Evolutionary computing

Brandon Marquez Salazar

July 1, 2025

Evolutionary programming

Following what John R. Koza said in his essay Human-Competitive Machine Intelligence [1], points that evolutionary algorithms can be used in order to make a computer program to solve (approximately) a problem.

The problem, the model and the solution I

The problem, the model and the solution II

Think about a problem. Usually, to solve some problem, we can solve it using tools that are already available. For example, if we want to know the position of a particle at a certain time in an uniform rectilinear motion problem, we can solve it by a simple equation.

$$X_f(t) = X_0 + V_0 t$$

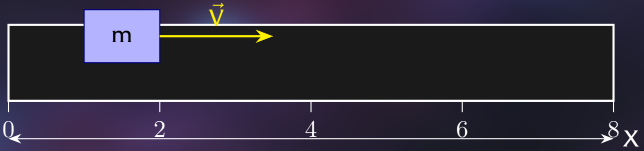
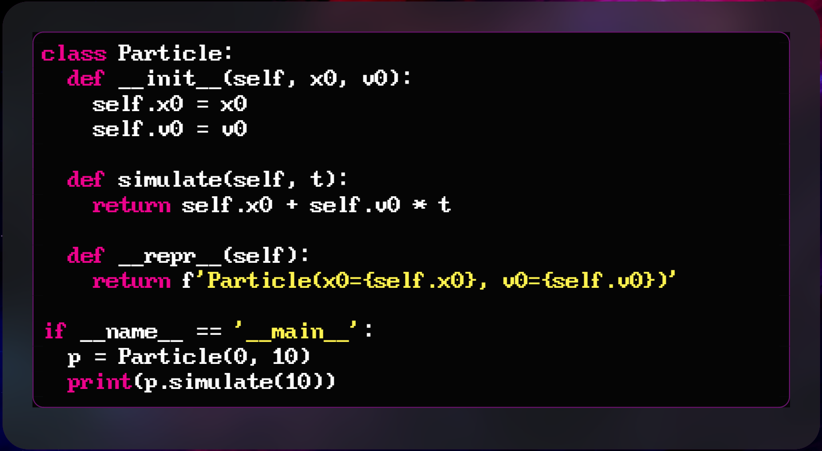


Figure: Uniform rectilinear motion [Deepseek]

The problem, the model and the solution III

And we can call it a simulation, which is the most common thing, and easiest thing to do with a computer.

The problem, the model and the solution IV



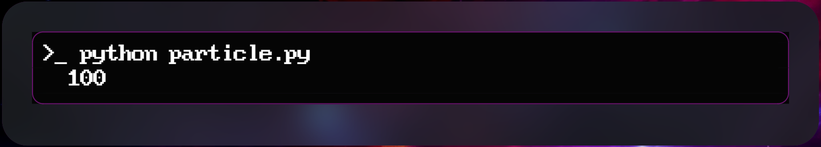
```python
class Particle:
  def __init__(self, x0, v0):
    self.x0 = x0
    self.v0 = v0

  def simulate(self, t):
    return self.x0 + self.v0 * t

  def __repr__(self):
    return f'Particle(x0={self.x0}, v0={self.v0})'

if __name__ == '__main__':
  p = Particle(0, 10)
  print(p.simulate(10))
```

The problem, the model and the solution V



```
>_ python particle.py
  100
```

The problem, the model and the solution VI

But now, what happens if we haven't an equation or an algorithm to solve the problem? If have data and we need to know how it's behaving? What if we need to predict it's behaviour? Maybe some nonlinear phenomena, a signal, classification, or something not so intuitive as common problems?

Here is where modeling comes in [3]. And here is where genetic programming started to gain relevance.

Modeling I

Genetic programming is an implementation of genetic algorithms where the individuals are programs which are intended to solve a given problem

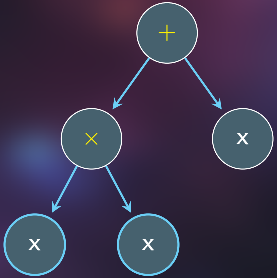


Figure: Genetic programming solution tree for $x^2 + x$ [Deepseek]

Modeling II

Aquí se muestra un ejemplo de la implementación de un algoritmo genético para programación genética utilizando el framework DEAP [2].

- Ejemplo HTML
- Ejemplo PDF
- Ejemplo Jupyter
- Ejemplo Python

References I

[1] Santa Fe Institute Editorial Board, ed. Perspectives on Adaptation in Natural and Artificial Systems. 2003.

[2] DEAP. DEAP Genetic Programming. URL: `https://deap.readthedocs.io/en/master/tutorials/advanced/gp.html` (visited on 05/29/2025).

[3] A.E. Eiben and J.E. Smith. Introduction to Evolutionary Computing. Springer Berlin Heidelberg, 2015. ISBN: 9783662448748. DOI: `10.1007/978-3-662-44874-8`. URL: `http://dx.doi.org/10.1007/978-3-662-44874-8`.

[4] Félix-Antoine Fortin; François-Michel De Rainville; Marc-André Gardner; Marc Parizeau; Christian Gagné. In: Journal of Machine Learning Research 2171-2175 (2012).

References II

[5] One Max Problem — DEAP 1.4.3 documentation. URL: `https://deap.readthedocs.io/en/master/examples/ga_onemax.html` (visited on 05/29/2025).

[6] PytLab. gaft github repository example. URL: `https://github.com/PytLab/gaft/blob/master/examples/ex01/ex01.py`.

[7] García Carlos Hugo Ramírez Guillermo Trujillo-Romero Felipe. “Comparativa de frameworks para cómputo evolutivo implementados en Python.” In: Congreso Internacional de Mecatrónica, Control e Inteligencia Artificial, UNAM (2023).

[8] Zhengjiang Shao. GAFT Documentation, Release GAFT. 2018. URL: `https://gaft.readthedocs.io/_/downloads/en/latest/pdf/` (visited on 05/29/2025).