Zero-Order Methods

Derivative-Free Search, Nature-Inspired Methods

Prof. Karl Ezra Pilario, Ph.D.

Process Systems Engineering Laboratory
Department of Chemical Engineering
University of the Philippines Diliman

Outline

- Introduction to NLP
- Necessary and Sufficient Conditions for Optimality
- Convex Programming
- Methods for Solving NLP
  - One-Dimensional, Unconstrained NLP
  - Multivariable, Unconstrained NLP
  - **Zero-order**, First-order, Second-order Methods
  - Constrained NLP

A Taxonomy of NLP Solvers

Zero-order Methods

Derivative-free, black-box

- Grid Search / Exhaustive Search
- Random Search
- Nelder-Mead Simplex

- Metaheuristic Search
  - Genetic Algorithms
  - Particle Swarm
  - Simulated Annealing
  - Differential Evolution
  - CMAES

- Bayesian Optimization / Surrogate-based

First-order Methods

- **Zero-order methods** are good for objective functions whose
  - exact expression is unknown, or
  - it is known but hard / impossible to differentiate.

- If we know the exact expression and it is differentiable, then it is better to use first-order / second-order methods.

- **Scenarios where zero-order methods are useful:**
  - Design of Experiments → Self-driving labs!
  - Fast prototyping of a product / material design → Materials Discovery!
  - Optimization of machine learning hyper-parameters or architectures
  - Surrogate optimization in chemical plants

Second-order Methods

ve information

d (BFGS)

t

Methods

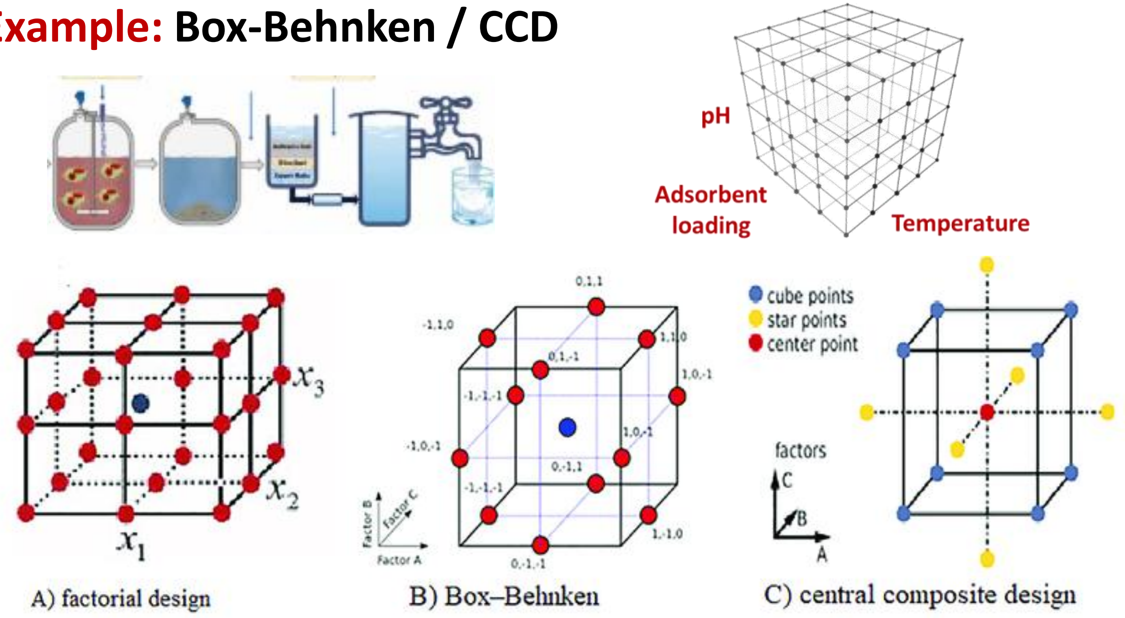
r-Point

Grid Search and Random Search

Grid Search

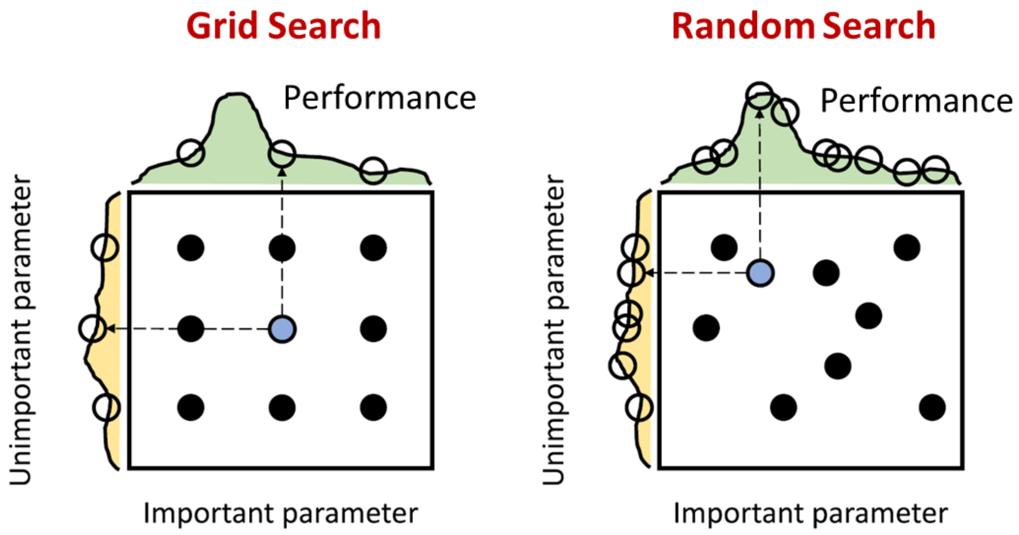
- Arrange all combinations as a grid (space-filling).
- Exhaustively try all combinations, within your budget.
- **Disadvantages:**
  - Most of the trials are not promising.
  - A low budget forces a low-resolution search.
  - None of the trials may be optimal. → Need RSM

Example: Box-Behnken / CCD



Random Search

- Try N *random* candidates, within a budget.
- Has a higher chance of finding optimality than grid search given that not all factors are important.
- **Disadvantages:**
  - Not exhaustive.
  - Still not guided by prior knowledge.



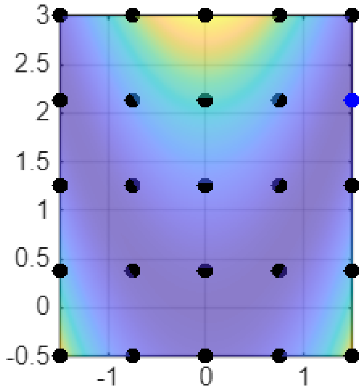
Grid Search and Random Search

Example

Find the minimum of the **Rosenbrock function** using only a budget of <u>25 trials</u>. Compare Grid search vs. Random search.

