Algoritmos de optimización bioinspirados

Algoritmos Evolutivos, aplicaciones y proyección

Inteligencia Artificial
INFO257

Profesor: Jorge Maturana
jorge.maturana@inf.uach.cl

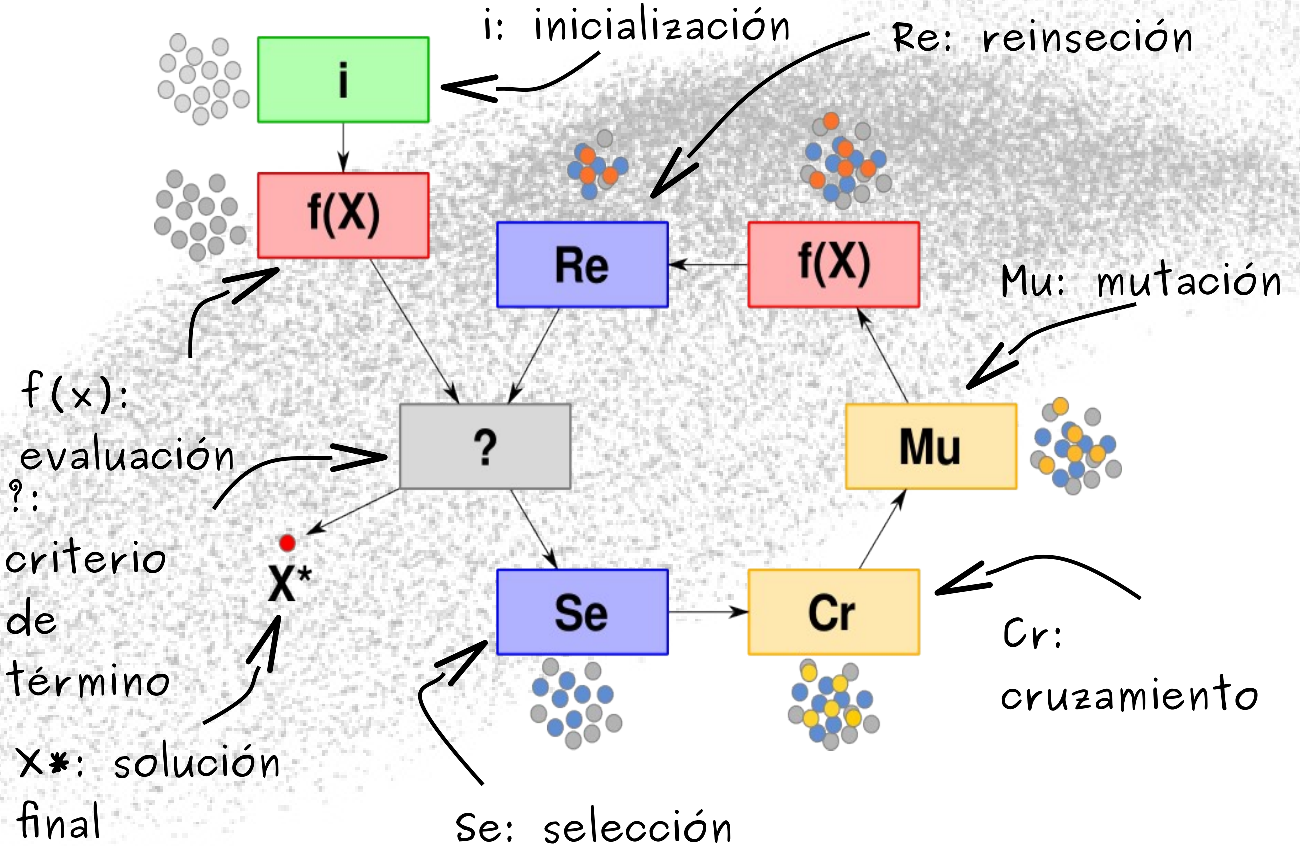
Evolución natural y artificial

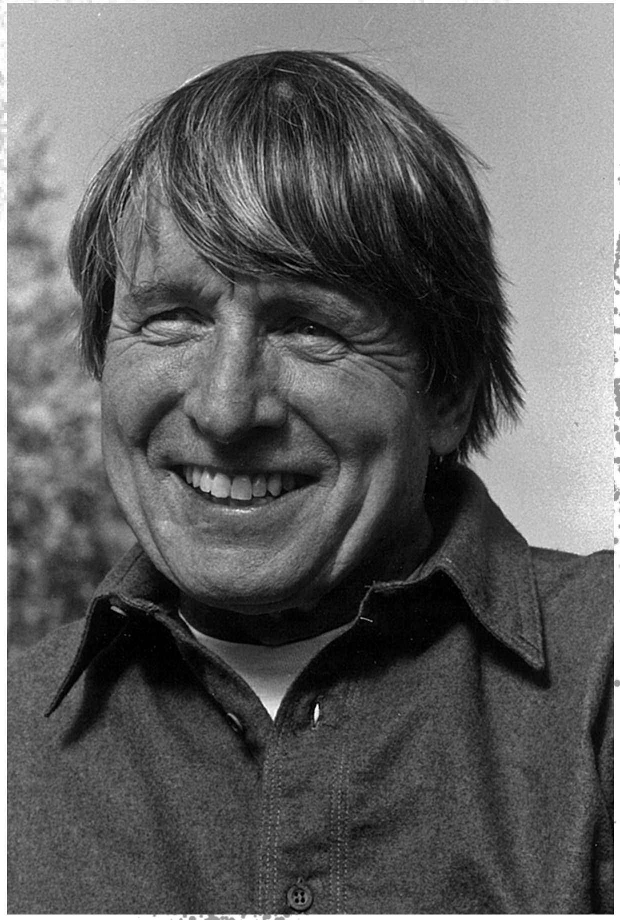
- Paralelo entre naturaleza y problemas de optimización:

Natural	Artificial
ADN del Individuo	Conjunto de variables a determinar
Aptitud	Función objetivo
Población de individuos	Conjunto de soluciones
Cruzamiento	Combinación de variables
Mutación	Perturbación aleatoria

Algoritmo Evolutivo

- Propuesta de base: ***Algoritmo Genético***
  - Propuesto por John Holland en 1975





