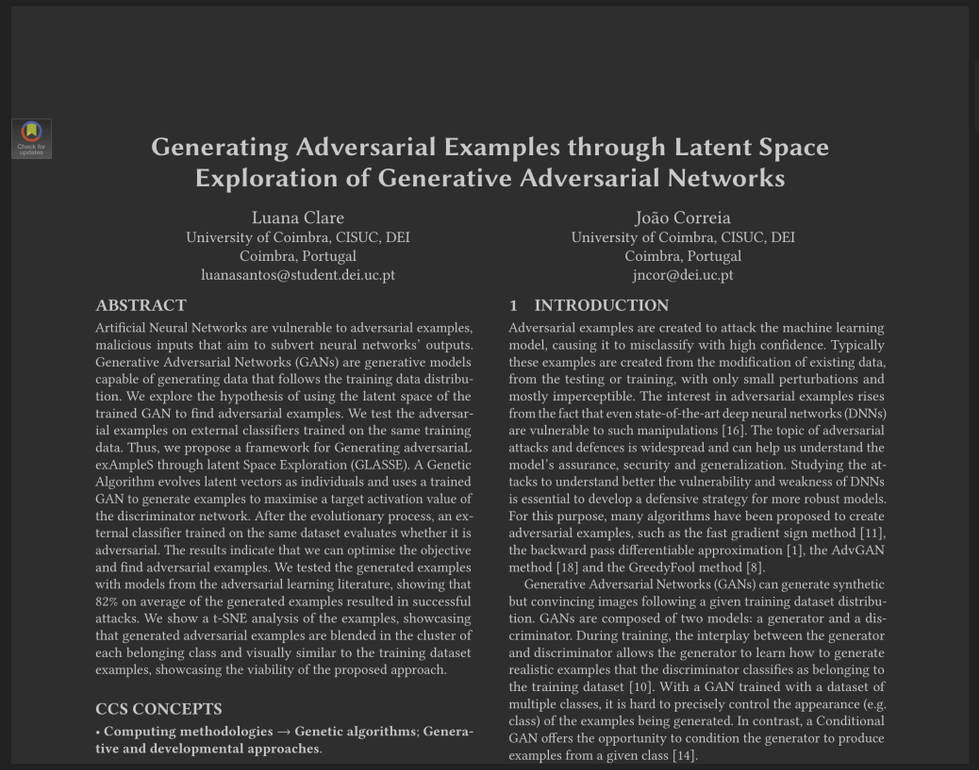
- Función Fitness:

$$\text{Fitness} = \frac{1}{3}\left[\sigma(\Delta_{\text{loss}}) + \sigma(\Delta_{\text{benchmark}}) + \text{LLM}_{\text{judge}}\right]$$



¿DE QUE SE TRATA ALGORITMOS EVOLUTIVOS I?

- "Generación de ejemplos antagónicos mediante la exploración del espacio latente de redes generativas antagónicas" (Clare et al., 2023)
- Usa un Algoritmo Genético para buscar vectores latentes (semillas óptimas) que produzcan imágenes que engañen al clasificador.



Generating Adversarial Examples through Latent Space Exploration of Generative Adversarial Networks

Luana Clare
University of Coimbra, CISUC, DEI
Coimbra, Portugal
luanasantos@student.dei.uc.pt

João Correia
University of Coimbra, CISUC, DEI
Coimbra, Portugal
jncor@dei.uc.pt

ABSTRACT

Artificial Neural Networks are vulnerable to adversarial examples, malicious inputs that aim to subvert neural networks' outputs. Generative Adversarial Networks (GANs) are generative models capable of generating data that follows the training data distribution. We explore the hypothesis of using the latent space of the trained GAN to find adversarial examples. We test the adversarial examples on external classifiers trained on the same training data. Thus, we propose a framework for Generating adversariaL exAmpleS through latent Space Exploration (GLASSE). A Genetic Algorithm evolves latent vectors as individuals and uses a trained GAN to generate examples to maximise a target activation value of the discriminator network. After the evolutionary process, an external classifier trained on the same dataset evaluates whether it is adversarial. The results indicate that we can optimise the objective and find adversarial examples. We tested the generated examples with models from the adversarial learning literature, showing that 82% on average of the generated examples resulted in successful attacks. We show a t-SNE analysis of the examples, showcasing that generated adversarial examples are blended in the cluster of each belonging class and visually similar to the training dataset examples, showcasing the viability of the proposed approach.

