Section 7
Evolutionary Algorithms

"If the only tool you have is a hammer,
it is tempting to treat everything as if it were a nail."

Abraham Maslow, 1966

Evolutionary Algorithms - Outline

Outline of this section:

- Exhaustive Search and Random Search
- Simulated Annealing
- Evolutionary Algorithms
  - **Evolution Strategies (ESs)**
  - **Genetic Algorithms (GAs)**
  - **Genetic Programming (GP)** (not covered here)
  - **Neuroevolution (NE)** (not covered here)

Exhaustive Search, Grid Search and Random Search

Finding the global optimum of a non-convex, non-linear, and non-differentiable function is an NP-hard problem.

If the domain is discrete, one could use the Exhaustive Search approach, which consists of evaluating the function on all the points of the search domain.

Unfortunately, this approach is impracticable for most of the real-world problems, due to its computational complexity, even in the Exascale era (or the upcoming Petascale one).

Grid Search is the equivalent of Exaustive Search when the meta-parameters are continous. In such case, all the meta-parameters are discretized between a minimum and a maximum value, then exaustive search is performed. Again, it scales very poorly with the number of meta-parameters (course of dimensionality).

Random Search is a stochastic search method which only requires a very good algorithm to generate random numbers, but, it is highly ineffective.
Stochastic search algorithms inspired by nature can be the solution.

(Physical) Annealing (Metallurgy)

- **(Physical) Annealing (materials science)** In metallurgy and materials science, annealing is a heat treatment that alters the physical and sometimes chemical properties of a material to increase its ductility and reduce its hardness, making it more workable. It involves heating a material above its recrystallization temperature, maintaining a suitable temperature for an appropriate amount of time and then cooling.
- In annealing, atoms migrate in the crystal lattice and the number of dislocations decreases, leading to a change in ductility and hardness. As the material cools it recrystallizes. For many alloys, including carbon steel, the crystal grain size and phase composition, which ultimately determine the material properties, are dependent on the heating rate and cooling rate. Hot working or cold working after the annealing process alters the metal structure, so further heat treatments may be used to achieve the properties required. With knowledge of the composition and phase diagram, heat treatment can be used to adjust from harder and more brittle to softer and more ductile.
- In the case of ferrous metals, such as steel, annealing is performed by heating the material (generally until glowing) for a while and then slowly letting it cool to room temperature in still air. Copper, silver and brass can be either cooled slowly in air, or quickly by quenching in water. In this fashion, the metal is softened and prepared for further work such as shaping, stamping, or forming.

Difference between Annealing and Tempering in Metallurgy (1)

Annealing and tempering are both heat treatment processes used in metallurgy to alter the properties of metals. However, they have different purposes and achieve different results.

Annealing is a process of heating a metal to a high temperature and then cooling it slowly. This process is used to:

- Soften the metal
- Improve ductility
- Relieve internal stresses
- Restore the original properties of a cold-worked metal

Tempering is a process of heating a hardened metal to a temperature below its critical temperature and then cooling it. This process is used to:

- Reduce the brittleness of a hardened metal
- Improve toughness.
- Achieve a balance between hardness and toughness.

The main difference between annealing and tempering is the temperature at which the metal is heated. Annealing is done at a higher temperature than tempering. Additionally, annealing is typically followed by slow cooling, while tempering can be followed by air cooling, oil quenching, or water quenching.

Difference between Annealing and Tempering in Metallurgy (2)

Here is a table that summarizes the key differences between annealing and tempering:

Characteristic	Annealing	Tempering
Purpose	Soften metal, improve ductility, relieve internal stresses, restore original properties	Reduce brittleness of hardened metal, improve toughness, achieve balance between hardness and toughness
Heating temperature	Higher	Lower
Cooling rate	Slow	Air cooling, oil quenching, or water quenching
Effect on microstructure	Coarse-grained	Fine-grained
Effect on properties	Soft, ductile, tough, machinable	Hard, tough
Applications	Softening a metal for further processing, relieving internal stresses in a metal, improving the ductility and toughness of a metal, restoring the original properties of a cold-worked metal	Reducing the brittleness of a hardened metal, improving the toughness of a hardened metal, achieving a balance between hardness and toughness in a metal

Simulated Annealing

Simulated annealing is a probabilistic optimization algorithm inspired by the annealing process in metallurgy. It is used to find near-optimal solutions to combinatorial optimization problems, particularly in cases where traditional gradient-based methods may get stuck in local optima.