Aplicaciones de los EAs

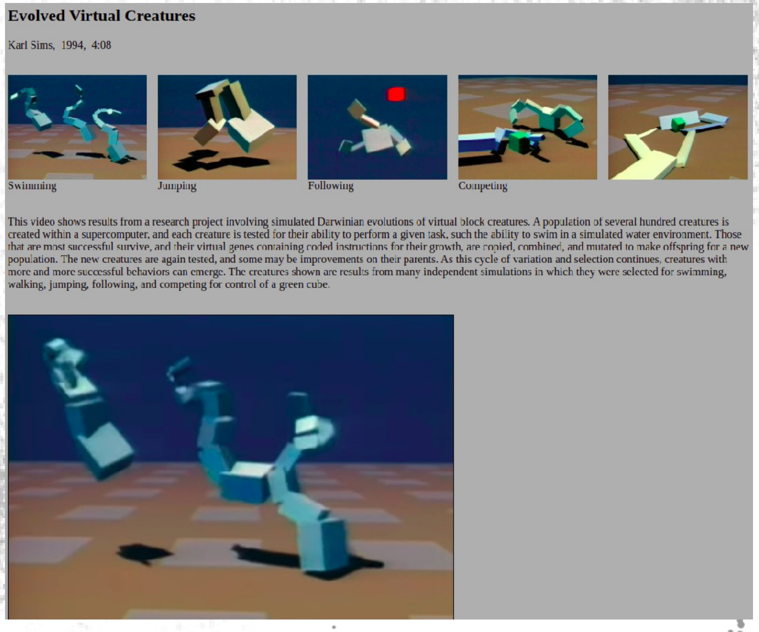
- Principal atributo: lograr soluciones **no intuitivas**
  - Prueba de miles de soluciones generadas al azar
  - Combinaciones de ellas
  - Sin sesgo humano hacia "soluciones correctas"

- Algunos ejemplos:
- **Diseño de antenas**
  - NASA NT5 spacecraft antenna (2005)
  - Montado en satélites
  - El primer objeto obtenido por evolución artificial en el espacio

Public Domain, https://commons.wikimedia.org/w/index.php?curid=671157

Aplicaciones de los EAs

- **Evolved Virtual Creatures**
  - Karl Sims, 1994
  - Proceso evolutivo mide desempeño de criaturas en ambiente virtual
  - Varios trabajos inspirados en este



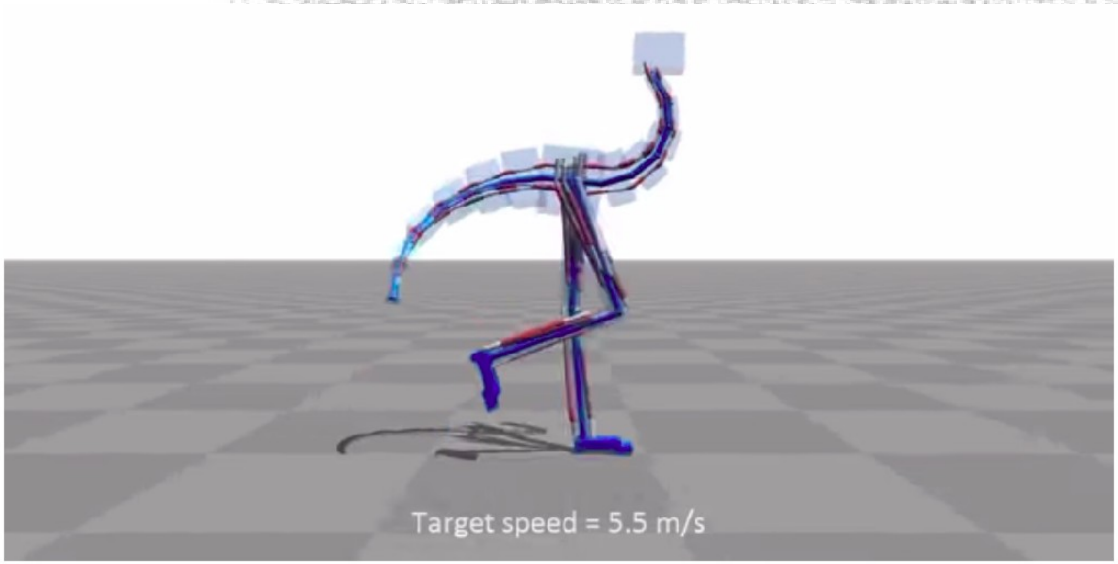
Evolved Virtual Creatures

Karl Sims, 1994, 4:08

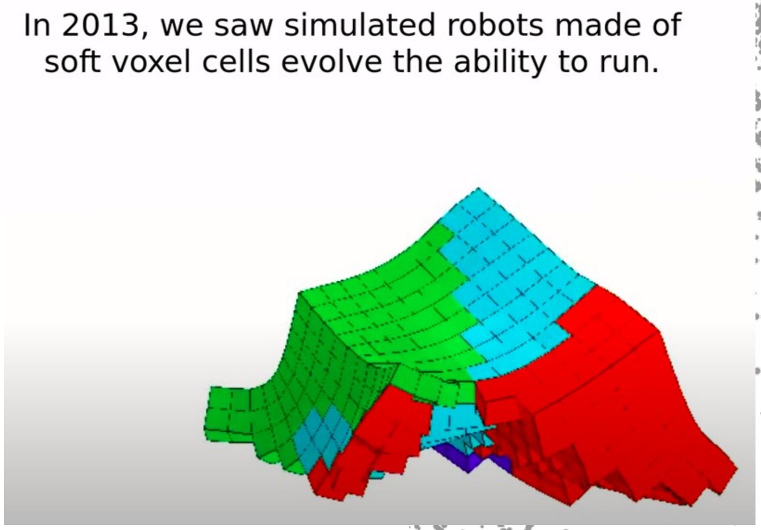
Swimming Jumping Following Competing

This video shows results from a research project involving simulated Darwinian evolutions of virtual block creatures. A population of several hundred creatures is created within a supercomputer, and each creature is tested for their ability to perform a given task, such the ability to swim in a simulated water environment. Those that are most successful survive, and their virtual genes containing coded instructions for their growth, are copied, combined, and mutated to make offspring for a new population. The new creatures are again tested, and some may be improvements on their parents. As this cycle of variation and selection continues, creatures with more and more successful behaviors can emerge. The creatures shown are results from many independent simulations in which they were selected for swimming, walking, jumping, following, and competing for control of a green cube.