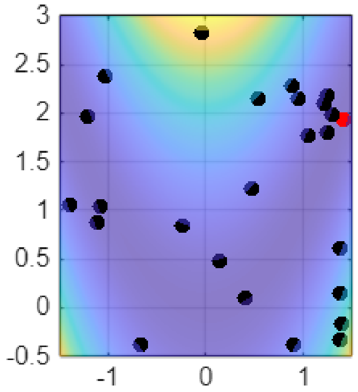
Note: In this example, we assume that the exact equation of the objective function is **unknown**, and the only way to know more about it is to *sample* it every trial.

Grid search



Minimum found: **1.81** at (1.50, 2.12)

Random search



Minimum found: **0.54** at (1.41, 1.93)

12 trials
Random search

24 trials
Grid search

A Taxonomy of NLP Solvers

Zero-order Methods

Derivative-free, black-box

- Grid Search / Exhaustive Search
- Random Search
- Nelder-Mead Simplex
- Metaheuristic Search
  - Genetic Algorithms
  - Particle Swarm
  - Simulated Annealing
  - Differential Evolution
  - CMAES
- Bayesian Optimization / Surrogate-based

First-order Methods

Uses 1st derivative information

- Steepest Descent or Gradient Descent
- Stochastic Gradient Descent (SGD)
  - RMSProp, Adam, etc.
- Conjugate Gradient methods (e.g. Fletcher-Reeves)
- Coordinate Descent

Second-order Methods

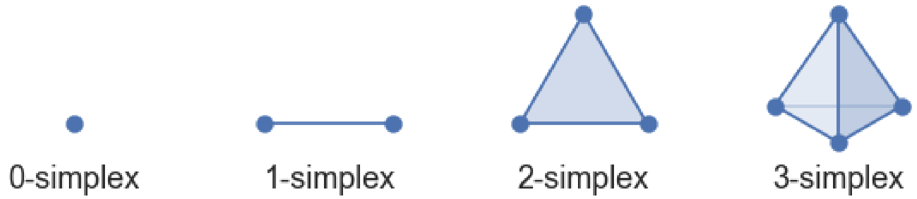
Uses 1st and 2nd derivative information

- *Newton's method*
- *Quasi-newton method (BFGS)*
- Levenberg-Marquardt
- Trust Region Newton Methods
- Active Set and Interior-Point Methods
- Sequential Quadratic Programming (SQP)
- IPOPT

We realize that we need to be **smart** about *where to sample next*, given the information from the past trials.

Nelder-Mead Simplex

- John Nelder and Roger Mead (1965)
- Uses a **simplex** (a polytope of $n + 1$ vertices in n dimensions) that moves roughly like gradient descent.



- The simplex has 4 possible moves every iteration:

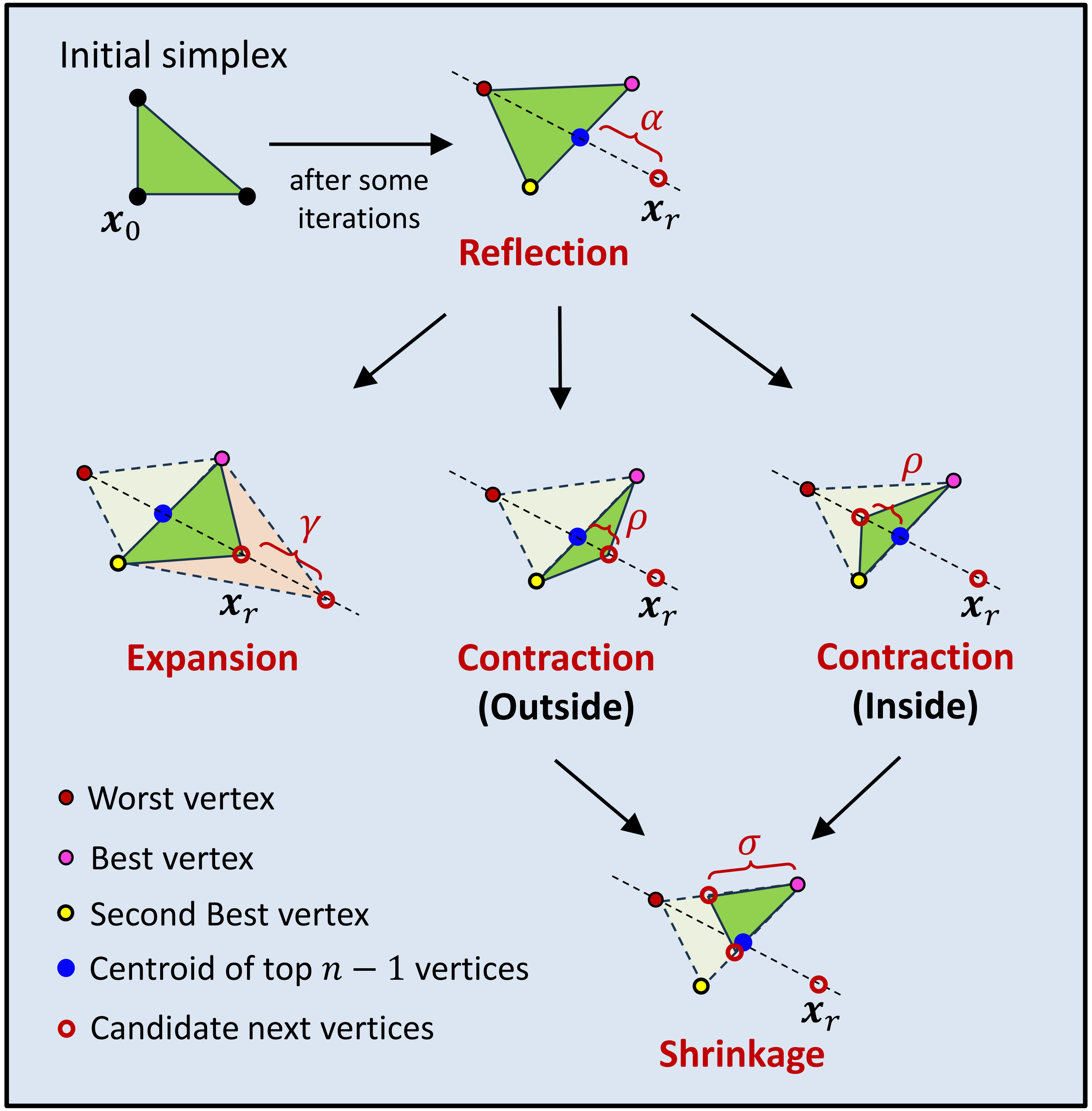
Reflection: $\boldsymbol{x}_r = \boldsymbol{x}_{cent} + \alpha(\boldsymbol{x}_{cent} - \boldsymbol{x}_{worst})$

Expansion: $\boldsymbol{x}_e = \boldsymbol{x}_{cent} + \gamma(\boldsymbol{x}_r - \boldsymbol{x}_{cent})$

Contraction: $\boldsymbol{x}_c = \boldsymbol{x}_{cent} + \rho(\boldsymbol{x}_r - \boldsymbol{x}_{cent})$

