Biologically-Inspired Computing I: Optimization

CSCI 4360/6360 Data Science II

What is Optimization?

$$\theta^* \in \arg\min_{\theta \in \Theta} \mathcal{L}(\theta)$$

- Formal: “Find a specific value of theta for which L is minimized”
- Colloquial: “Give me the parameters that result in the function’s minimum”

Tried and true

- Where do we start?

$$\frac{\partial}{\partial a_1} f(a_1, a_2)$$

- Partial derivative
- Set partial to zero
- Solve for the minimum

- **Why doesn't this work in modern ML?** (even logistic regression!)

Automatic Differentiation

- Aka, “autodiff”
- Core to TensorFlow, Keras, PyTorch

AUTOMATIC DIFFERENTIATION WITH `TORCH.AUTOGRAD`

- Requires a computational graph
  - Computes gradients during backpropagation

Stochastic Differentiation

- aka, Stochastic Gradient Descent (SGD)!

$$\mathbb{E}\left[\boldsymbol{g}_t\right] = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})|_{\boldsymbol{\theta}_t}$$

- The source of noise in this case: the data (or lack thereof)
  - Partials w.r.t. a sample, or even a single data point
- Other versions to reduce the noise
  - Preconditioned SGD
  - Variance reduction

What do all these methods require?

- Plenty of other optimization strategies along these lines
- One thing they all share: **require derivatives**
  - Requires the function we're optimizing L to have an explicit or known form
  - Requires evaluation of the derivative to be fairly inexpensive
- Are there alternatives? **Yes**

Derivative-free Optimization (DFO)

- Hill-climbing
- Stochastic local search
- Random search
- Optimal transport (e.g., Wasserstein)

- **Today**
    - Simulated Annealing
    - Evolutionary Algorithms
    - Particle Swarm

Annealing

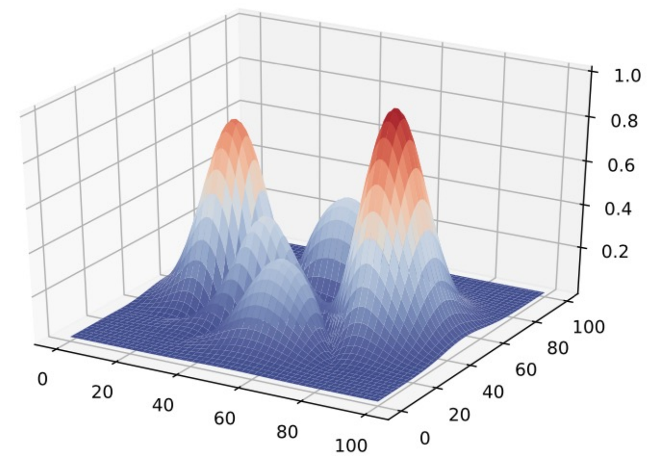
- Physical process of heating a solid until thermal stresses are released, then cooling it (very slowly) until crystals are perfectly arranged
  - Corresponds to a **minimum energy state**
- Define an *energy function*

$$p_T(\boldsymbol{x}) = \exp(-\mathcal{E}(\boldsymbol{x})/T)$$

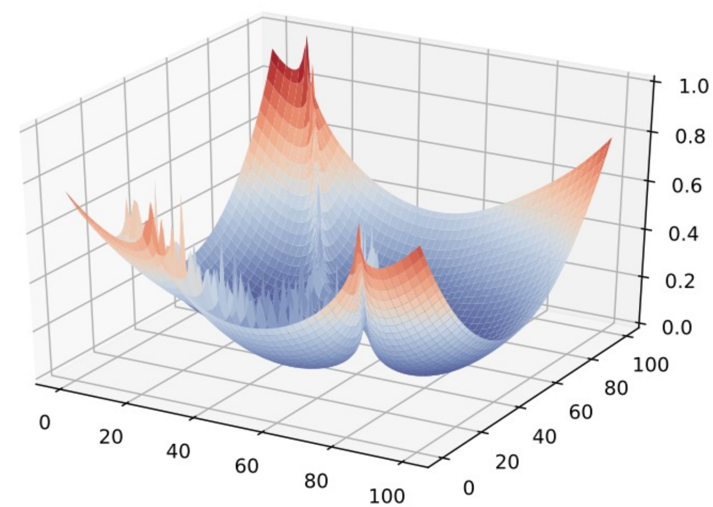
- Temperature *T* is slowly decreased over time