https://www.karlsims.com/evolved-virtual-creatures.html



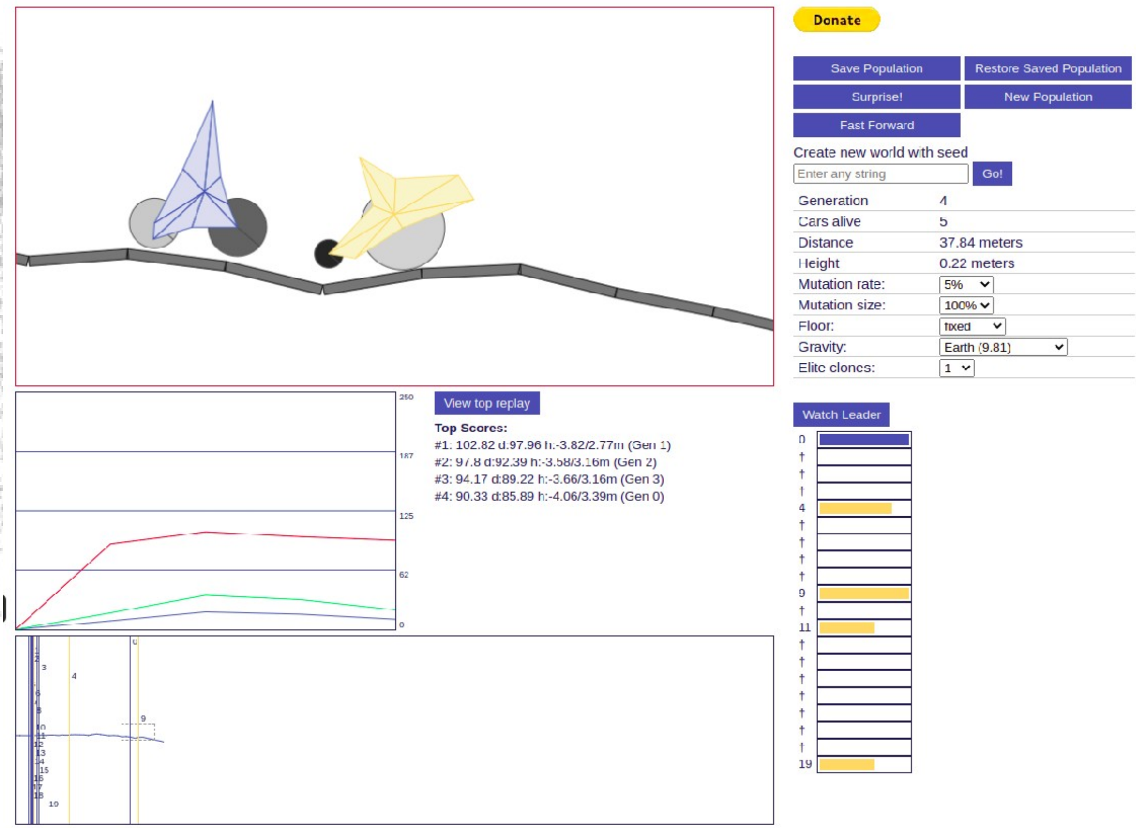
https://www.goatstream.com/research/papers/SA2013/



Cheney, Clune, Lipson, GECCO 2013
https://www.youtube.com/watch?v=HgWQ-gPIvt4

Aplicaciones de los EAs

- **Evolución de diseños de vehículos**
  - “ADN” del individuo determina su forma
  - No existe función de evaluación, esta se calcula con la simulación



https://rednuht.org/genetic_cars_2/

Aplicaciones de los EAs

- **Diseño de estructuras**
  - Diseño de arcos de puentes de celosía
  - Integrado con método de elementos finitos
  - Determinar la forma del puente
  - Minimizar peso total de la estructura
  - Soportar carga
  - Restricciones de tensión y deformación máximas

Journal of Constructional Steel Research 66 (2010) 1011–1017

Contents lists available at ScienceDirect

Journal of Constructional Steel Research

journal homepage: www.elsevier.com/locate/jcsr

ELSEVIER

Optimum design of steel truss arch bridges using a hybrid genetic algorithm

Jin Cheng

Department of Bridge Engineering, Tongji University, Shanghai, 200092, China

A R T I C L E I N F O

Article history:
Received 4 December 2009
Accepted 4 March 2010

Keywords:
Steel truss arch bridge
Genetic algorithm
Minimum weight design
Finite element method
Optimization

A B S T R A C T

The design of steel truss arch bridges is formulated as an optimization problem. The objective function considered is the weight of the steel truss arch bridge. The objective function is minimized subjected to the design constraints of strength (stress) and serviceability (deflection). An efficient, accurate, and robust algorithm is proposed for optimal design of steel truss arch bridges. The proposed algorithm integrates the concepts of the genetic algorithm (GA) and the finite element method. A real-coded/integer-coded method is used to realistically represent the values of the design variables. Three GA operators consisting of constraint aggregate selection procedure, arithmetic crossover, and non-uniform mutation are proposed. Finite element method is used to compute values of implicit objective functions. A numerical example involving a detailed computational model of a long span steel truss arch bridge with a main span of 552 m is presented to demonstrate the applicability and merits of the proposed method.