$\boldsymbol{x}_c = \boldsymbol{x}_{cent} + \rho(\boldsymbol{x}_{worst} - \boldsymbol{x}_{cent})$

Shrinkage: $\boldsymbol{x}_i = \boldsymbol{x}_{best} + \sigma(\boldsymbol{x}_i - \boldsymbol{x}_{best})$



https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

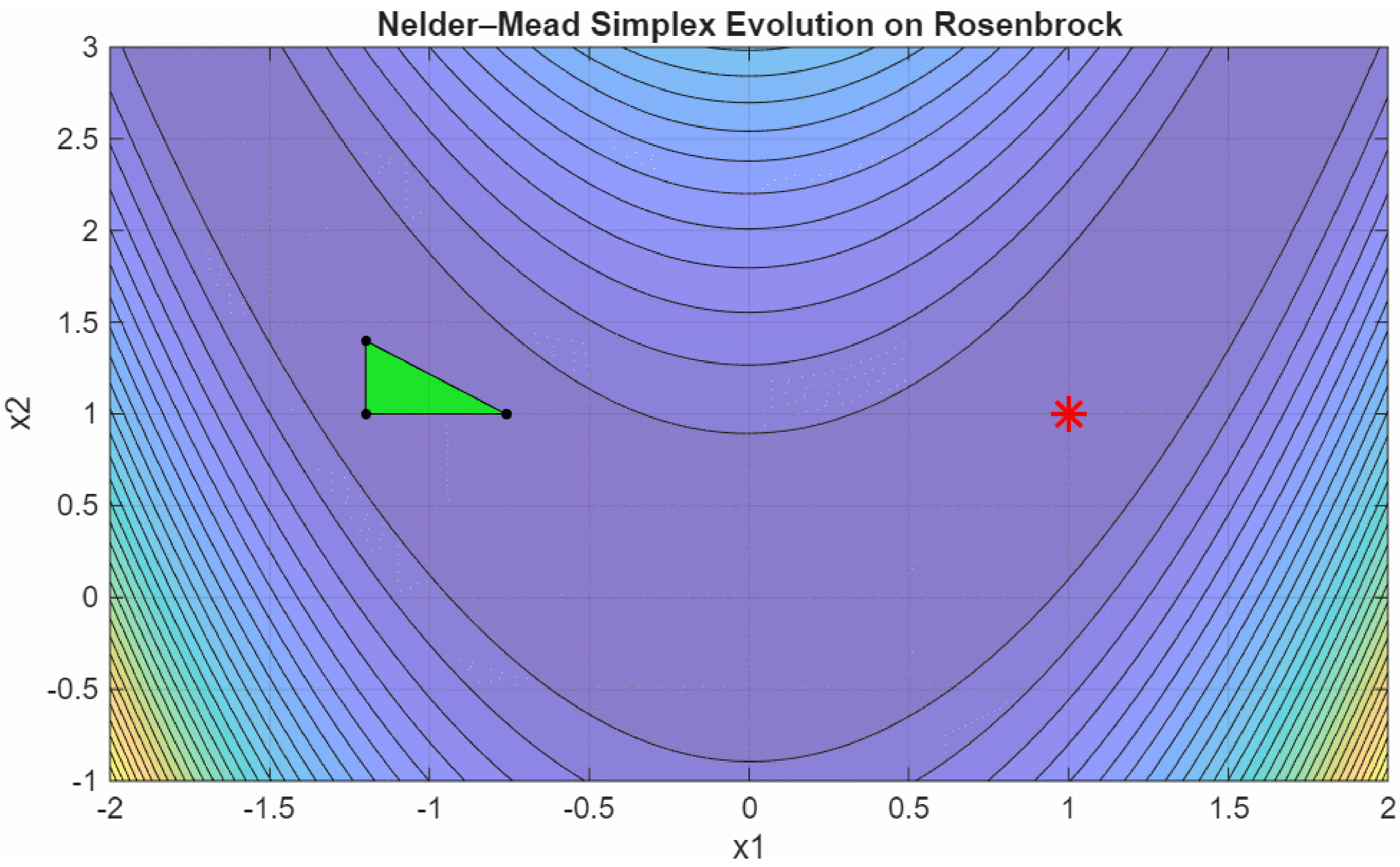
Nelder-Mead Simplex

Example

Find the minimum of the **Rosenbrock function** using Nelder-Mead simplex, with the following settings:

- Initial guess: $x_0 = [-1.2, 1.0]$
- Max iterations: 500
- Max Obj. Func. Evaluations: 2000
- Function tolerance: 1e-6
- Solution tolerance: 1e-8
- $\alpha = 1.0, \gamma = 2.0, \rho = 0.5, \sigma = 0.5$
- Initial scale: 0.05

```
Answer: x = [1.00026, 1.00052], f = 7.21993e-08
```



A Taxonomy of NLP Solvers

Zero-order Methods

Derivative-free, black-box

- Grid Search / Exhaustive Search
- Random Search
- Nelder-Mead Simplex

- Metaheuristic Search
  - Genetic Algorithms
  - Particle Swarm
  - Simulated Annealing
  - Differential Evolution
  - CMAES

- Bayesian Optimization / Surrogate-based

First-order Methods

Uses 1st derivative information

- Steepest Descent or Gradient Descent
- Stochastic Gradient Descent (SGD)
  - RMSProp, Adam, etc.
- Conjugate Gradient methods (e.g. Fletcher-Reeves)

Some optimization algorithms took inspiration from nature. They claim to be *global* optimizers.

