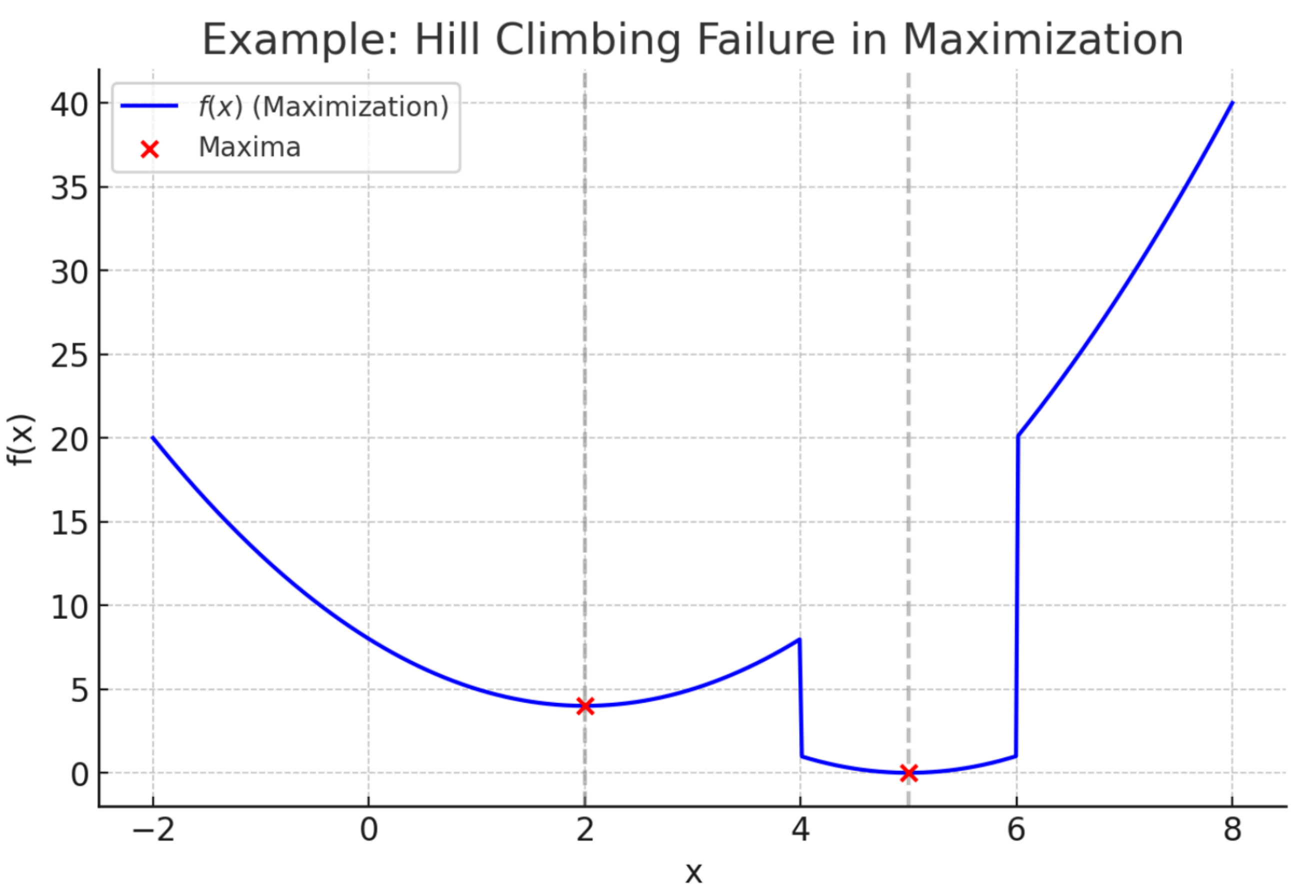
Zeroth order optimization techniques (a subset of gradient-free optimization techniques) are often used when the search space is discrete (look at the Traveling Salesman Problem (TSP), the Boolean Satisfiability Problem (SAT) and the Job-Scheduling Problem). It is preferable over exact algorithms like gradient descent or branch and bound in scenarios where an approximate solution is sufficient.

Why use Zeroth Order Optimization Techniques if Gradient-Based Techniques give us exact optima?

Gradient descent fails in the following scenarios:

- non-differentiable functions
- highly convex landscapes with many local minima
- flat regions and vanishing gradients
- noisy or stochastic gradients
- high-dimensional discrete optimization
- adversarial or non-smooth loss surfaces

Hill Climbing fails when a local optimum is reached.



In these cases, we can leverage:

- Zeroth Order Optimization (ZOO) Algorithms
    - Simulated Annealing
    - Evolutionary Algorithms
        - Genetic Algorithms
    - Random Search
    - Bayesian Optimization
- Finite-Difference-Based Methods
    - Finite Gradient Approximation (gradients are estimated via small perturbations)
- Direct Search Methods
    - Nelder-Mead (Simplex) Algorithm
    - Pattern Search
- Heuristic and Metaheuristic Methods
    - Ant Colony Optimization (ACO)
    - Particle Swarm Optimization (PSO)

Simulated Annealing

Simulated annealing - Wikipedia

Simulated Annealing (SA) is a zeroth order optimization algorithm and a probabilistic technique for approximating the global optimum of a given function. For large numbers of local optima, SA can find the global optimum. The name of the algorithm comes from annealing in metallurgy, a technique involving heating and controlled cooling of a material to alter its physical properties.

SA is a candidate-based optimization where a possible solution $w \in \Omega$ picked at random and an objective function E (often the Gibbs Free Energy) is minimized. In each iteration, the algorithm probabilistically chooses to move to a neighboring solution w'. These probabilities ultimately lead the system to move to states of lower energy. This step is repeated until the systems reaches a good enough solution or the computation budget has been exhausted.

Algorithm

Algorithm Parameters: Objective function $E : \Omega \to \mathbb{R}$, Cooling Schedule $\mathcal{T}$, Neighboring Candidate Generator $\mathcal{N}$, ϵ
Input: Parameter Space Ω, Starting Temperature t, Stopping Temperature t_0

Algorithm Simulated Annealing

$w \leftarrow \text{randomSample}(\Omega)$
$E(w) \leftarrow \infty$
while $t \geq t_0$ **do**
  $w' \leftarrow \mathcal{N}(w, rand(0,1))$
  if $||E(w) - E(w')|| < \epsilon$ **then**
    $w \leftarrow w'$

else
$w \leftarrow w'$ with probability $e^{-(E(w')-E(w))/t}$
end if
$t = \mathcal{T}(t)$
end while

Factors: Random Sampling, Neighborhood Domain, Cooling Schedule

$\mathcal{T}(t) = \alpha t,\ \alpha \in (0,1),\ \alpha \in \mathbb{R}$ is a widely used cooling function. A cooling schedule must follow the stochastic approximation conditions to ensure convergence

$$\Sigma_n t_n = \infty$$
$$\Sigma_n t_n^2 < \infty$$

Example

Traveling Salesman Problem (TSP) solved using Simulated Annealing (SA).