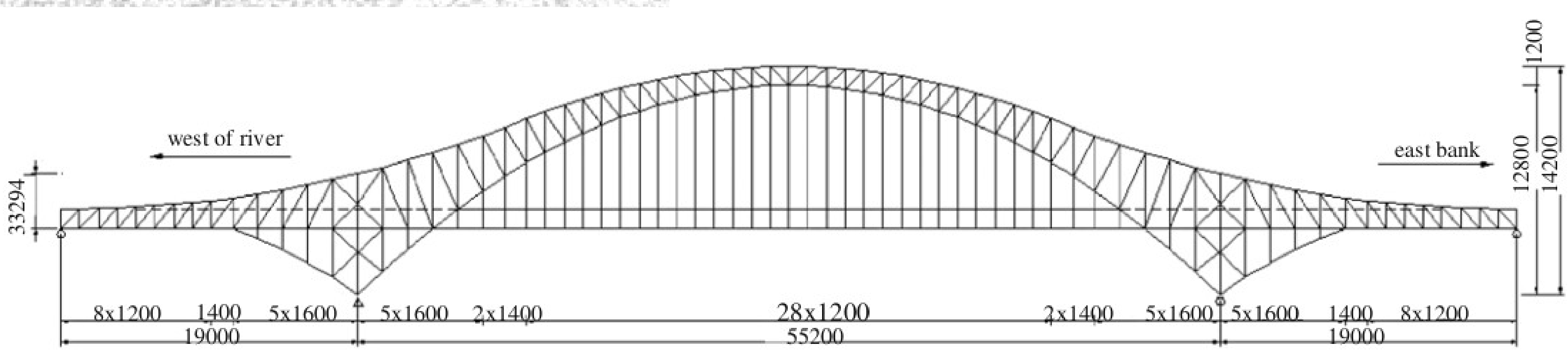


Fig. 2. Elevation view of Chaotianmen Bridge (Unit: mm).

Public Domain, https://commons.wikimedia.org/w/index.php?curid=671157

Aplicaciones de los EAs

- **Layout design**
  - Distribuir máquinas en líneas de producción
  - Maximizar uso del área
  - Disminuir costos de transporte entre estaciones
  - Gran impacto en costos operacionales

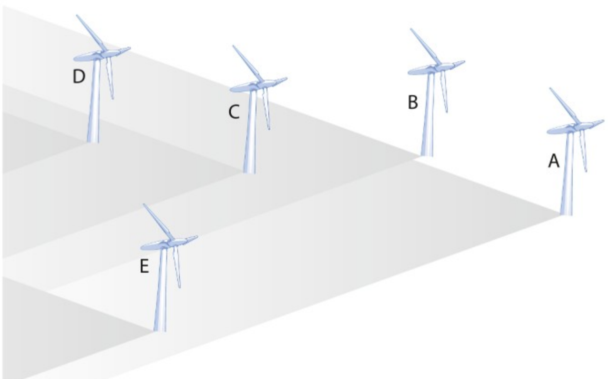
Plant Layout Optimization Using Evolutionary Algorithms

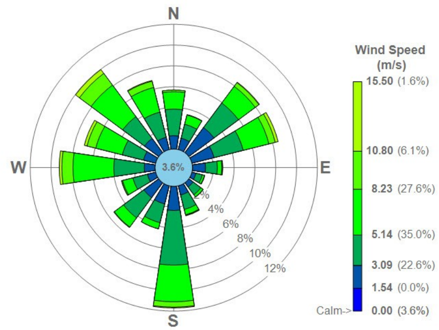
Jiří Kubalík[1(✉)], Petr Kadera[1], Václav Jirkovský[1], Lukáš Kurilla[2], and Šimon Prokop[2]

[1] Czech Institute of Informatics, Robotics, and Cybernetics, Czech Technical University in Prague, Jugoslávských partyzánů 1580/3, 160 00 Prague, Czech Republic
{jiri.kubalik,petr.kadera,vaclav.jirkovsky}@cvut.cz
[2] Faculty of Architecture, Czech Technical University in Prague, Thákurova 9, 166 34 Prague, Czech Republic
{kurilluk,prokosim}@fa.cvut.cz

Abstract. Facility layout problems, i.e., optimal placement of production units in a plant, become an inseparable part of manufacturing systems design and management. They are known to greatly impact the system performance. This paper proposes a new formulation of the facility layout problem where workstations are to be placed into a hall. Within the hall, obstacles and communications can be defined. Each workstation can have multiple handling spaces attached to its sides and oriented links can be defined between workstations. A new evolutionary-based approach to solve this facility layout problem is proposed in single-objective as well as multi-objective variant. The method is experimentally evaluated on a set of standard VLSI floorplanning benchmarks as well as on the data set created specifically for the proposed facility layout problem. Results show the method is both competitive to the state-of-the-art floorplanners on the VLSI benchmarks and produces high-quality solutions to the proposed facility layout problem.

- **Wind farm layout**
  - Considerar dirección predominante del viento
  - Considerar efecto entre generadores
  - Maximizar eficiencia





Applied Energy 248 (2019) 429–445

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

ELSEVIER

Wind farm layout optimization using self-informed genetic algorithm with information guided exploitation

Xinglong Ju[a], Feng Liu[b,c,*]

[a] Department of Industrial, Manufacturing and Systems Engineering, University of Texas at Arlington, Arlington, TX 76019, USA
[b] Department of Anesthesia, Critical Care and Pain Medicine, Massachusetts General Hospital, Harvard Medical School, Boston, MA 02114, USA
[c] The Picower Institute for Learning and Memory, MIT, Cambridge, MA 02139, USA

HIGHLIGHTS

- Self-awareness mechanism which is inspired from the nature is incorporated in the genetic algorithm.
- The proposed algorithm showed improved efficiency in the wind farm layout problem.
- A Python package is developed and will be made public available to help researchers in this research field.

