Nature-inspired Algorithms

Lecture 1

Algorithm Engineering Group
Hasso Plattner Institute, University of Potsdam

24 April 2017

Search and Optimization

Search

Given: a set $\mathcal{X}$ and $S \subseteq \mathcal{X}$,
Find: $x \in S$

Optimization

Given: a set $\mathcal{X}$ and $f \colon \mathcal{X} \to \mathbb{R}$,
Find: x such that for all $y \in \mathcal{X}$, $f(x) \geq f(y)$

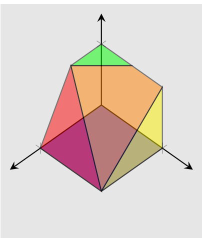
Examples

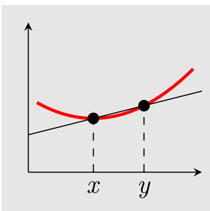
Search

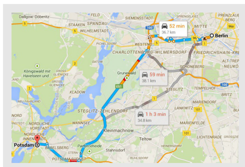
- Given a list of numbers $L = \{\ell_1, \ell_2, \ldots, \ell_n\}$, find a prime number in L.
- Given an undirected graph $G = (V, E)$, find a Hamiltonian path in G

Optimization

- Given vectors c, b and a matrix A, find x with $Ax \leq b$ and $c^\top x$ maximal.
- Given a convex function f find the minimum
- Given a network of routes, find the shortest path







Algorithm

Definition

A step-by-step procedure for solving a problem or accomplishing some end especially by a computer

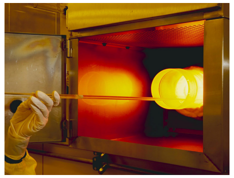
A *well-defined* computational process

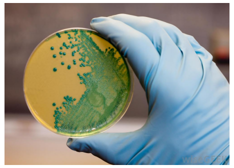
input $\longrightarrow$ $\boxed{\text{transformation}}$ $\longrightarrow$ output

Nature-inspired algorithm

Natural processes as computational processes

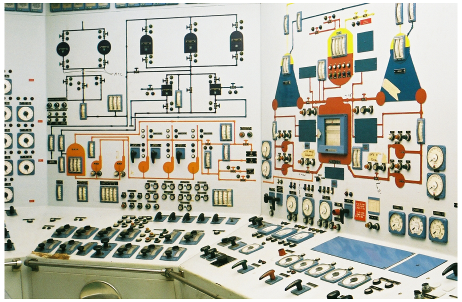
input → $\boxed{\text{transformation}}$ → output

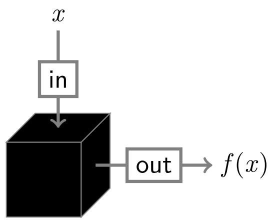


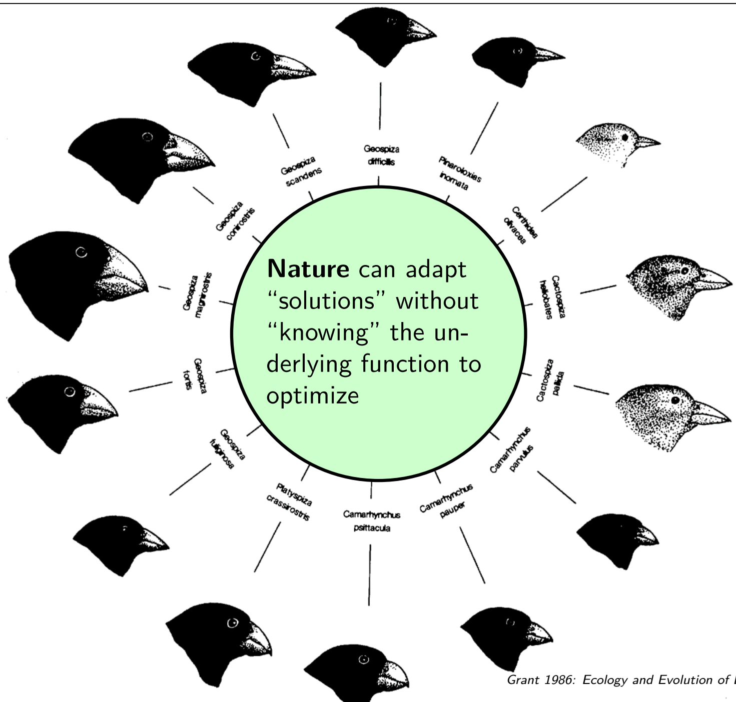


Why consider such algorithms?