CCS CONCEPTS

• **Computing methodologies** → **Genetic algorithms**; **Generative and developmental approaches**.

1 INTRODUCTION

Adversarial examples are created to attack the machine learning model, causing it to misclassify with high confidence. Typically these examples are created from the modification of existing data, from the testing or training, with only small perturbations and mostly imperceptible. The interest in adversarial examples rises from the fact that even state-of-the-art deep neural networks (DNNs) are vulnerable to such manipulations [16]. The topic of adversarial attacks and defences is widespread and can help us understand the model's assurance, security and generalization. Studying the attacks to understand better the vulnerability and weakness of DNNs is essential to develop a defensive strategy for more robust models. For this purpose, many algorithms have been proposed to create adversarial examples, such as the fast gradient sign method [11], the backward pass differentiable approximation [1], the AdvGAN method [18] and the GreedyFool method [8].

Generative Adversarial Networks (GANs) can generate synthetic but convincing images following a given training dataset distribution. GANs are composed of two models: a generator and a discriminator. During training, the interplay between the generator and discriminator allows the generator to learn how to generate realistic examples that the discriminator classifies as belonging to the training dataset [10]. With a GAN trained with a dataset of multiple classes, it is hard to precisely control the appearance (e.g. class) of the examples being generated. In contrast, a Conditional GAN offers the opportunity to condition the generator to produce examples from a given class [14].

TEMAS Y TÉCNICAS A ABORDAR EN EL CURSO

TEMAS Y TÉCNICAS A ABORDAR EN EL CURSO

- Conceptualización de términos utilizados en AE
- Algoritmos Genéticos (GA)
- Optimización por Enjambre de Partículas (PSO)
- Optimización basada en Enseñanza Aprendizaje (TLBO)
- Optimización basada en Colonia de Hormigas (ACO)
- Búsqueda Tabú (TS)
- Hiperheurísticas

CURSADO

- Parte 1 (~ 80 min.)
- Break (~ 10-20 min.)
- Parte 2 (~ 80 min.)

EVALUACIÓN

EVALUACIÓN

- Se aprueba la materia implementando 1 desafío práctico presentado en forma:
  - ✓ Individual.
  - ✓ Grupal (hasta 4 integrantes).
- La entrega consta de:
  - ✓ Informe explicativo de al menos 3 carillas (.pdf).
  - ✓ Código fuente en Python (.py, .ipynb, etc.) transcripto parcialmente en el informe y completo en un repositorio (el enlace al repositorio debe proporcionarse en el informe).
- La modalidad de entrega se realizará:
  - ✓ Durante la cursada o al final subiendo el .pdf al campus de la FIUBA.
  - ✓ Sin exposición en la última clase.
- Los desafíos son abiertos. El estudiante o el grupo elige un caso de uso según una o varias de las técnicas de Aprendizaje por Refuerzo vistas durante la cursada y elabora el desafío práctico.

REFERENCIAS BIBLIOGRÁFICAS Y WEB (I)

- Eiben, A. E., & Smith, J. E. (2015). Introduction to evolutionary computing. Springer-Verlag Berlin Heidelberg.
- Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT press.
- Mitchell, M. (1998). An introduction to genetic algorithms. MIT press.
- Michalewicz Z. (1996). Genetic Algorithms + Data Structures = Evolution Programs. Springer Verlag.
- Udayan Das et al. (2024). Introduction to Python Programming. OpenStax. ISBN 978-1-961584-45-7.
- Introducing Python (Third release). Lubanovic B. (2019). O'Reilly. ISBN 978-1-449-35936-2.
- https://www.python.org/downloads/release/python-3130b1/
- https://matplotlib.org
- Roy, S., & Chakraborty, U. (2013). Introduction to soft computing: neuro-fuzzy and genetic algorithms. Pearson.
- Hillier, F. S. Lieberman, G. J. (2010). Introducción a la Investigación de Operaciones.