ARTICLE INFO

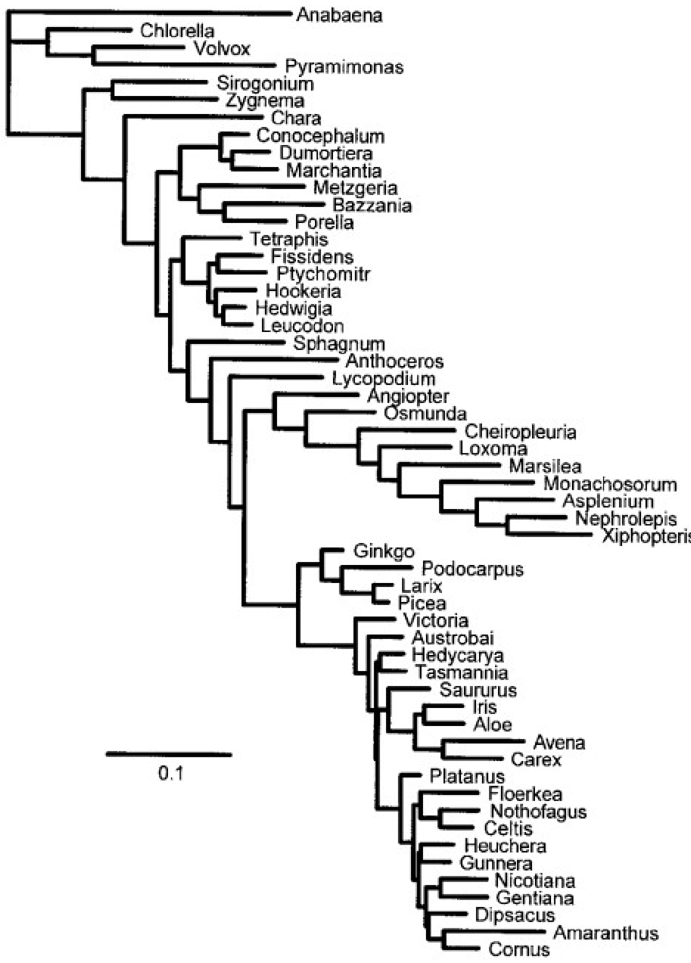
Keywords:
Wind farm layout optimization
Genetic algorithm
Monte Carlo simulation
Multivariate Adaptive Regression Splines (MARS)

ABSTRACT

Wind energy which is known for its cleanliness and cost-effectiveness has been one of the main alternatives for fossil fuels. An integral part is to maximize the wind energy output by optimizing the layout of wind turbines. In this paper, we first discuss the drawbacks of Conventional Genetic Algorithm (CGA) by investigating into the implications of crossover and mutation steps of CGA for the wind farm layout problem, which explains why CGA has a higher possibility of convergence to a suboptimal solution. To address the limitations of CGA, we propose novel algorithms by incorporating the self-adaptivity capability of individuals, which is an essential step observed in the natural world, called Adaptive Genetic Algorithm (AGA) and Self-Informed Genetic Algorithm (SIGA). To be specific, the individual's chromosomes in a population will conduct a self-examination on the efficiency of all the wind turbines, and thus gaining self-awareness on which part of the solution is currently the bottleneck for further improvement. In order to relocate the worst turbine, we first propose to relocate the worst turbine randomly with AGA, and then an improved version called SIGA is developed with information guided relocation to find a good location using a surrogate model from Multivariate Adaptive Regression Splines (MARS) regression based on Monte Carlo Simulation. Extensive numerical results under multiple wind distributions and different wind farm sizes illustrate the improved efficiency of SIGA and AGA over CGA. In the end, an open-source Python package is made available on github (https://github.com/JuXinglong/WFLOP_Python).

Aplicaciones de los EAs

- Construcción de árboles filogenéticos
  - Reconstruir historia evolutiva en base a comparación de ADN de distintas especies
  - Construir el árbol que maximice la parsimonia (simpleza de la hipótesis)



A Genetic Algorithm for Maximum-Likelihood Phylogeny Inference Using Nucleotide Sequence Data