Suppose we know **nothing** (or almost nothing) *a priori* about a function to optimize

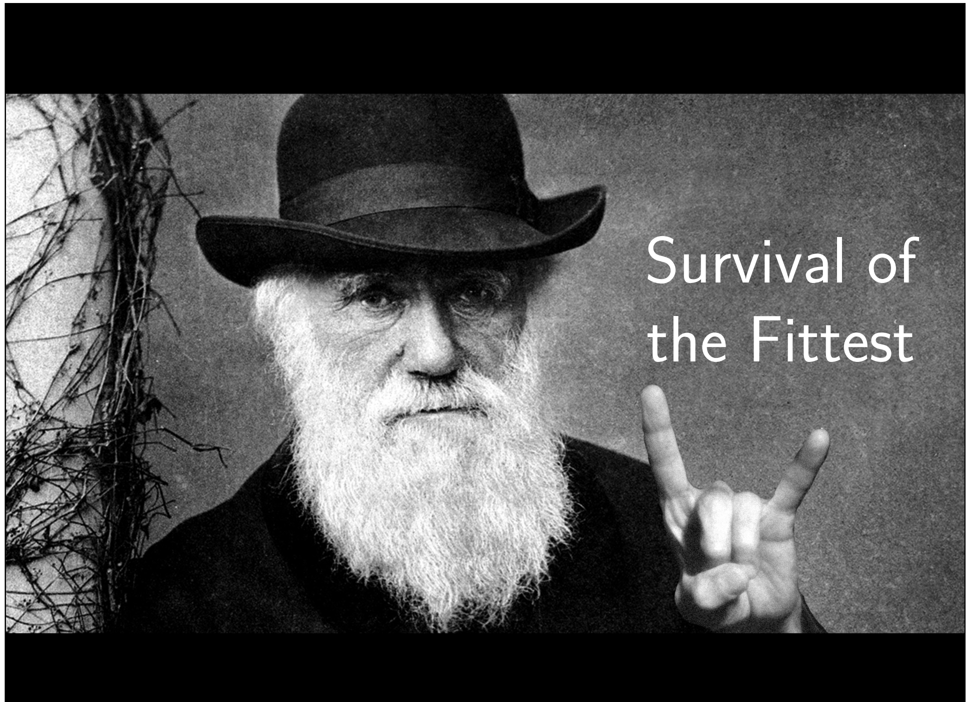






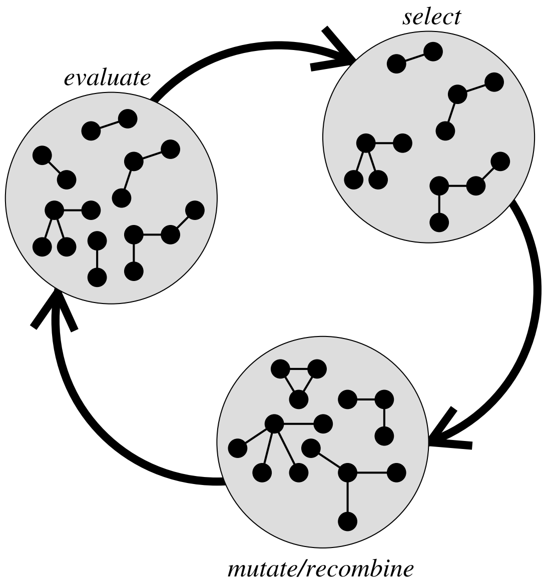
Nature can adapt "solutions" without "knowing" the underlying function to optimize

Grant 1986: Ecology and Evolution of Darwin's finches



Survival of the Fittest

Evolutionary Algorithms



Some success stories

NASA

- communication antennas on ST-5 mission (evolutionary algorithm)
- deployed on spacecraft in 2006

REFERENCE: Jason D. Lohn, Gregory S. Hornby and Derek S. Linden, "Human-competitive evolved antennas", *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, volume 22, issue 3, pages 235–247 (2008).

Some success stories

Boeing

- 777 GE engine: turbine geometry evolved with a genetic algorithm

REFERENCE: Charles W. Petit, "Touched by nature: putting evolution to work on the assembly line." *US News & World Report*, volume 125, issue 4, pages 43–45 (1998).

Some success stories

Oral B

- cross-action toothbrush design optimized by Creativity Machine (evolutionary algorithm)

REFERENCE: Robert Plotnick, "The Genie in the Machine: How Computer-Automated Inventing Is Revolutionizing Law and Business", Stanford Law Books, (2009)

Some success stories

Nutech Solutions

- improved car frame for GM (genetic algorithms, neural networks, simulated annealing, swarm intelligence)

BMW

- optimized acoustic and safety parameters in car bodies (simulated annealing, genetic and evolutionary algorithms)

REFERENCE: Fabian Duddeck, "Multidisciplinary Optimization of Car Bodies", *Structural and Multidisciplinary Optimization*, volume 35, pages 375–389 (2008).

