Numerical Optimization
12/12/2025

Numerical optimization is an important mathematical tool in many scientific, technical, and operational fields. It enables complex problem solving in finding optimal solutions in engineering, physics, and biology. It also optimizes operations in logistics, finance, and manufacturing. This article provides a brief summary of various problem types and the solution methods. Some examples of complex functions are provided as well.

General Form

The optimization problems are to find the minimum (or maximum) of an objective function, f, and has the following general form:

$$\min_{x \in R^n} f(x)$$

subject to (*s.t.*) the constraints:

$$g_i(x) = 0 \qquad i = 1, 2, \ldots, p$$
$$h_j(x) \geq 0 \qquad j = 1, 2, \ldots, q$$

where $f, g,$ and h are assumed to be differentiable over R^n. The feasible set, Ω, is defined as:

$$\Omega = \{x \in R^n \mid g(x) = 0, h(x) \geq 0\}$$

$x^* \in R^n$ is called a global minimizer *iff* $x^* \in \Omega$ and $\forall x \in \Omega$: $f(x) \geq f(x^*)$. x^* is therefore the solution to the optimization problem. However, very often instead of finding the global minimizer, $x^* \in R^n$ is called a local minimizer *iff* $x^* \in \Omega$ and there exists a neighborhood N of x^* such that $\forall x \in (\Omega \cap N)$: $f(x) \geq f(x^*)$.

Types of Optimization Problems

Non-linear programming: The optimization problems with $f, g,$ and h all assumed to be continuously differentiable.

Linear programming: The optimization problems with $f, g,$ and h all *affine* (functions that preserve straight lines, parallelism, and ratios of distances on a line).

Quadratic Programming: The optimization problems with constraints g and h *affine* but f a linear-quadratic function.

Convex Optimization Problems: The optimization problems with convex feasible set, Ω, and convex objective function, f. A set $\Omega \subset R^n$ is convex if $\forall x, y \in \Omega, t \in [0, 1]$: $x + t(y - x) \in \Omega$. A function is convex if Ω is convex and $\forall x, y, \in \Omega, t \in [0, 1]$: $f(x + t(y - x)) \leq f(x) + t(f(y) - f(x))$. In other words, all connecting lines lie inside set and all scants are above graph.

Unconstrained Optimization Problems: The optimization problems with no constraints, or the feasible set is R^n.

Non-differentiable Optimization Problems: The optimization problems with one or more of $f, g,$ and h non-differentiable. A few solvers, ***e.g.***, Nelder-Mead, may be used to solve these problems.

Mixed-Integer Programming: The optimization problems have the following general form:

$$\min_{\substack{x \in R^n \\ z \in Z^m}} f(x, z)$$

subject to (***s.t.***) the constraints:

$$g_i(x, z) = 0 \qquad i = 1, 2, \ldots, p$$
$$h_j(x, z) \geq 0 \qquad j = 1, 2, \ldots, q$$

Optimization Algorithms

Golden Section Search: Instead of dividing the search zone in exactly half and the binary search method, the golden ratio ($\phi = (\sqrt{5} - 1)/2$) search divides the search zone $[a, b]$ into unequal parts. The objective function, f, is evaluated at $x_1 = a + (1 - \phi)(b - a)$ and $x_2 = a + \phi(b - a)$. If $f(x_1) < f(x_2)$, the search continues in $[a, x_1]$, otherwise in $[x_2, b]$.

Nelder-Mead: Divides the search zone in n parts using the simplex method and evaluate the objective function, f, at $p_1, p_2, \ldots, p_n$. If the best value satisfies the goal, the procedure stops. Otherwise, p_i with the highest value is replaced with a new point and repeat the procedure.

Conjugate Gradient (CG): Uses the initial guess to compute the search direction, $-\nabla f(x_0)$. Select the next point by moving to the search direction at step size λ. Repeat the procedure until $\nabla f(x^*) = 0$. This is also called gradient descent used in neural networks.