Annealing

• A function we want to optimize

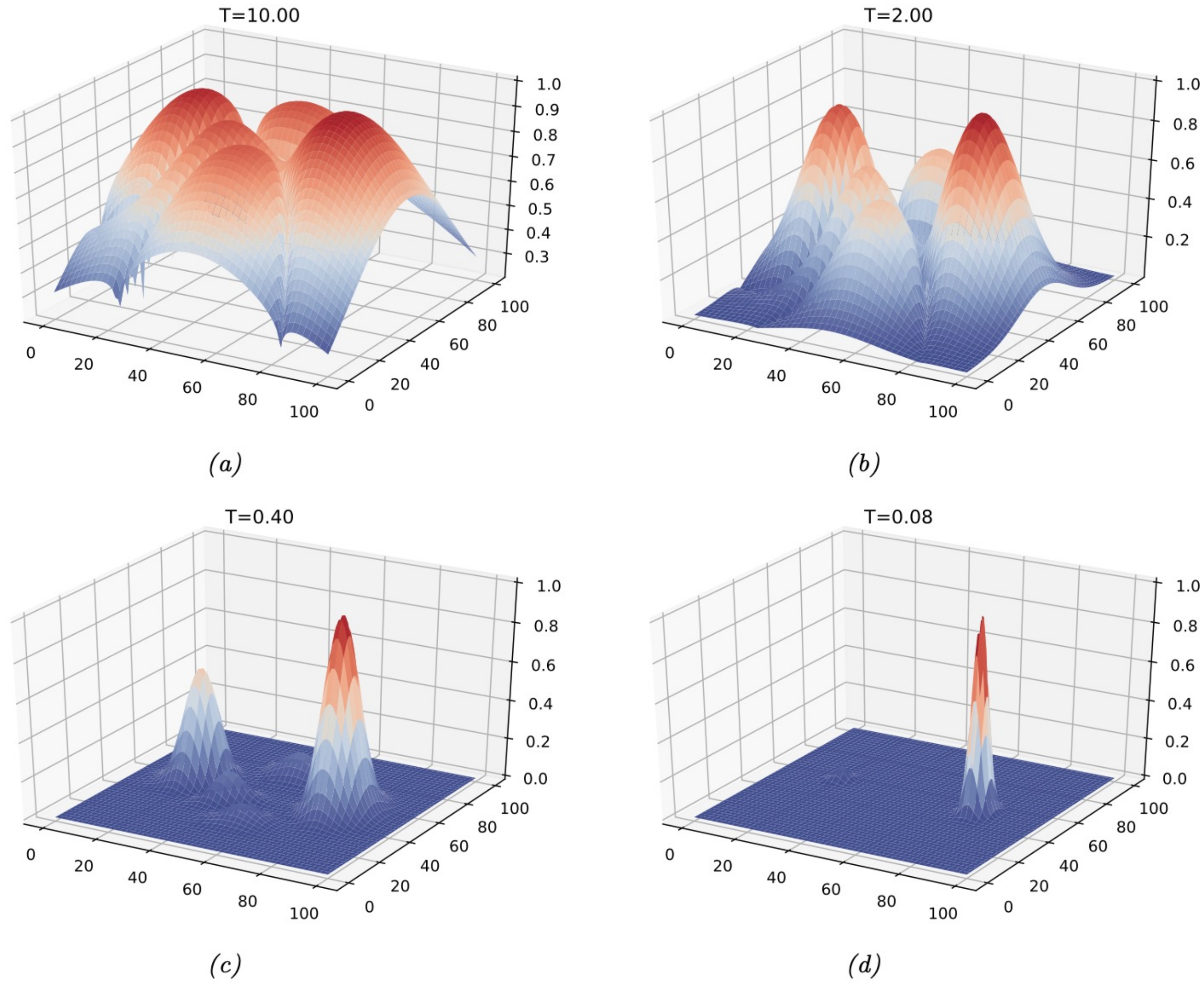


• Its corresponding energy function



Annealing

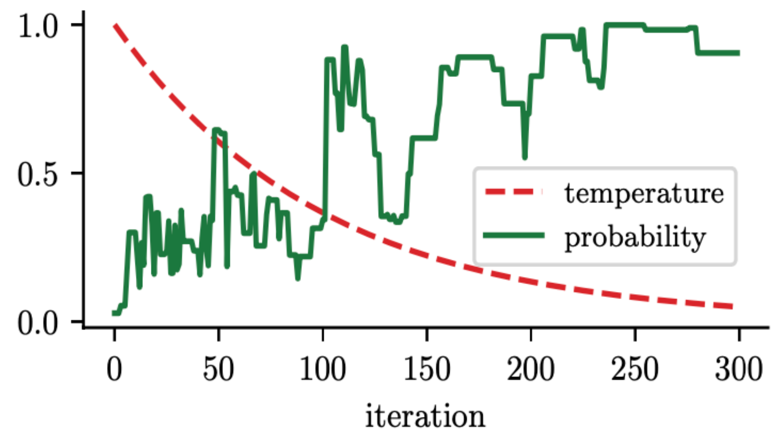
- Annealed versions of the energy function at different temperatures
  - T >>> 1, energy landscape is flatter