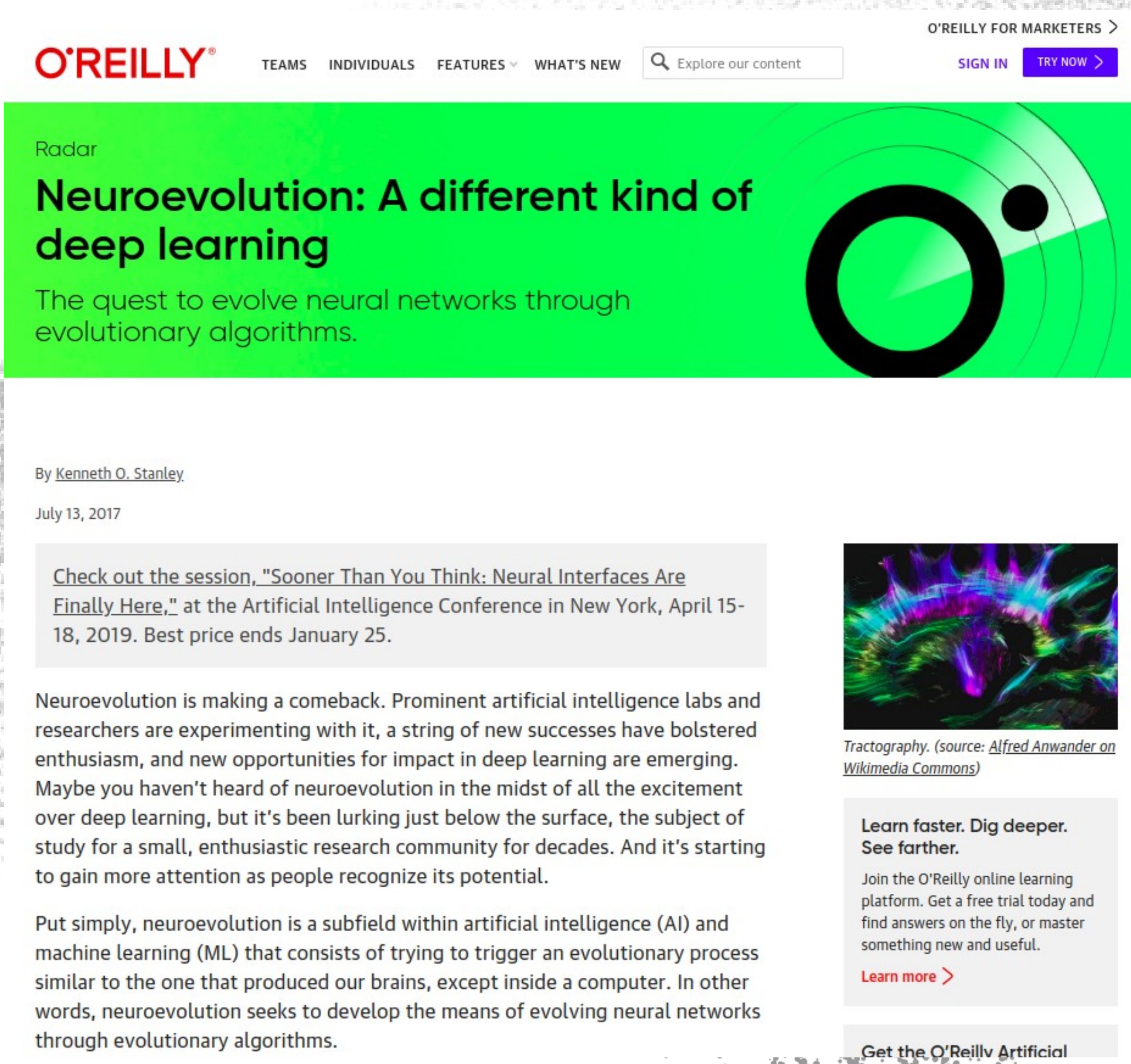
Paul O. Lewis

Department of Biology, University of New Mexico

Phylogeny reconstruction is a difficult computational problem, because the number of possible solutions increases with the number of included taxa. For example, for only 14 taxa, there are more than seven trillion possible unrooted phylogenetic trees. For this reason, phylogenetic inference methods commonly use clustering algorithms (e.g., the neighbor-joining method) or heuristic search strategies to minimize the amount of time spent evaluating nonoptimal trees. Even heuristic searches can be painfully slow, especially when computationally intensive optimality criteria such as maximum likelihood are used. I describe here a different approach to heuristic searching (using a genetic algorithm) that can tremendously reduce the time required for maximum-likelihood phylogenetic inference, especially for data sets involving large numbers of taxa. Genetic algorithms are simulations of natural selection in which individuals are encoded solutions to the problem of interest. Here, labeled phylogenetic trees are the individuals, and differential reproduction is effected by allowing the number of offspring produced by each individual to be proportional to that individual's rank likelihood score. Natural selection increases the average likelihood in the evolving population of phylogenetic trees, and the genetic algorithm is allowed to proceed until the likelihood of the best individual ceases to improve over time. An example is presented involving *rbc*L sequence data for 55 taxa of green plants. The genetic algorithm described here required only 6% of the computational effort required by a conventional heuristic search using tree bisection/reconnection (TBR) branch swapping to obtain the same maximum-likelihood topology.

Proyección de los EAs

- O'Reilly
- **Neuroevolution**
  - Evolucionar la arquitectura de redes neuronales artificiales



O'REILLY® TEAMS INDIVIDUALS FEATURES WHAT'S NEW Explore our content O'REILLY FOR MARKETERS SIGN IN TRY NOW

Radar

Neuroevolution: A different kind of deep learning

The quest to evolve neural networks through evolutionary algorithms.

By Kenneth O. Stanley

July 13, 2017

Check out the session, "Sooner Than You Think: Neural Interfaces Are Finally Here," at the Artificial Intelligence Conference in New York, April 15-18, 2019. Best price ends January 25.

Neuroevolution is making a comeback. Prominent artificial intelligence labs and researchers are experimenting with it, a string of new successes have bolstered enthusiasm, and new opportunities for impact in deep learning are emerging. Maybe you haven't heard of neuroevolution in the midst of all the excitement over deep learning, but it's been lurking just below the surface, the subject of study for a small, enthusiastic research community for decades. And it's starting to gain more attention as people recognize its potential.

Put simply, neuroevolution is a subfield within artificial intelligence (AI) and machine learning (ML) that consists of trying to trigger an evolutionary process similar to the one that produced our brains, except inside a computer. In other words, neuroevolution seeks to develop the means of evolving neural networks through evolutionary algorithms.

Tractography. (source: Alfred Anwander on Wikimedia Commons)

Learn faster. Dig deeper. See farther.

Join the O'Reilly online learning platform. Get a free trial today and find answers on the fly, or master something new and useful.