Second-order Methods

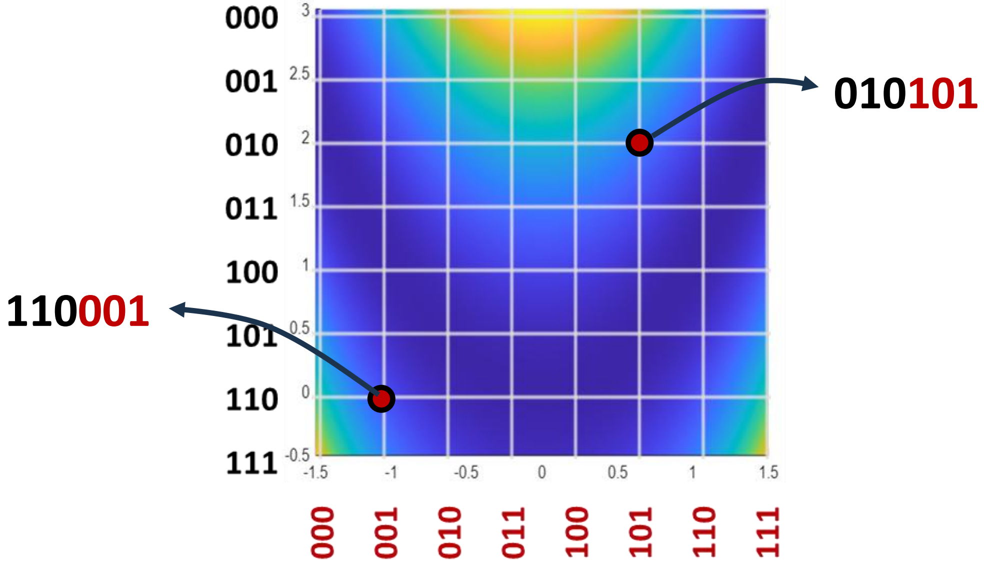
Uses 1st and 2nd derivative information

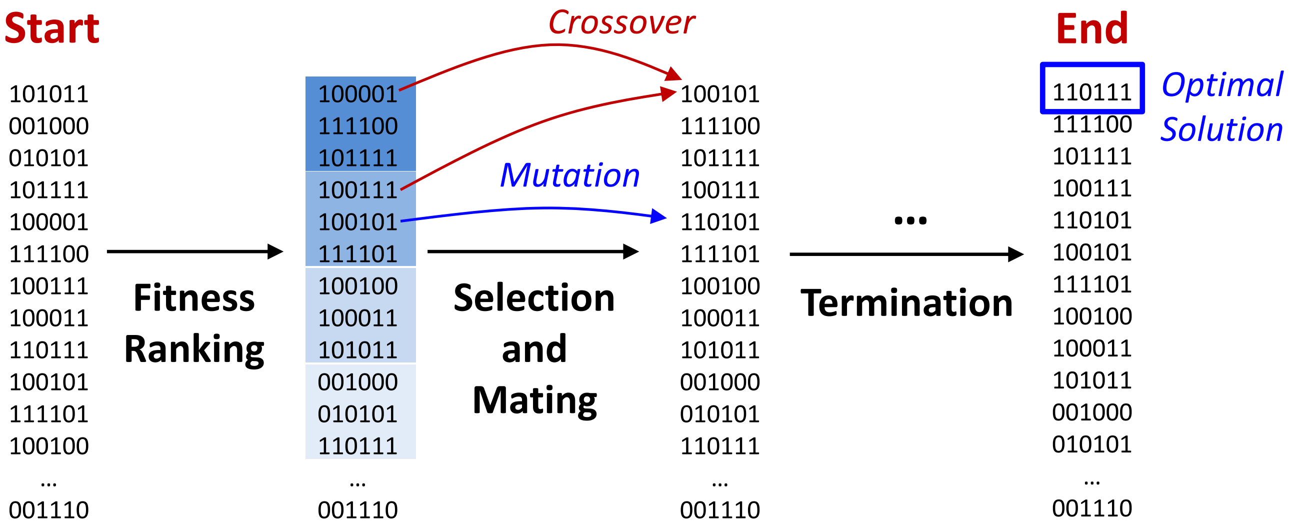
- *Newton's method*
- *Quasi-newton method (BFGS)*

- Levenberg-Marquardt
- Trust Region Newton Methods
- Active Set and Interior-Point Methods
- Sequential Quadratic Programming (SQP)
- IPOPT

Genetic Algorithm

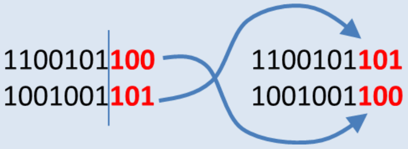
- John Holland, *Adaptation in Natural and Artificial Systems* (1975)
- Inspired by organisms "optimized" through years of evolution via **survival of the fittest**.
- Represent solutions as **chromosomes** (e.g. bit strings).
- In every generation, chromosomes are allowed to mate (**crossover**) and mutate (**mutation**) until convergence.





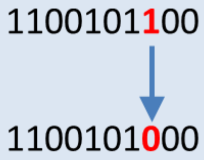
Crossover

- Promotes **exploitation.**
- Breed parents.
- The higher the fitness, the higher chance of being selected as parent.



Mutation

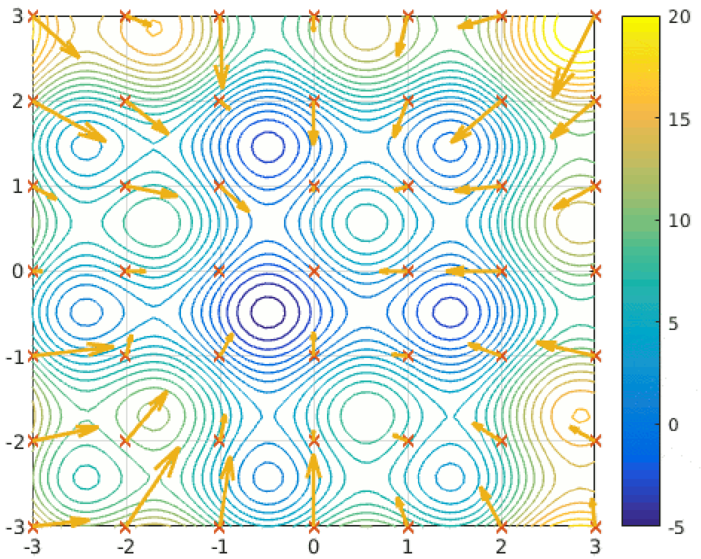
- Promotes **exploration.**
- Select a chromosome, then randomly change 1 or some bits.



Particle Swarm Optimization

- James Kennedy, Russell Eberhart, Yuhui Shi (1995)
- Inspired by **swarming of bees** and **flocks of birds** that can navigate together by sharing nearby information.
- Update solutions by computing their velocities, $\mathbf{V}_\mathrm{p}^i$:

$$\mathbf{x}_\mathrm{p}^i = \mathbf{x}_\mathrm{p}^{i-1} + \mathbf{V}_\mathrm{p}^i$$



$$\boxed{\mathbf{V}_\mathrm{p}^i = \mathrm{c}_0 \mathbf{V}_\mathrm{p}^{i-1} + \mathrm{c}_1 \times \mathrm{rand} \times \left(\mathbf{x}_p^{\mathrm{best}} - \mathbf{x}_\mathrm{p}^{i-1}\right) + \mathrm{c}_2 \times \mathrm{rand} \times \left(\mathbf{x}_\mathrm{g}^{\mathrm{best}} - \mathbf{x}_\mathrm{p}^{i-1}\right)}$$

c_0 **Inertia weight**

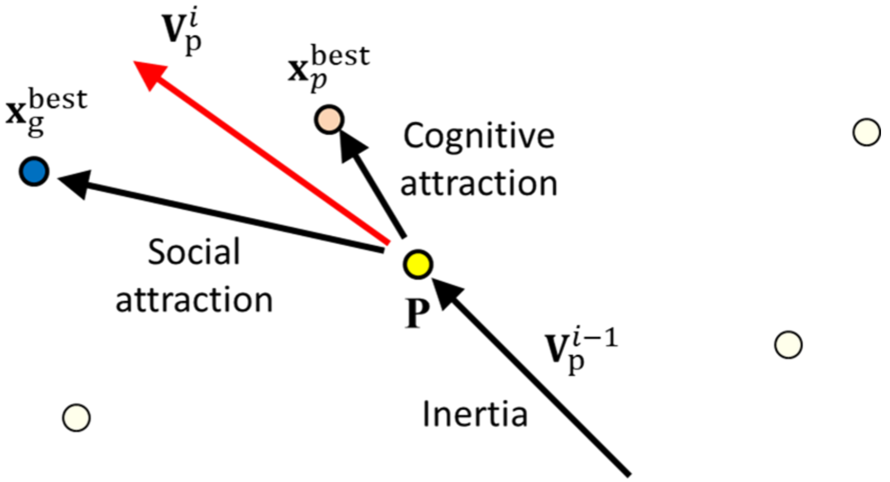
c_1 **Cognitive parameter**

c_2 **Social parameter**

$\mathbf{V}_\mathrm{p}^i$ Velocity for the current iteration (i)