  - As T -> 0, landscape becomes sharper, highlighting high-probability global extrema



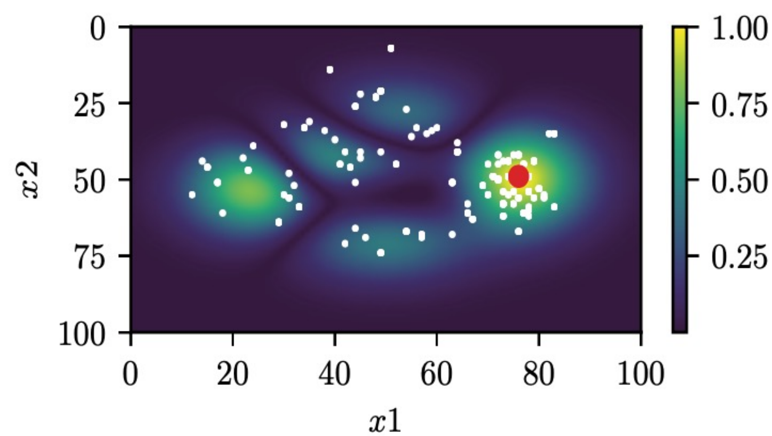
(a) (b) (c) (d)

Annealing

- The longer the experiment, the more confident the annealing will be



- (Theoretically) Guaranteed to final global optimum
  - Run long enough
  - Good cooling schedule
  - etc