The algorithm is named after the annealing process, which involves heating and slowly cooling a material to reduce its defects and reach a more stable state. Simulated annealing mimics this process by iteratively exploring the solution space, allowing for occasional uphill moves that may lead to better solutions.

The basic idea of simulated annealing is to start with an initial solution and iteratively move to neighboring solutions, accepting both improvements and occasionally accepting worse solutions. The acceptance of worse solutions is controlled by a temperature parameter, which is gradually decreased over time.

At higher temperatures, the algorithm has a higher probability of accepting worse solutions, allowing for exploration of the solution space and avoiding getting trapped in local optima. As the temperature decreases, the algorithm becomes more selective and tends to converge towards the optimal or near-optimal solution.

Simulated Annealing (2)

Here's a high-level overview of the simulated annealing algorithm:

Start with an initial solution.
Initialize the temperature and cooling schedule parameters.
Iterate until the stopping criterion is met:

- **a.** Generate a neighboring solution.
- **b.** Evaluate the objective function for the new solution.
- **c.** Compare the objective function values of the current and new solutions.
- **d.** If the new solution is better, accept it as the current solution.
- **e.** If the new solution is worse, calculate the acceptance probability based on the temperature and a criterion such as the Metropolis criterion. Accept the worse solution with a certain probability.
- **f.** Adjust the temperature according to the cooling schedule.

Return the best solution found during the iterations.

Simulated Annealing (3)

- The simulated cooling schedule determines how the temperature decreases over time. It controls the balance between exploration (at higher temperatures) and exploitation (at lower temperatures).
- Common cooling schedules include linear cooling, logarithmic cooling, and exponential cooling.
- Simulated annealing is particularly useful when dealing with complex optimization problems where the search space is large and rugged, and where finding the global optimum is challenging.
- It is applicable to a wide range of problems, including traveling salesman problems, scheduling problems, and parameter optimization in machine learning algorithms.
- It has been proved that asymptotically, the SA is able to find the global optimum:

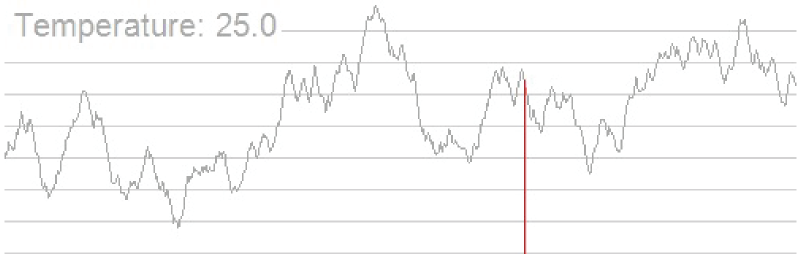
Granville, V. & Krivanek, M. & Rasson, J.-P. (1994). "Simulated annealing: A proof of convergence". IEEE Transactions on Pattern Analysis and Machine Intelligence. 16 (6): 652–656.

Simulated Annealing (4) - Pseudocode

```
function [best_x, best_y] = SA_RE_1D(f,a,b,niter)
% Simulated Annealing for the minimisation of a given function f
% over the one dimensional interval (a,b).
% A Real Encoding of the input variable is assumed.

time = 1:niter;
temperature = exp(-lambda*time); % generate the temperature profile

x1 = generate_initial_random_solution_within_a_and_b(a,b);
y1 = evaluate_objective_function_on_x1(f,x1);

current_x = x1; % set the current solution
current_y = y1; % set the valute for the function of the current solution

best_x = x1;    % initialise the best solution found so far
best_y = y1;    % initialise the best objective value found so far

for i=2:niter
    new_x = perturb(current_x,a,b);
    new_y = feval(f,new_x);
    if (new_y < current_y) || ( rand() < temperature(i) )
        current_x = new_x;
        current_y = new_y;
    end
    if new_y < best_y
        best_x = new_x;
        best_y = new_y;
    end
end
```

Simulated Annealing (5) - An animation

The wikipedia page dedicated to SA has a nice animation about the sequence of points generated by SA when looking for the maximum of a one-dimensional function:
https://en.wikipedia.org/wiki/Simulated_annealing



Test functions for optimisation

The link below contains a few test functions for optimisation:
https://en.wikipedia.org/wiki/Test_functions_for_optimization