$\mathbf{V}_\mathrm{p}^{i-1}$ Velocity from the previous iteration $(i-1)$

$\mathbf{x}_\mathrm{p}^{i-1}$ Previous position of the particle $(i-1)$



$\mathbf{x}_p^{\mathrm{best}}$ Best position the individual particle p encountered so far

$\mathbf{x}_\mathrm{g}^{\mathrm{best}}$ Global best position encountered by whole swarm so far

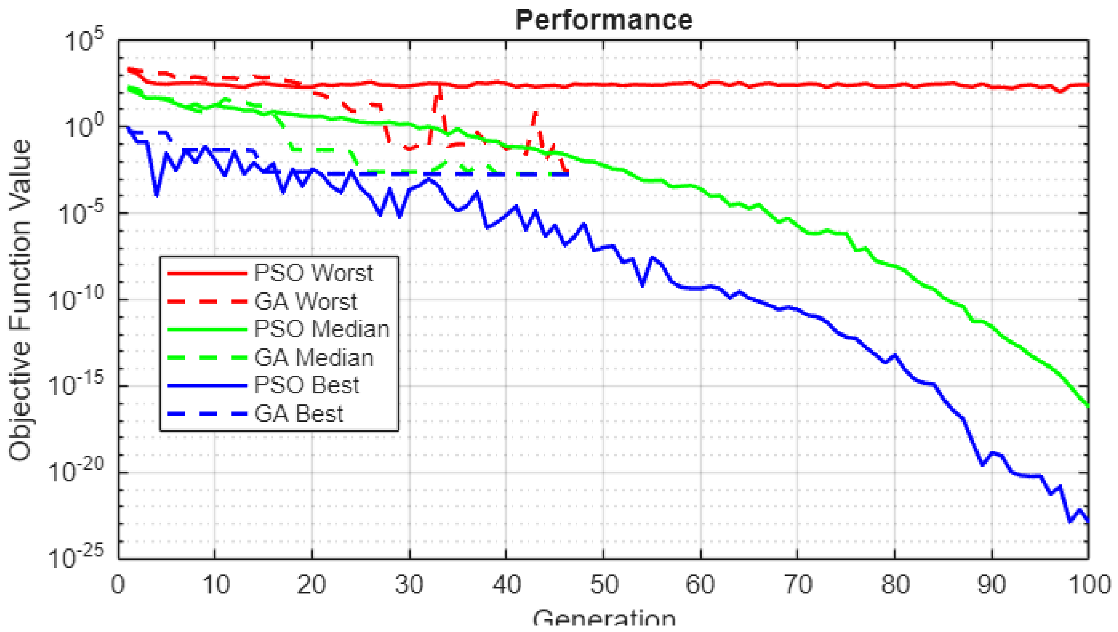
rand a random real number from 0 to 1 that′s different every time.

GA and PSO

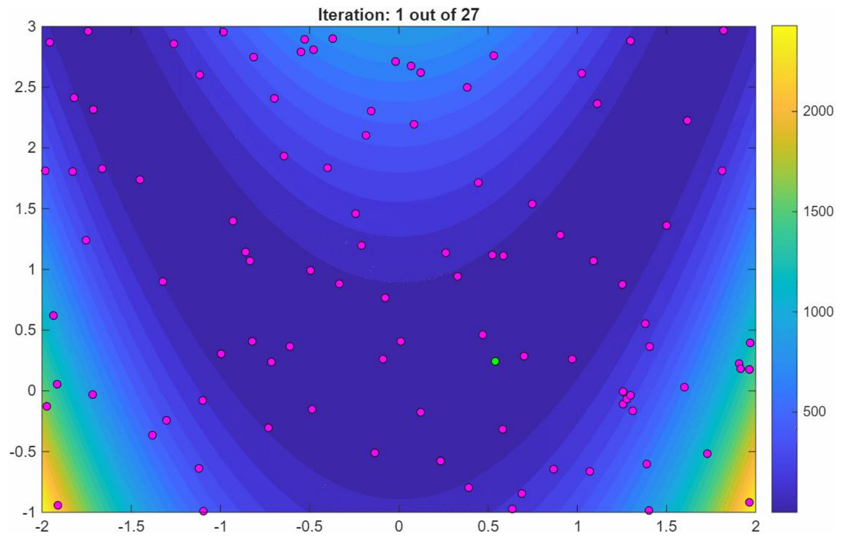
Example

Find the minimum of the **Rosenbrock function** using GA and PSO with the following settings:

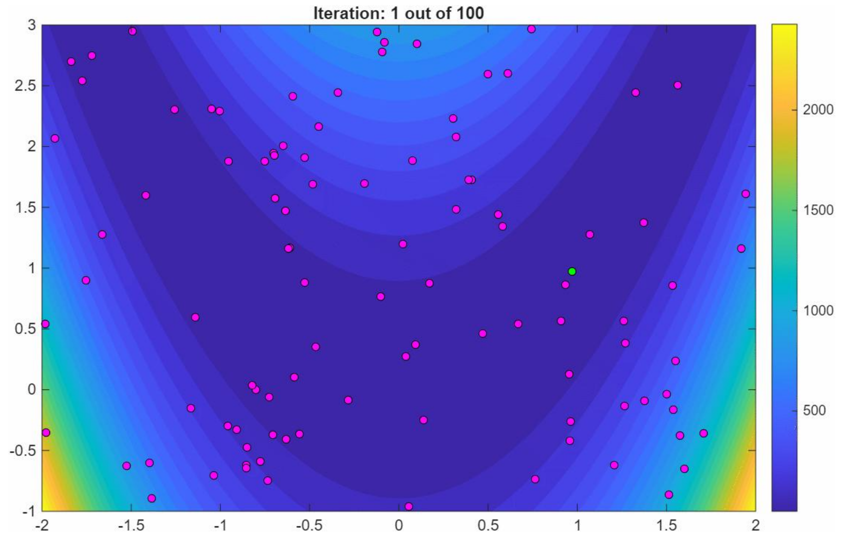
- Population size: 100 (both)
- No. of iterations: 100 (both)
- GA: Crossover rate = 1.0, Mutation rate = 0.01
- PSO: Cognitive = 2.0, Social = 2.0, inertia = 0.9 to 0.1