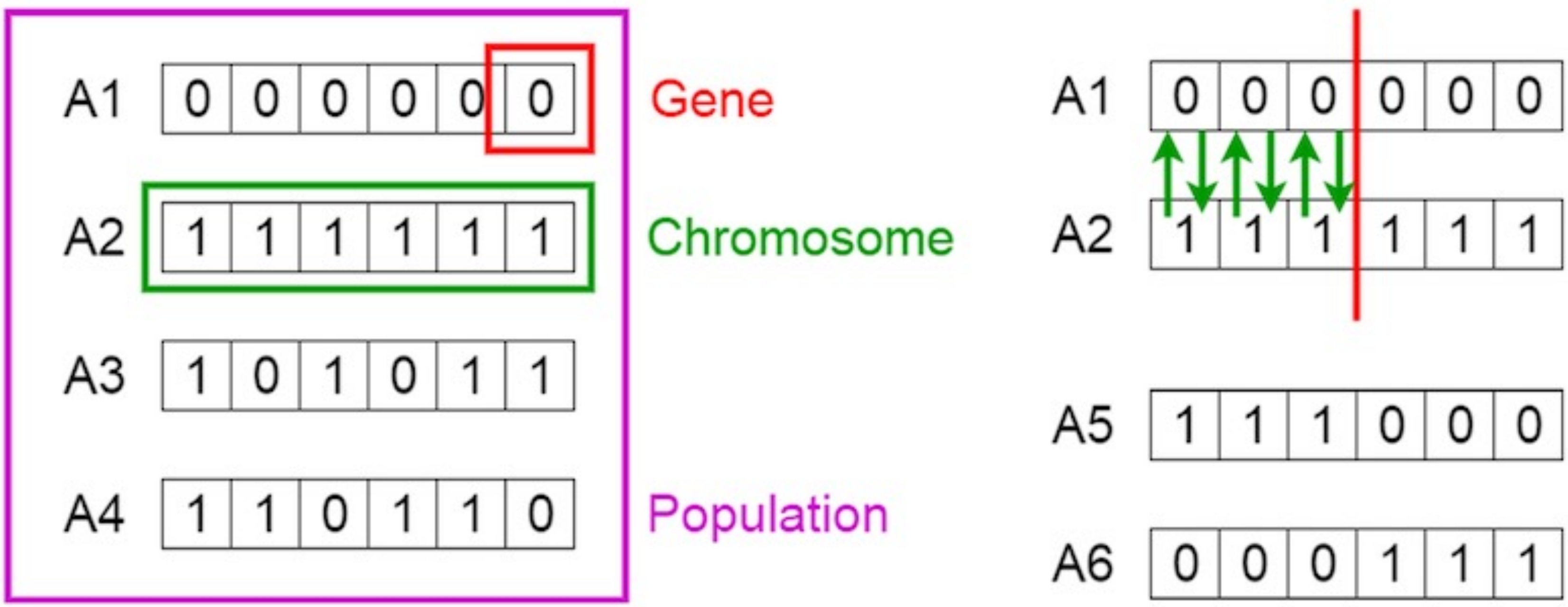
Evolutionary Algorithms (EA)

- Annealing = (technically) guaranteed to find global optimum
- EA = (technically) NOT guaranteed to find global optimum

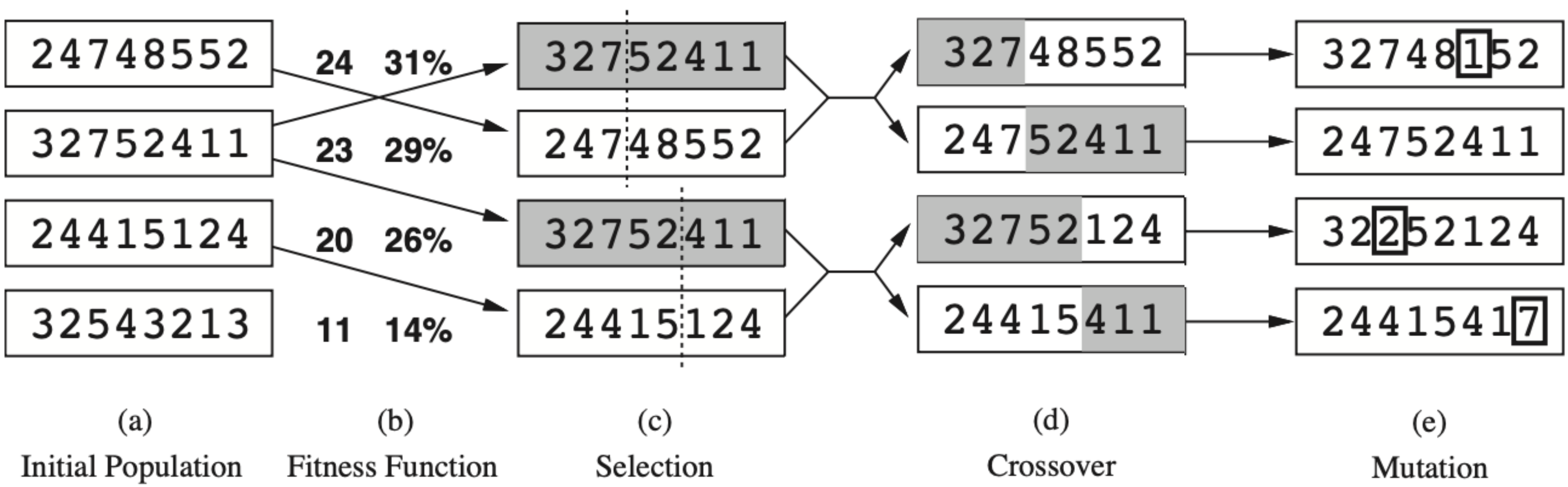
- Lots of caveats, on both

- EA is a form of “stochastic local search”
  - Balance exploitation (local search) and exploration (global search)
  - Can find you a good local optimum quickly, with good chance of global optimum in a reasonable time frame