REFERENCIAS BIBLIOGRÁFICAS Y WEB (II)

- Lu Y, Hao J, Wu Q. (2022). Solving the clustered traveling salesman problem via traveling salesman problem methods. PeerJ Computer Science 8:e972
- Pratihar, D. K. (2007). Soft computing. Alpha Science International, Ltd.
- A. Cauchy. (1847). Méthode générale pour la résolution des systèms d'équations simultanées, Comp. Rend. Sci. Paris, 25. 536-538.
- Chong, E. K., & Żak, S. H. (2013). An introduction to optimization (Vol. 75). John Wiley & Sons.
- Glover, F. (1986). Future paths for integer programming and links to artificial intelligence. Computers & operations research, 13(5), 533-549.
- J. Kennedy, R. Eberhart. (1995). Particle swarm optimization (in Neural Networks). Proceedings., IEEE International Conference on, vol. 4, pp. 1942 –1948 vol.4.
- Engelbrecht, A. P. (2007). Computational intelligence: an introduction. John Wiley & Sons.
- Clerc, M. (2010). Particle swarm optimization (Vol. 93). John Wiley & Sons.
- Shi, Y., & Eberhart, R. (1998). A Modified Particle Swarm Optimizer. Proceedings of the IEEE International Conference on Evolutionary Computation, 1998. IEEE World Congress on Computational Intellince (Cat. No.98TH8360), Anchorage, AK, USA, 1998, pp. 69-73. DOI: 10.1109/ICEC.1998.699146.
- Bratton, D., & Kennedy, J. (2007, April). Defining a standard for particle swarm optimization. In 2007 IEEE swarm intelligence symposium (pp. 120-127). IEEE.

REFERENCIAS BIBLIOGRÁFICAS Y WEB (III)

- Parsopoulos, K. E., & Vrahatis, M. N. (2002). Particle swarm optimization method for constrained optimization problems. Intelligent technologies–theory and application: New trends in intelligent technologies, 76(1), 214-220.
- Hu, X., Eberhart, R. C., & Shi, Y. (2003, April). Engineering optimization with particle swarm. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706) (pp. 53-57). IEEE.
- Coello Coello, C. A., "Use of a self-adaptive penalty approach for engineering optimization problems", Elsevier Science, Computers in Industry 41, 2000, pp. 113-127.
- Parsopoulos, K. E., and Vrahatis, M. N., "Particle Swarm Optimization Method for Constrained Optimization Problems", in Proceedings of the Euro-International Symposium on Computational Intelligence, 2002.
- Engelbrecht, A. P., "Fundamentals of Computational Swarm Intelligence", John Wiley & Sons Ltd, England, 2005.
- Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 6(2), 182-197.
- Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. TIK report, 103.
- Coello, C. C., & Lechuga, M. S. (2002, May). MOPSO: A proposal for multiple objective particle swarm optimization. In Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600) (Vol. 2, pp. 1051-1056). IEEE.
- López, J. (2013). Optimización multiobjetivo: aplicaciones a problemas del mundo real. Buenos Aires, Argentina, Universidad Nacional de la Plata.
- Yucra López, Carlos Enrique (2021). Biblioteca para la comparación estadística de algoritmos evolutivos. Proyecto Fin de Carrera / Trabajo Fin de Grado, Madrid, España.

REFERENCIAS BIBLIOGRÁFICAS Y WEB (IV)

- https://www.youtube.com/watch?v=oSrUsM0hoPs CMOS Tech: NMOS and PMOS Transistors in CMOS Inverter (3-D View)
- https://www.youtube.com/watch?v=1Lad28K3Xi0 CMOS Fabrication Process (Animation)
- https://www.makinarocks.ai/en/application-specific-integrated-circuit-asic-floorplan-automation-part-ii/
- Liu, Y., Nan, Y., Xu, W., Hu, X., Ye, L., Qin, Z., & Liu, P. (2025). Alphago moment for model architecture discovery. arXiv preprint arXiv:2507.18074.
- Clare, L., & Correia, J. (2023, July). Generating adversarial examples through latent space exploration of generative adversarial networks. In Proceedings of the Companion Conference on Genetic and Evolutionary Computation (pp. 1760-1767).