Genetic Algorithm

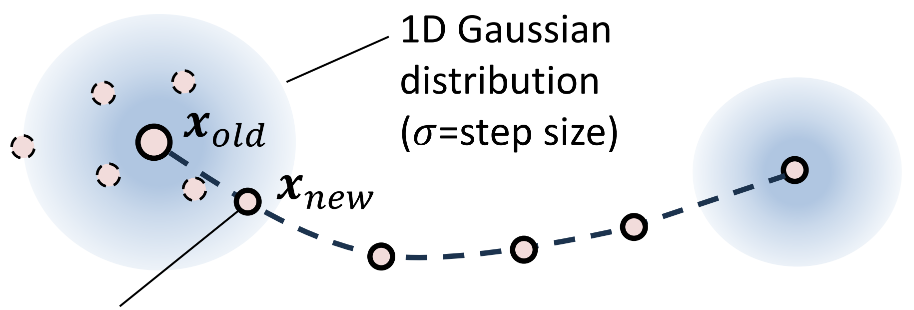


Particle Swarm Optimization



Simulated Annealing

- Kirkpatrick, Gelatt Jr., and Vecchi (1983)
- Inspired by **annealing** in metallurgy: controlled cooling of a material to alter its physical properties.
- Choose a *random next guess* (Gaussian distributed) then accept it by **some probability**.



Accept only if:

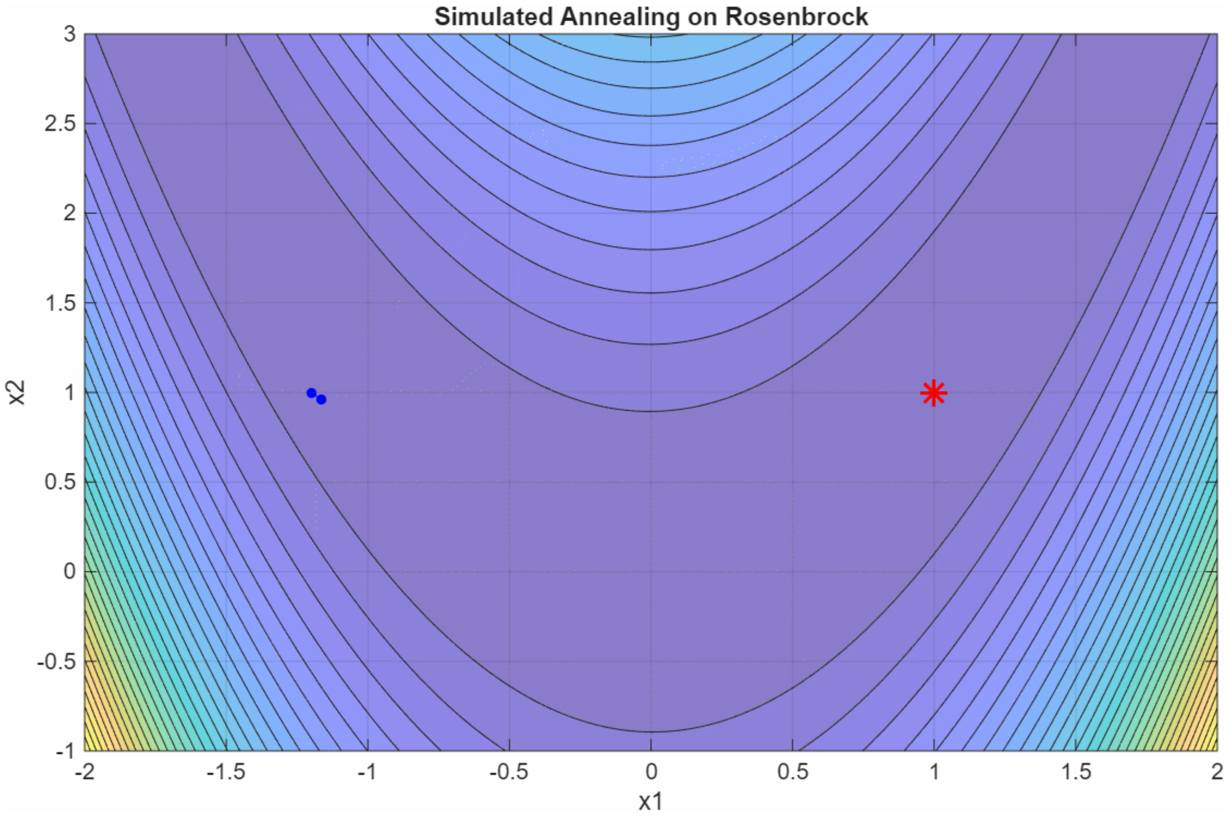
- x_{new} is better, $f(x_{new}) < f(x_{old})$
- or a random number in [0, 1] satisfies: $\text{rand} < \exp\left(\dfrac{f(x_{new}) - f(x_{old})}{T}\right)$

T = temperature (decreases every iteration)

Example

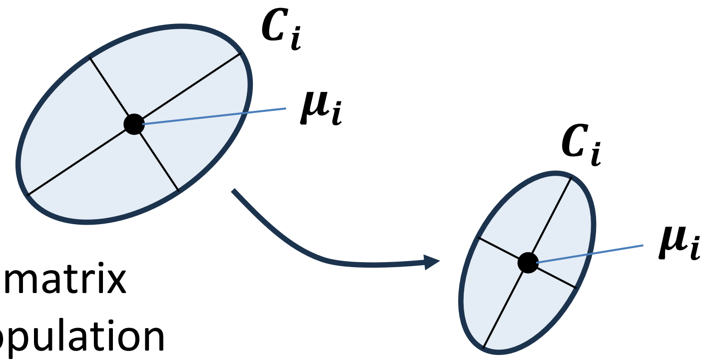
Find the minimum of the **Rosenbrock function** using Simulated Annealing, with the following settings:

- Initial guess: $x_0 = [-1.2, 1.0]$
- Max iterations: 5000
- Initial Temperature: 1.0
- Step size: 0.2



CMAES

- **C**ovariance **M**atrix **A**daptation - **E**volution **S**trategy
- Hansen et al. (1995, 1996, 1997, 2001)
- Inspired also by **evolution**, but instead of chromosomes, a population of candidates from a **multivariate Gaussian distribution** is being evolved.