Some success stories

Hitachi

- improved nose cone for N700 bullet train (genetic algorithm)

REFERENCE: Takenori Wajima, Masakazu Matsumoto and Shinichi Sekino, "Latest System Technologies for Railway Electric Cars", *Hitachi Review* volume 54, issue 4, pages 161–168 (2005).

Some success stories

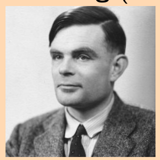
Merck Pharmaceutical Company

- discovered first clinically-approved antiviral drug for HIV (Isentress) using AutoDock software (uses a genetic algorithm)

REFERENCE: `http://autodock.scripps.edu/news/autodocks-role-in-developing-the-first-clinically-approved-hiv-integrase-inhibitor`

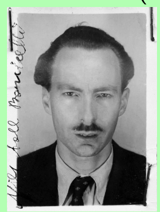
Evolution and Computing

Alan Turing (1948)



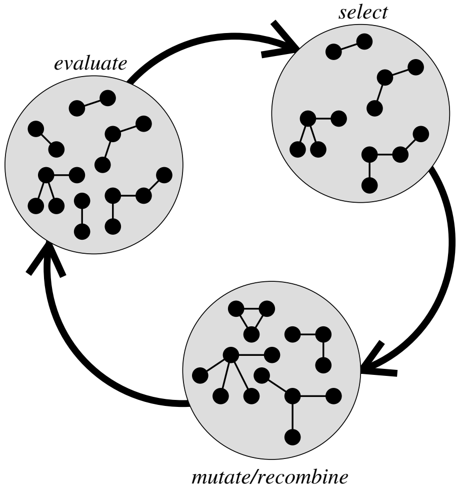
proposed a "learning machine" that would parallel the principles of evolution

Nils A. Baricelli (1950s)



computer simulation of evolution at Institute for Advanced Study, Princeton NJ

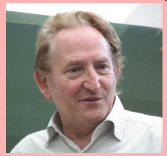
Darwinian principles for automated problem solving



Lawrence J. Fogel (1960s)

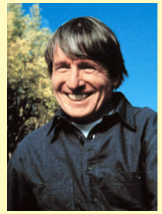
Evolutionary Programming: simulated evolution as a learning process to generate AI

Ingo Rechenberg and **Hans-Paul Schwefel** (1960s, 1970s)



Evolution Strategies: solving complex engineering problems

John Henry Holland (1970s)



Genetic Algorithm, evolving cellular automata

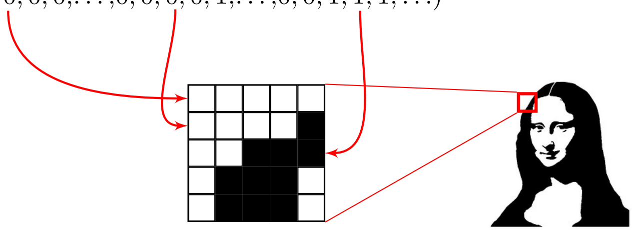
Simple example

Evolving "bitstrings": strings that look like $(1, 1, 0, 0, 1, 0, \ldots, 0, 1)$

Popular choice, as everything is represented by a bitstring in a computer

Our example: evolving pictures

$(0, 0, 0, 0, 0, \ldots, 0, 0, 0, 0, 1, \ldots, 0, 0, 1, 1, 1, \ldots)$



Example

Domain: 256×120 pixel images.

Each picture is a string of $n = 30720$ bits

Fitness is Hamming distance to a target picture t:

$$f(x) = |\{i : x[i] \neq t[i]\}|.$$

GeneticAlgorithm(m, n, p_r)

$P_0 \leftarrow$ a set of m random strings of length n;
for $t \leftarrow 0$ **to** ∞ **do**
  Select $y, z \in P_t$ u.a.r.;
  if $\textsc{Random}() \leq p_r$ **then** $x \leftarrow \textsc{Mutate}(\textsc{Crossover}(y, z))$;
  else $x \leftarrow \textsc{Mutate}(y)$;
  Worst individual in $P_t \cup \{x\}$ dies off to create P_{t+1};
end

General-purpose algorithms for optimization

Assumptions

1. Solutions encoded as length-n bitstrings (elements of $\{0,1\}^n$),
2. want to maximize some $f\colon \{0,1\}^n \to \mathbb{R}$.