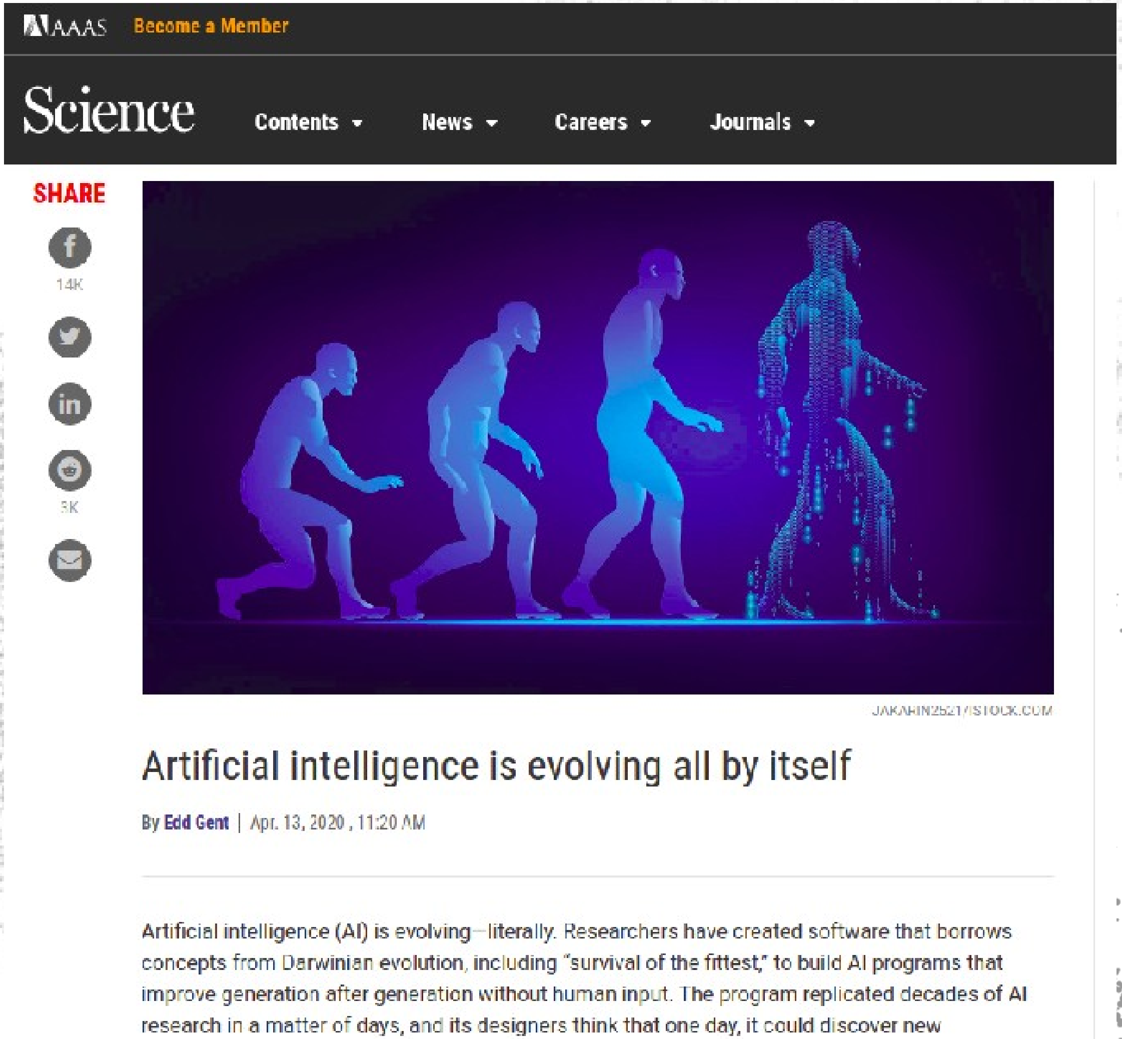
Learn more

Get the O'Reilly Artificial

https://www.oreilly.com/radar/neuroevolution-a-different-kind-of-deep-learning/

Proyección de los EAs

- Science Magazine: AI is evolving



AAAS Become a Member

Science Contents News Careers Journals

SHARE

14K

3K

JAKARIN2521/ISTOCK.COM

Artificial intelligence is evolving all by itself

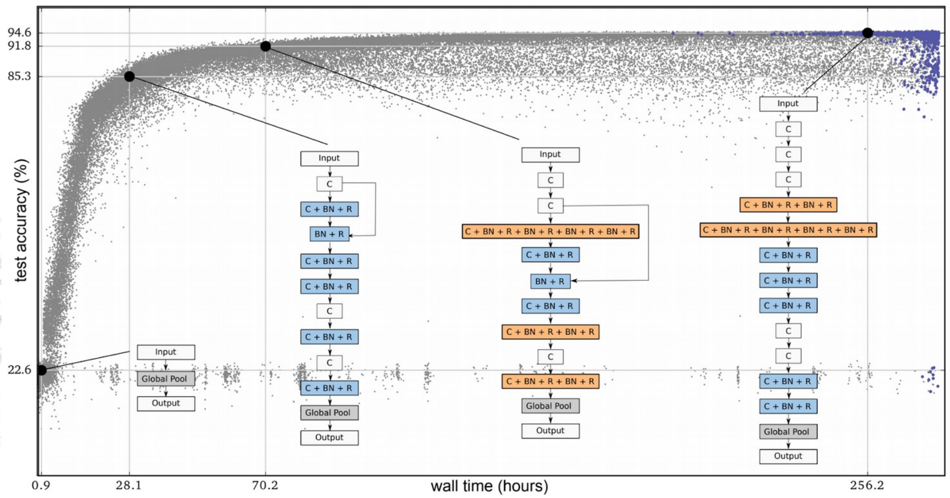
By Edd Gent | Apr. 13, 2020 , 11:20 AM

Artificial intelligence (AI) is evolving—literally. Researchers have created software that borrows concepts from Darwinian evolution, including "survival of the fittest," to build AI programs that improve generation after generation without human input. The program replicated decades of AI research in a matter of days, and its designers think that one day, it could discover new

https://www.sciencemag.org/news/2020/04/artificial-intelligence-evolving-all-itself

Proyección de los EAs

- **Arquitecturas de redes neuronales**
  - Crece interés en redes neuronales (deep learning)
  - Necesidad de automatizar el diseño de su estructura
  - Maximizar exactitud (accuracy)



Google AI Blog

The latest news from Google AI

Using Evolutionary AutoML to Discover Neural Network Architectures

Thursday, March 15, 2018

Posted by Esteban Real, Senior Software Engineer, Google Brain Team

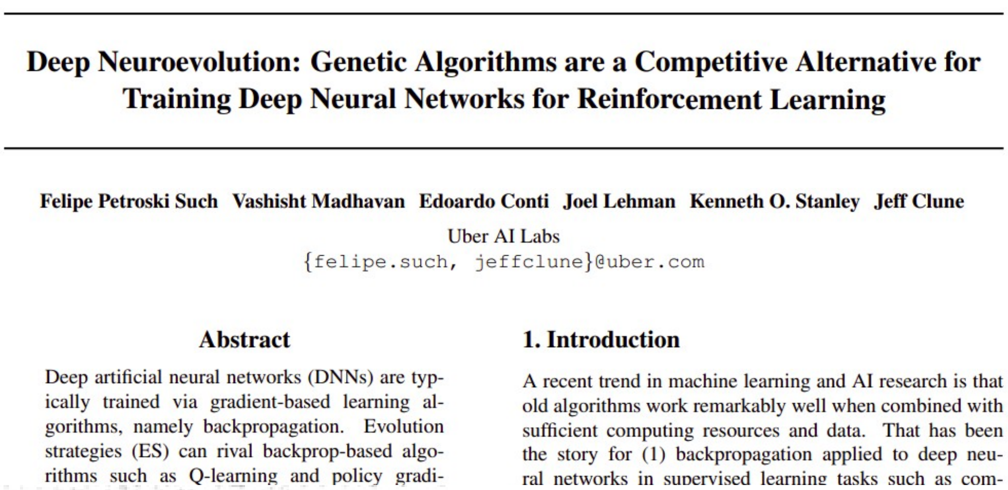
The brain has evolved over a long time, from very simple worm brains 500 million years ago to a diversity of modern structures today. The human brain, for example, can accomplish a wide variety of activities, many of them effortlessly — telling whether a visual scene contains animals or buildings feels trivial to us, for example. To perform activities like these, artificial neural networks require careful design by experts over years of difficult research, and typically address one specific task, such as to find what's in a photograph, to call a genetic variant, or to help diagnose a disease. Ideally, one would want to have an automated method to generate the right architecture for any given task.