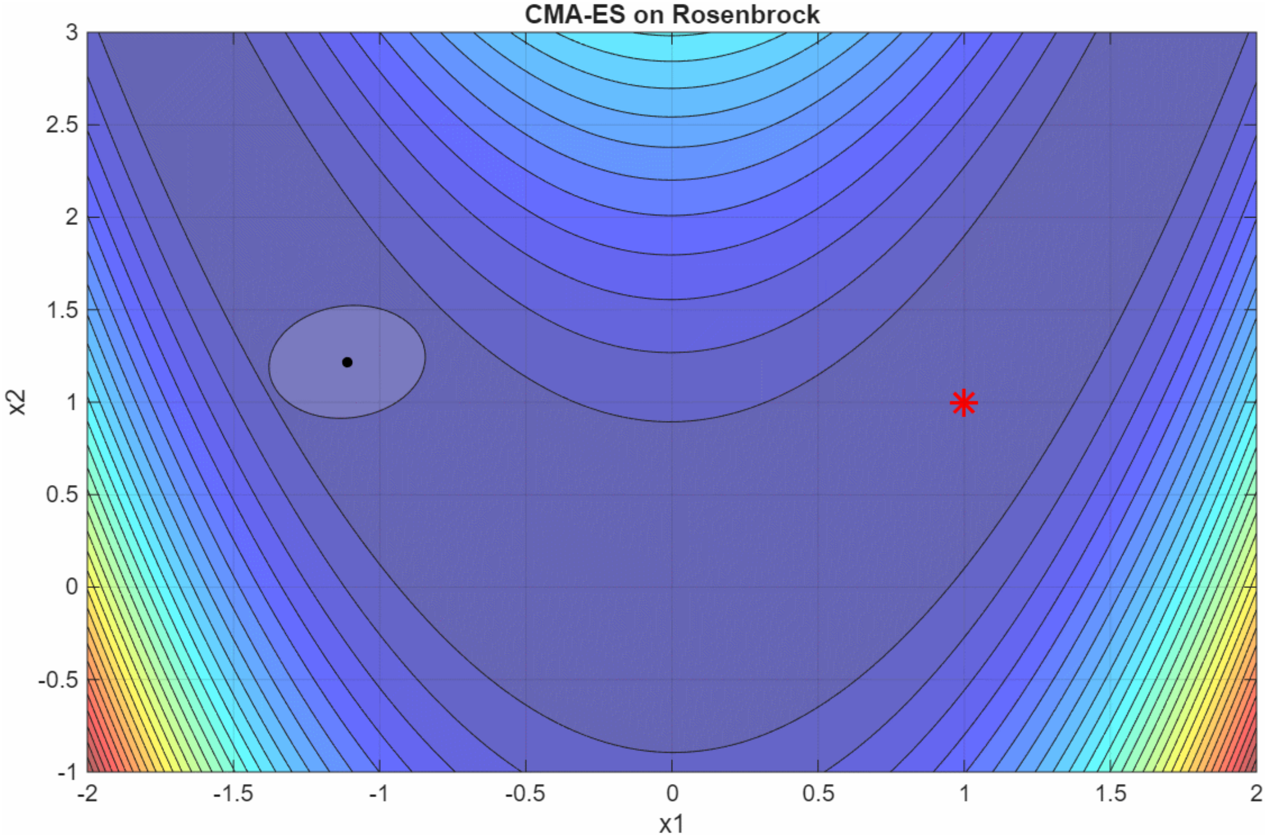
$\boldsymbol{C}_i$ = covariance matrix
$\boldsymbol{\mu}_i$ = mean of population

$$\mathcal{N}(\boldsymbol{X}|\boldsymbol{\mu},\boldsymbol{C}) = \frac{1}{(2\pi)^{D/2}|\boldsymbol{C}|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu})^T\boldsymbol{C}^{-1}(\boldsymbol{X}-\boldsymbol{\mu})\right)$$

Example

Find the minimum of the **Rosenbrock function** using Simulated Annealing, with the following settings:

- Initial guess: $x_0 = [-1.2, 1.0]$
- Max iterations: 5000
- Initial Temperature: 1.0
- Step size: 0.2



Nature-Inspired Algorithms

By August 2020, there are already more than 200 different nature-inspired optimization algorithms!

Source: Tzanetos et al. (Aug 2020). A comprehensive database of Nature-Inspired Algorithms. Data in Brief. Elsevier. doi: 10.1016/j.dib.2020.105792

Ant Colony Optimization
Bacterial Foraging Optimization
BeeHive
Bee Colony Optimization
The Bees Algorithm
Artificial Bee Colony
Bee Swarm Optimization
Hunting Search
Bat Algorithm
Cockroach Swarm Optimization
Eurygaster Algorithm
Blind naked Mole-rat Algorithm
OptBees
Wolf Search
African Wild Dog Algorithm
Seven-spot Ladybird Optimization
Penguins Search Optimization Algorithm
Grey Wolf Optimizer
Chicken Swarm Optimization
Cockroach Swarm Evolution
Butterfly Optimization Algorithm
Whale Optimization Algorithm
Bird Swarm Algorithm
Dolphin Swarm Optimization Algorithm
Raven Roosting Optimisation Algorithm
Spotted Hyena Optimizer
Whale Swarm Algorithm
Meerkat Clan Algorithm
Yellow Saddle Goatfish
Sailfish Optimizer
Manta Ray Foraging Optimization
Tunicate Swarm Algorithm
Fish-swarm Algorithm
Glowworm Swarm-based Optimization
Roach Infestation Optimization
Bumblebees
Cuckoo Search
Dolphin Partner Optimization
Firefly Algorithm
Locust Swarms

Termite Colony Optimization
Krill Herd
Social Spider Optimization
Swallow Swarm Optimization Algorithm
Spider Monkey Optimization Algorithm
Wolf Pack Algorithm
Earthworm Optimization Algorithm
African Buffalo Optimization
Elephant Herding Optimization
Elephant Search Algorithm
See-See Partridge Chicks Optimization
Dolphin Swarm Algorithm
Lion Optimization Algorithm
Emperor Penguin Optimizer
Coyote Optimization Algorithm
Anglerfish Algorithm
Dominion Algorithm
Emperor Penguins Colony
Blue Monkey
Black Widow Optimization
Black Widow Optimization Algorithm
Stochastic Diffusion Search
Particle Swarm Optimization
Sheep Flocks Heredity Model Algorithm
Bacterial Swarming Algorithm
Jumping Frogs Optimization
Fish School Search
Artificial Searching Swarm Algorithm
Hierarchical Swarm Model
Bumble Bees Mating Optimization
Migrating Birds Optimization
Weightless Swarm Algorithm
Pigeon-inspired Optimization
Animal Migration Optimization
Moth Flame Optimization
Cockroach Colony Optimization
Monarch Butterfly Optimization
Dragonfly Algorithm
Grasshopper Optimization Algorithm
Salp Swarm Algorithm

Selfish Herd Optimizer
Bison
Biology Migration Algorithm
Lion Pride Optimization Algorithm
Wildebeests Herd Optimization
Artificial Social Cockroaches
Meerkat-inspired Algorithm
Seagull Optimization Algorithm
Vocalization of Humpback Whale Optimization Algorithm
Military Dog Optimizer
Fitness Dependent Optimizer
Collective Animal Behavior
Side-Blotched Lizard Algorithm
Marriage in Honey Bees
Bacterial Chemotaxis
Honey-bees Mating Optimization
Shuffle frog-leaping Algorithm
Cat Swarm Optimization
Group Search Optimizer
Monkey Search
Japanese tree frogs
Biogeography-based Optimization
Bee Collecting Pollen Algorithm
Viral Systems
Monkey Algorithm
Fruit Fly Optimization
Mosquitos Oviposition
Bat Sonar Algorithm
Hoopoe Heuristic Optimization
Lion Pride Optimizer
The Great Salmon Run
The Lion's Algorithm
Superbug Algorithm
Keshtel Algorithm
Swine Influenza Models Based Optimization
Dolphin Echolocation
Green Herons Optimization Algorithm
Egyptian Vulture Optimization Algorithm
Jumper Firefly Algorithm
Cuttlefish Algorithm