Random Search

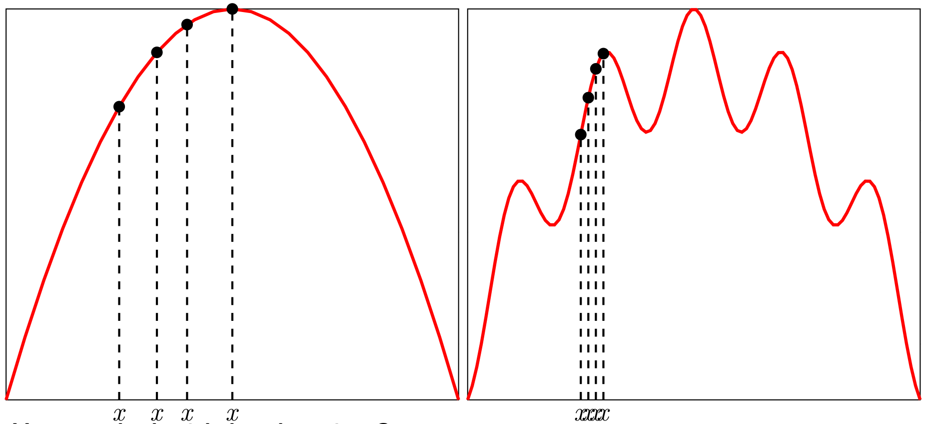
Choose x uniformly at random from $\{0,1\}^n$;
while *stopping criterion not met* **do**
  Choose y uniformly at random from $\{0,1\}^n$;
  if $f(y) \geq f(x)$ **then** $x \leftarrow y$;
end

General-purpose algorithms for optimization

Random(ized) Local Search (RLS)

Choose x uniformly at random from $\{0,1\}^n$;
while *stopping criterion not met* **do**
  $y \leftarrow x$;
  Choose i uniformly at random from $\{1,\ldots,n\}$;
  $y_i \leftarrow (1 - y_i)$;
  if $f(y) \geq f(x)$ **then** $x \leftarrow y$;
end

Local Optima



How to deal with local optima?

- Restart the process when it becomes trapped (ILS)
- Accept disimproving moves (MA, SA)
- Take larger steps (EA, GA)

General-purpose algorithms for optimization

Metropolis Algorithm

Choose x uniformly at random from $\{0,1\}^n$;
while *stopping criterion not met* **do**
  $y \leftarrow x$;
  Choose i uniformly at random from $\{1,\ldots,n\}$;
  $y_i \leftarrow (1 - y_i)$;
  if $f(y) \geq f(x)$ **then** $x \leftarrow y$;
  else $x \leftarrow y$ with probability $e^{(f(x)-f(y))/T}$;
end

- Method developed for generating sample states of a thermodynamic system (1953)
- T is **fixed** over the iterations

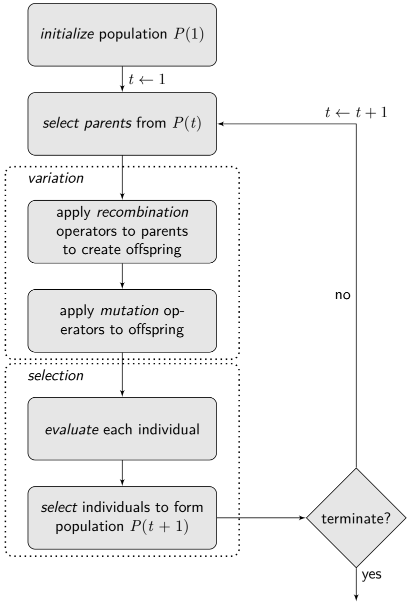
General-purpose algorithms for optimization

Simulated Annealing

Choose x uniformly at random from $\{0,1\}^n$;
while *stopping criterion not met* **do**
  $y \leftarrow x$, $t \leftarrow 0$;
  Choose i uniformly at random from $\{1,\ldots,n\}$;
  $y_i \leftarrow (1-y_i)$;
  if $f(y) \geq f(x)$ **then** $x \leftarrow y$;
  else $x \leftarrow y$ with probability $e^{(f(y)-f(x))/T_t}$;
  $t \leftarrow t+1$;
end

- Heating and controlled cooling of a material to increase crystal size and reduce their defects.
- High temperature $\Rightarrow$ many random state changes
- Low temperature $\Rightarrow$ system prefers "low energy" states (high fitness)
- Idea is to carefully settle the system down over time to its lowest energy state (highest fitness) by **cooling**
- T_t is **dependent on** t, typically decreasing.

Evolutionary Algorithms



Evolutionary Algorithms

- Allow larger jumps
- Long (destructive) jumps should be rare

(1+1) EA

Choose x uniformly at random from $\{0,1\}^n$;
while *stopping criterion not met* **do**
  $y \leftarrow x$;
  foreach $i \in \{1,\ldots,n\}$ **do**
    With probability $1/n$, $y_i \leftarrow (1-y_i)$;
  end
  if $f(y) \geq f(x)$ **then** $x \leftarrow y$;
end