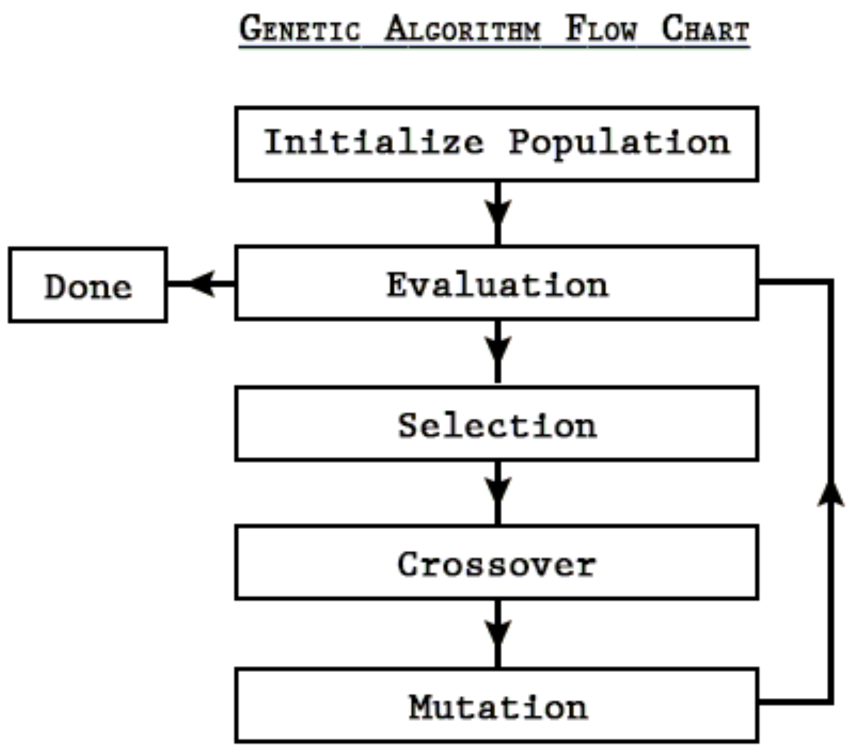
Evolutionary / Genetic Algorithms



Evolutionary / Genetic Algorithms



Evolutionary / Genetic Algorithms



Evolutionary / Genetic Algorithms

- One way to conceptualize EAs/GAs: search!

- Goal: find an optimal (or nearly-optimal) parameter combination *without* having to evaluate all possible parameter values

- GA trade-offs exploration vs exploitation through population size, number of generations, cross-over, mutation

Particle Swarm Optimization (PSO)

- The inspiration comes from watching large swarms of birds or schools of fish moving somewhat in unison, but with a few members taking some unexpected deviations.
- A single particle in PSO compares well to a single individual in EA