Mussels Wandering Optimization
Prey-Predator Algorithm
Magnetotactic Bacteria Optimization Algorithm
Symbiotic Organisms Search
Coral Reefs Optimization Algorithm
Bird Mating Optimizer
Competition over Resources
Ant Lion Optimizer
Artificial Algae Algorithm
Jaguar Algorithm
Sperm Whale Algorithm
Virus Colony Search
Shark Smell Optimization
Crow Search Algorithm
Camel Algorithm
Virulence Optimization Algorithm
Mosquito flying optimization
Red Deer Algorithm
Rhino Herd
Polar Bear Optimization Algorithm
Fish Electrolocation Optimization
Satin Bowerbird Optimizer
Artificial Butterfly Optimization
Beetle Antennae Search Algorithm
Killer Whale Algorithm
Laying Chicken Algorithm
Cyclical Parthenogenesis Algorithm
Cyclical Parthenogenesis Algorithm
Beetle Swarm Optimization Algorithm
Eagle Piercing Optimizer
Rhinoceros Search Algorithm
Dolphin Pod Optimization
Owl Search Algorithm
Pontogammarus Maeoticus Swarm Optimization
Barnacles Mating Optimizer
Mouth Brooding Fish algorithm
Pity Beetle Algorithm
Cheetah Based Optimization Algorithm
Circular Structures of Puffer Fish
Water-Tank Fish Algorithm
Neuronal Communication

Cheetah Chase Algorithm
Falcon Optimization Algorithm
Flying Squirrel Optimizer
Harris Hawks Optimization
Sooty Tern Optimization Algorithm
Donkey and Smuggler Optimization Algorithm
Artificial Feeding Birds
Squirrel Search Algorithm
Andean Condor Algorithm
Genetic Algorithm
Bald Eagle Search
Raccoon Optimization Algorithm
Sea Lion Optimization Algorithm
Electric Fish Optimization
Urban Pigeon-Inspired Optimiser
Xerus Optimization Algorithm
Artificial Ecosystem-based Optimization
Marine Predators Algorithm
Shuffled Shepherd Optimization Algorithm
Water Strider Algorithm
Photosynthetic Algorithm
Saplings Growing up Algorithm
Invasive Weed Optimization
Plant Growth Optimization
Paddy Field Algorithm
Plant Propagation Algorithm
Artificial Plant Optimization Algorithm
Ecology-inspired Optimization Algorithm
Flower Pollination Algorithm
Physarum Optimization
Root Mass Optimization
Forest Optimization Algorithm
Root Growth Algorithm
Seed-based Plant Propagation Algorithm
Tree Seed Algorithm
Runner-root Algorithm
Natural Forest Regeneration
Waterweeds Algorithm
Root Tree Optimization Algorithm
Artificial Tree Algorithm
Mushroom Reproduction Optimization
Invasive Tumor Growth Optimization
Allostatic Optimization
Artificial Infectious Disease Optimization

Chemotherapy Science Algorithm
Kidney-inspired Algorithm
Artificial Coronary Circulation System
Artificial Ecosystem-based Optimization
Space Gravitational Algorithm
Big Bang - Big Crunch
Gravitational Search Algorithm
Galaxy-based Search Algorithm
Gravitational Interactions Optimization
Black Hole
Space Gravity Optimization
Multi-verse Optimizer
Supernova Optimizer
Simulated Annealing
Intelligent Water Drop
River Formation Dynamics
Wind Driven Optimization
Water Cycle Algorithm
Atmosphere Clouds Model Optimization
Artificial Raindrop Algorithm
Hurricane-based Optimization
Lightning Search Algorithm
Water Wave Optimization
Artificial Showering Algorithm
Circular Water Waves
Vortex Search Algorithm
Water Evaporation Optimization
Crystal Energy Optimization Algorithm
Lightning Attachment Procedure Optimization
Rainfall Optimization Algorithm
Hydrological Cycle Algorithm
Nuclear Fission–Nuclear Fusion
Nuclear Reaction Optimization
Whirlpool Algorithm
Electrostatic Discharge Algorithm
Targeted Showering Optimization
Harmony Search
Hysteretic Optimization
Electromagnetism-like Optimization
Particle Collision Algorithm
Central Force Optimization
Zombie Survival Optimization
Stem Cells Optimization Algorithm
Heart algorithm

Artificial Physics Optimization
Charged System Search
Chemical-reaction-inspired Optimization
Grenade Explosion Method
Ray Optimization
Mine Blast Optimization
Spiral Dynamics Algorithm
Gases Brownian Motion Optimization
Colliding Bodies Optimization
Kinetic Gas Molecule Optimization
States of Matter
General Relativity Search Algorithm
Optics Inspired Optimization
Stochastic Fractal Search
Heat Transfer Search
Ions Motion Algorithm
Sine Cosine Algorithm
Attraction Force Optimization
Electromagnetic Field Optimization
Ideal Gas Molecular Movement Algorithm
Thermal Exchange Optimization
Vibrating Particles System Algorithm
Sonar Inspired Optimization
Spring Search Algorithm
Flow Regime Algorithm
Coulombs Franklins Algorithm
Vapour-liquid Equilibrium-based Algorithm
Atom Search Optimization
Henry Gas Solubility Optimization
Artificial Electric Field Algorithm
Evolutionary Centers Algorithm
Ohm's Law Optimization algorithm
Electron Radar Search Algorithm
Spherical Search Optimizer
Equilibrium Optimizer
Newton Metaheuristic Algorithm
Tree Growth Algorithm
Farmland Fertility Algorithm
Tree Based Optimization
BladderWorts Suction Algorithm
Physarum-energy Optimization Algorithm
Fertile Field Algorithm
Clonal Selection Algorithm
Reincarnation Algorithm

A Taxonomy of NLP Solvers

Zero-order Methods

Derivative-free, black-box

- Grid Search / Exhaustive Search
- Random Search
- Nelder-Mead Simplex

- Metaheuristic Search
  - Genetic Algorithms
  - Particle Swarm
  - Simulated Annealing
  - Differential Evolution
  - CMAES

- Bayesian Optimization / Surrogate-based

First-order Methods

Uses 1st derivative information

- Steepest Descent or Gradient Descent
- Stochastic Gradient Descent (SGD)
  - RMSProp, Adam, etc.
- Conjugate Gradient methods (e.g. Fletcher-Reeves)
- Coordinate Descent

All of the above takes too many iterations (samples of the objective) to give a good solution.

We need a **sample-efficient method**.

Second-order Methods

Uses 1st and 2nd derivative information

- *Newton's method*
- *Quasi-newton method (BFGS)*

- Levenberg-Marquardt
- Trust Region Newton Methods
- Active Set and Interior-Point Methods
- Sequential Quadratic Programming (SQP)

IPOPT