Newton-Raphson: The method approximates the first derivative of the objective function starting from the initial guess, x_n, $f'(x_{n+1}) = f'(x_n) + f"(x_n)(x_{n+1} - x_n)$. Set the first derivative to zero and solve for $x_{n+1} = x_n - f'(x_n)/f"(x_n)$ until $f'(x_n)$ is sufficiently small.

Broyden-Fletcher-Goldfarb-Shanno (BFGS): Similar to Newton-Raphson, BFGS is specifically for multivariate optimization thereby the second derivative in Newton-Raphson procedure is replaced with Hessian matrix. Hessian matrix calculation is computationally very expensive, as such, L-BFGS-B is an improvement by more efficient memory usage.

Simulated Annealing (SANN): Intends to find the global minimum of the objective function, SANN is a probabilistic method by exploring the search space with a decreasing probability over time. Potentially worse solutions may be accepted in the process. However, in this way, it can escape the traps of local optima.

Examples

Figure 1 shows a function defined as below:

$$f(x, y) = sin(x) + sin(y) + cos(3xy) + x^2 + y^2$$

The R function nlm(f,c(0,0))$estimate with an initial guess of 0, 0 found the minimum at x=-0.9425063 and y= -0.9425063.

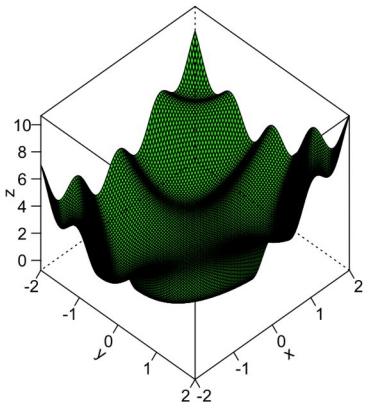

Figure 1, An example function with minimum found at x=-0.9425063 and y= -0.9425063.

Figure 2 shows the Himmelblau function:

$$f(x, y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2$$

The R function optim(c(-4,-4), f)$par with the initial guess at -4, -4 found the minimum at x=-3.779347 and y=-3.283172.

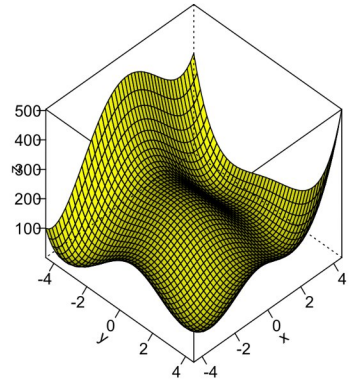

Figure 2 Himmelblau function.

Figure 3 shows the Goldstein-Price function:

$$f(x,y) = \left(1 + (x+y+1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)\right)$$
$$\times \left(30 + (2x-3y)^2(18 - 32x + 12y^2 + 48y - 36xy + 27y^2)\right)$$

The R function optim(c(1,1), f, method="Nelder-Mead")$par with the initial guess at 1, 1 found a local minimum at x=1.2000535 and y=0.8000135. However, the SANN method found the global minimum at x=0.001742546 and y=-0.998184680.

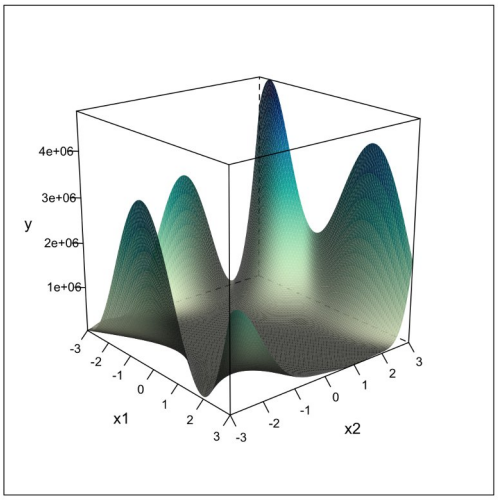

Figure 3, Goldstein-Price function

Figure 3 shows the “Egg Crate” function:

$$f(x,y) = -(y+47)\sin(\sqrt{|y + x/2 + 47|}) - x\sin(\sqrt{|x - (y+47)|})$$

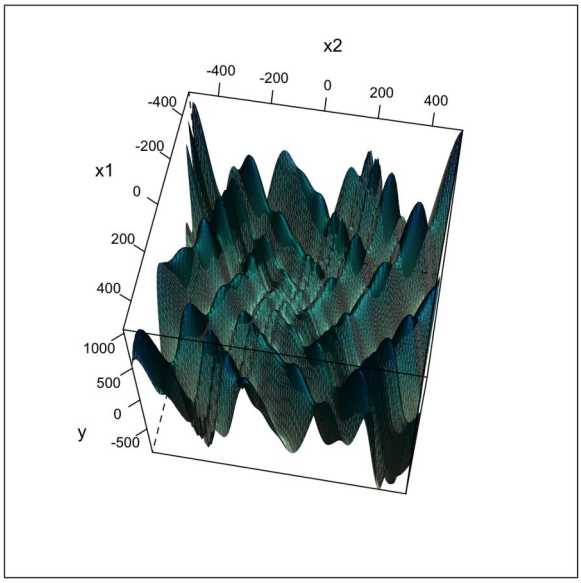

Figure 4, “Egg Crate” function.

The R function optim(c(500,400), f, method="Nelder-Mead")$par with the initial guess at 500, 400 found a local minimum at x=482.3553 and y=432.8814. The SANN method, on the other hand, found the global minimum at x=522.1481 and y=413.3001.

Figure 5 shows the Styblinski-Tang function:

$$f(x) = \frac{1}{2}\sum_{i=1}^{n} x_i^4 - 16x_i^2 + 5x_i$$

The R function optim(c(-5,-5), f, method="Nelder-Mead")$par with the initial guess at -5, -5 found a local minimum at x=-2.902994 and y=-2.903284. The SANN method found a similar solution at x= -2.904041 and y=-2.902545 as the global minimum.

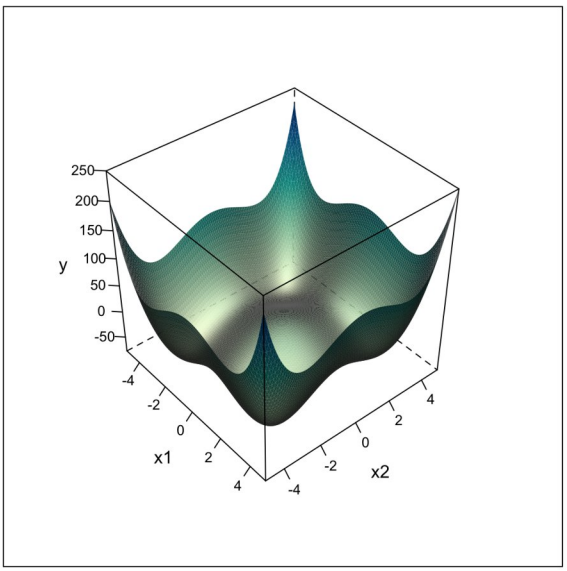

Figure 5. Styblinski-Tang function

Concluding Remarks

Many methods have been developed for solving the optimization problems. Some of them (Newton-Raphson, Conjugate Gradient, and BFGS etc) require the function's derivative information or the Hessian matrix (for multivariate problems). These methods are therefore computationally expensive. Other methods (Golden Ratio, Nelder-Mead, etc) do not require the function's derivative information. These methods are essentially search procedures but quite effective. Conjugate Gradient (CG) method also performs finite difference to estimate the derivative information, if the derivative function is not already provided. Therefore, these "derivative-free" or "gradient-free" methods have been widely used for solving optimization problems. Simulated annealing promises to escape local minima and find the global minimum if the original guess is not favorable to the global minimum.