PSO: Exploration vs Exploitation

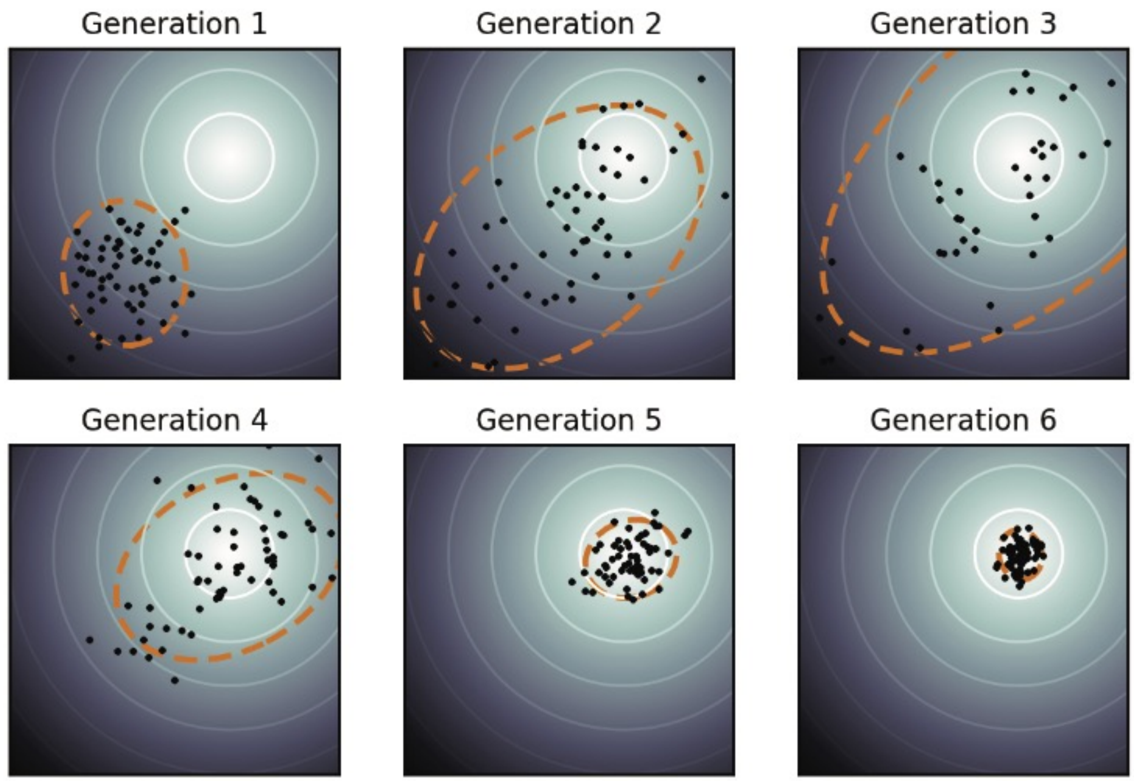
- *Exploration*: breadth over depth
  - Trying out a large range of values quickly
  - “Building an intuition” phase
- *Exploitation:* depth over breadth
  - Zeroing in on a high-probability area
  - “Deep dive” phase

- **Pretty much every optimization strategy involves some trade-off between these two**
  - If you notice your optimization procedure starting fast and then slowing down, it’s shifting from exploration into exploitation

Particle Swarm Optimization

- Like EA, you have a population of particles
- Unlike EA, these particles remain the same
  - Each particle tracks its own search progress
  - Some global parameters to track
  - Hyperparameters modulating the exploration/exploitation trade-off
- Some stochasticity (analogous to mutation) to prevent getting stuck
- Simulate

Particle Swarm Optimization



Advantages of Derivative-free methods

- Can optimize pretty much anything
  - No need for a known or closed form derivative (“black box optimization”)
  - Just need some way of evaluating whether or not a specific guess is “good” (e.g., a fitness function)
- Straightforward to implement
  - You implemented part of PSO in the midterm, EA in HW4 ☺
  - Compare to your HW1 implementation of gradient descent
- Only real constraint (usually) is time
  - Fairly resource-light, can scale up to use available resources

Disadvantages of Derivative-free methods

- Often no convergence guarantees
  - EA is guaranteed to find a local optimum
  - Annealing is *theoretically* guaranteed to find a global optimum (but could be waiting until the heat death of the universe)
- Relies heavily on hand-tuned hyperparameters
  - Temperature protocol, mutation rate, cognitive / social parameters
- ”Long tail” convergence
  - Can usually find a decent solution quickly, but optimal solutions may take a very long time

Stay tuned

- More biologically inspired computing methods: **neural networks!**

References

- "Probabilistic Machine Learning", by Kevin Murphy
  https://probml.github.io/pml-book/
  - Book 2: "Probabilistic Machine Learning: Advanced Topics", ch. 6

Quick Notes

- Still finishing HW3 and Midterm grades
  - Aiming to have these on eLC **tomorrow**
- All workshop presentations and final project proposals are on eLC!
  - Double-check to make sure you have the grade you expect

- **Final Project Update #1 is due TUESDAY, Oct 31**
- **Homework 4 is due TUESDAY, Oct 31**