One approach to generate these architectures is through the use of evolutionary algorithms. Traditional research into neuro-evolution of topologies (e.g. Stanley and Miikkulainen 2002) has laid the foundations that allow us to apply these algorithms at scale today, and many groups are working on the subject, including OpenAI, Uber Labs, Sentient Labs and DeepMind. Of course, the Google Brain team has been thinking about AutoML too. In addition to learning-based approaches (eg. reinforcement learning), we wondered if we could use our computational resources to programmatically *evolve* image classifiers at unprecedented scale. Can we achieve solutions with minimal expert participation? How good can today's artificially-evolved neural networks be? We address these questions through two papers.

https://ai.googleblog.com/2018/03/using-evolutionary-automl-to-discover.html

Proyección de los EAs

- **Neuroevolution** optimización de redes neuronales mediante algoritmos evolutivos



Deep Neuroevolution: Genetic Algorithms are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning

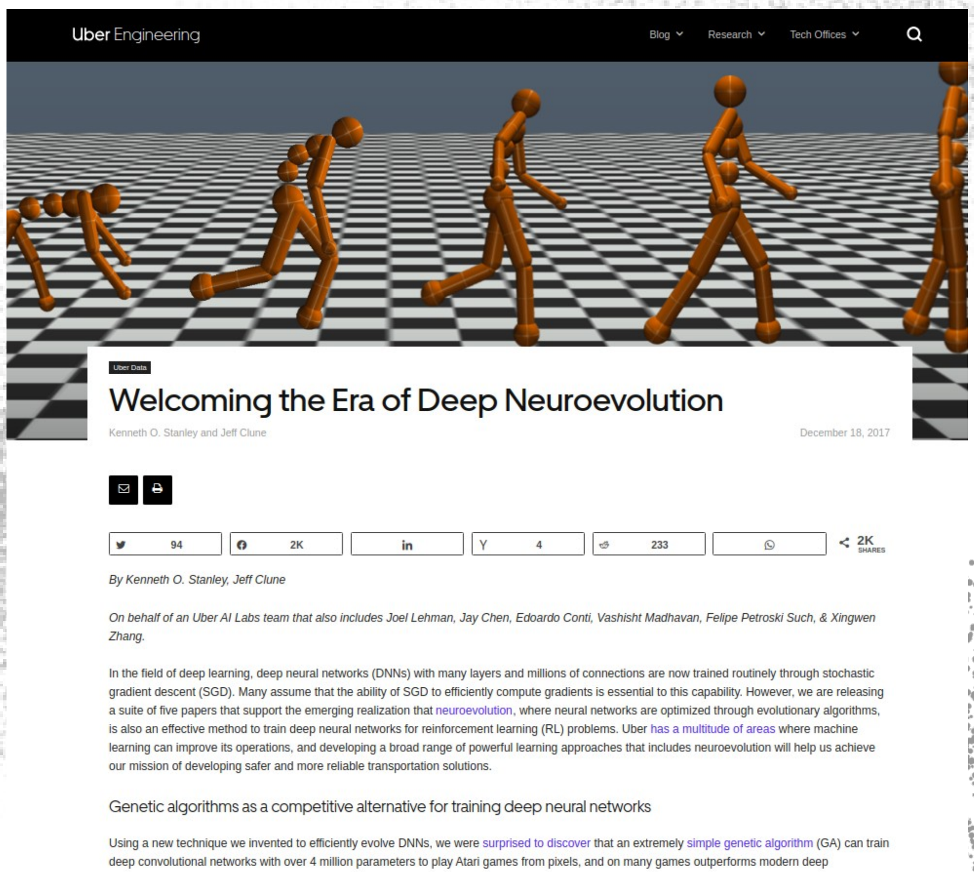
Felipe Petroski Such Vashisht Madhavan Edoardo Conti Joel Lehman Kenneth O. Stanley Jeff Clune
Uber AI Labs
{felipe.such, jeffclune}@uber.com

Abstract

Deep artificial neural networks (DNNs) are typically trained via gradient-based learning algorithms, namely backpropagation. Evolution strategies (ES) can rival backprop-based algorithms such as Q-learning and policy gradi-

1. Introduction

A recent trend in machine learning and AI research is that old algorithms work remarkably well when combined with sufficient computing resources and data. That has been the story for (1) backpropagation applied to deep neural networks in supervised learning tasks such as com-



Uber Engineering Blog Research Tech Offices

Uber Data

Welcoming the Era of Deep Neuroevolution

Kenneth O. Stanley and Jeff Clune December 18, 2017

94 2K in Y 4 233 2K SHARES

By Kenneth O. Stanley, Jeff Clune

On behalf of an Uber AI Labs team that also includes Joel Lehman, Jay Chen, Edoardo Conti, Vashisht Madhavan, Felipe Petroski Such, & Xingwen Zhang.

In the field of deep learning, deep neural networks (DNNs) with many layers and millions of connections are now trained routinely through stochastic gradient descent (SGD). Many assume that the ability of SGD to efficiently compute gradients is essential to this capability. However, we are releasing a suite of five papers that support the emerging realization that neuroevolution, where neural networks are optimized through evolutionary algorithms, is also an effective method to train deep neural networks for reinforcement learning (RL) problems. Uber has a multitude of areas where machine learning can improve its operations, and developing a broad range of powerful learning approaches that includes neuroevolution will help us achieve our mission of developing safer and more reliable transportation solutions.

Genetic algorithms as a competitive alternative for training deep neural networks

Using a new technique we invented to efficiently evolve DNNs, we were surprised to discover that an extremely simple genetic algorithm (GA) can train deep convolutional networks with over 4 million parameters to play Atari games from pixels, and on many games outperforms modern deep

https://eng.uber.com/deep-neuroevolution/

Proyección de los EAs

- **Cognizant**
  - Compañia de servicios TI
  - NASDAQ-100
  - Fortune 500
- Generaron LEAF (Learning Evolutionary Algortihm Framework) para optimización de decisiones de negocio

https://www.cognizant.com/ai/evolutionary-ai