Numerical Optimization 11: Evolutionary Methods

Qiang Zhu

University of Nevada Las Vegas

May 20, 2020

Overview

Population Methods

Previous lecture discussed some methods require a group of points to colleHaving a large number of individuals distributed throughout the design space can help the algorithm avoid becoming stuck in a local minimum. Information at different points in the design space can be shared between individuals to globally optimize the objective function. Most population methods are stochastic in nature, and it is generally easy to parallelize the computation.

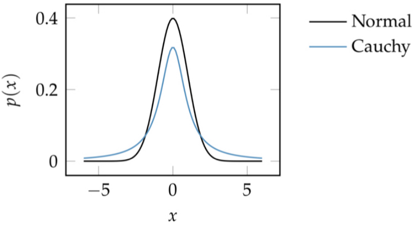
These methods typically have the following steps

- Initialization
- Encoding
- Mutation
- Crossover
- Selection

Initialization

Population methods begin with an initial population, just as descent methods require an initial design point. The initial population should be spread over the design space to increase the chances that the samples are close to the best regions. The following strategies can be applied

- Uniform distribution in a bounded region
- Multivariate normal distribution centered over a region of interest.
- The Cauchy distribution has an unbounded variance and can cover a much broader space.



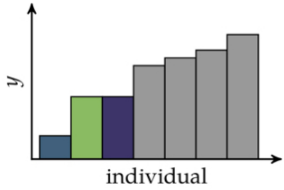
Chromosomes

There are several ways to represent chromosomes. The simplest is the binary string chromosome, a representation that is similar to the way DNA is encoded.

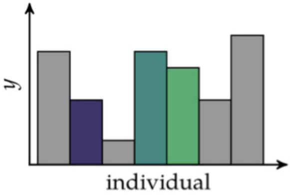
Selection

Selection is the process of choosing chromosomes to use as parents for the next generation. For a population with m chromosomes, a selection method will produce a list of m parental pairs for the m children of the next generation. The selected pairs may contain duplicates.

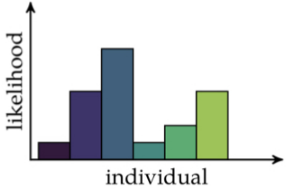
- Truncation, random one from the best k truncation
- Tournament, the fittest out of k randomly chosen
- Roulette wheel, chosen with a probability proportional to the fitness



Truncation



Tournament



Roulette wheel

Covariance Matrix Adaptation

Covariance matrix adaptation maintains a mean vector $\boldsymbol{\mu}$, a covariance matrix $\boldsymbol{\Sigma}$, and an additional step-size scalar δ. The covariance matrix only increases or decreases in a single direction with every iteration, whereas the step-size scalar is adapted to control the overall spread of the distribution. At every iteration, m designs are sampled from the multivariate Gaussian

$$\boldsymbol{x} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \Sigma)$$

The designs are then sorted according to their objective function values such that $f(x^1) \leq f(x^2) \leq \cdots \leq f(x^m)$. A new mean vector $\boldsymbol{\mu}^{k+1}$ is formed using a weighted average of the sampled designs:

$$\boldsymbol{\mu}^{k+1} \leftarrow \sum_{i=1}^{m} w_i \boldsymbol{x}^i$$

$$\sum_{i}^{m} w_i = 1 \qquad w_1 > w_2 > \cdots > w_m > 0$$

Particle Swarm Optimization

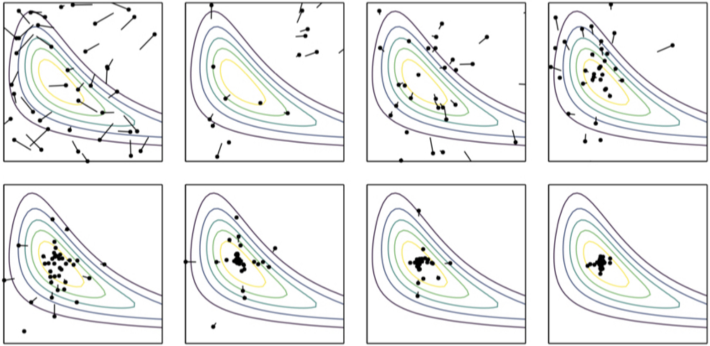
Particle swarm optimization introduces momentum to accelerate convergence toward minima. Each individual (particle), in the population keeps track of its current position, velocity, and the best position it has seen so far. Momentum allows an individual to accumulate speed in a favorable direction, independent of local perturbations.

$$\boldsymbol{x}^i \leftarrow \boldsymbol{x}^i + \boldsymbol{v}^i$$
$$\boldsymbol{v}^i \leftarrow w\boldsymbol{v}^i + c_1 r_1(\boldsymbol{x}^i_{lbest} - x^i) + c_2 r_2(\boldsymbol{x}_{gbest} - x^i)$$

where

- x_{lbest}: the current local best locations for the given population
- x_{gbest}: the global best locations
- w, c_1, c_2: empirical parameters
- r_1, r_2: random numbers drawn from $U(0, 1)$

PSO search



Firefly Algorithm

The firefly algorithm was inspired by the manner in which fireflies flash their lights to attract mates. In the firefly algorithm, each individual in the population is a firefly and can flash to attract other fireflies. At each iteration, all fireflies are moved toward all more attractive fireflies. A firefly x_a is moved toward a firefly x_b with greater attraction according to

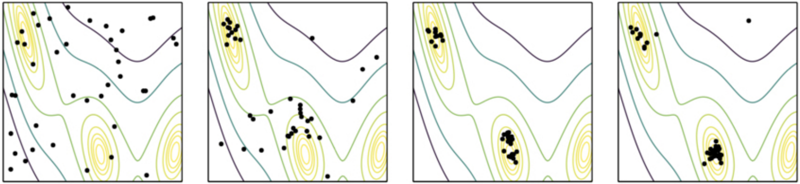
$$x_a \leftarrow x_a + \beta I(||x_b - x_a||)(x_b - x_a) + \alpha\epsilon$$

where I is the intensity of the attraction and β is the source intensity. When $\beta = 0$, it returns to a random walk. where ϵ is drawn from a zero-mean, unit covariance multivariate Gaussian, and α scales the step size. The resulting update is a random walk biased toward brighter fireflies The intensity I decreases as the distance r between the two fireflies increases and is defined to be 1 when $r = 0$. It can be approximated as

$$I(r) = e^{-\gamma r^2}$$

Firefly search

Firefly search with $\alpha = 0.5$, $\beta = 1$,and $\gamma = 0.1$ applied to the Branin function.



Summary

- Population methods use a collection of individuals in the design space to guide progression toward an optimum.
- Genetic algorithms leverage selection, crossover, and mutations to produce better subsequent generations.
- Particle swarm optimization and the firefly algorithm include rules and mechanisms for attracting design points to the best individuals in the population while maintaining suitable state space exploration.
- Population methods can be extended with local search approaches to improve convergence.