Chapter 2
Optimization Algorithms

Xin-She Yang

Abstract. The right choice of an optimization algorithm can be crucially important in finding the right solutions for a given optimization problem. There exist a diverse range of algorithms for optimization, including gradient-based algorithms, derivative-free algorithms and metaheuristics. Modern metaheuristic algorithms are often nature-inspired, and they are suitable for global optimization. In this chapter, we will briefly introduce optimization algorithms such as hill-climbing, trust-region method, simulated annealing, differential evolution, particle swarm optimization, harmony search, firefly algorithm and cuckoo search.

2.1 Introduction

Algorithms for optimization are more diverse than the types of optimization, though the right choice of algorithms is an important issue, as we discussed in the first chapter where we have provided an overview. There are a wide range of optimization algorithms, and a detailed description of each can take up the whole book of more than several hundred pages. Therefore, in this chapter, we will introduce a few important algorithms selected from a wide range of optimization algorithms [4, 27, 31], with a focus on the metaheuristic algorithms developed after the 1990s. This selection does not mean that the algorithms not described here are not popular. In fact, they may be equally widely used. Whenever an algorithm is used in this book, we will try to provide enough details so that readers can see how they are implemented; alternatively, in some cases, enough citations and links will be provided so that interested readers can pursue further research using these references as a good start.

Xin-She Yang
Mathematics and Scientific Computing,
National Physical Laboratory,
Teddington, Middlesex TW11 0LW, UK
e-mail: xin-she.yang@npl.co.uk

S. Koziel & X.-S. Yang (Eds.): Comput. Optimization, Methods and Algorithms, SCI 356, pp. 13–31.
springerlink.com

2.2 Derivative-Based Algorithms

Derivative-based or gradient-based algorithms use the information of derivatives. They are very efficient as local search algorithms, but may have the disadvantage of being trapped in a local optimum if the problem of interest is not convex. It is required that the objective function is sufficiently smooth so that its first (and often second) derivatives exist. Discontinuity in objective functions may render such methods unsuitable. One of the classical examples is the Newton's method, while a modern example is the method of conjugate gradient. Gradient-base methods are widely used in many applications and discrete modelling [3, 20].

2.2.1 Newton's Method and Hill-Climbing

One of the most widely used algorithms is Newton's method, which is a root-finding algorithm as well as a gradient-based optimization algorithm [10]. For a given function $f(x)$, its Tayler expansions

$$f(x) = f(x_n) + (\nabla f(x_n))^T \Delta x + \frac{1}{2}\Delta x^T \nabla^2 f(x_n)\Delta x + ..., \tag{2.1}$$

in terms of $\Delta x = x - x_n$ about a fixed point x_n leads to the following iterative formula

$$x = x_n - H^{-1}\nabla f(x_n), \tag{2.2}$$

where $H^{-1}(x^{(n)})$ is the inverse of the symmetric Hessian matrix $H = \nabla^2 f(x_n)$, which is defined as

$$H(x) \equiv \nabla^2 f(x) \equiv \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial f^2}{\partial x_n^2} \end{pmatrix}. \tag{2.3}$$

Starting from an initial guess vector $x^{(0)}$, the iterative Newton's formula for the nth iteration becomes

$$x^{(n+1)} = x^{(n)} - H^{-1}(x^{(n)})\nabla f(x^{(n)}). \tag{2.4}$$

In order to speed up the convergence, we can use a smaller step size $\alpha \in (0, 1]$ and we have the modified Newton's method

$$x^{(n+1)} = x^{(n)} - \alpha H^{-1}(x^{(n)})\nabla f(x^{(n)}). \tag{2.5}$$

It is often time-consuming to calculate the Hessian matrix using second derivatives. In this case, a simple and yet efficient alternative is to use an identity matrix I to approximate H so that $H^{-1} = I$, which leads to the quasi-Newton method

$$x^{(n+1)} = x^{(n)} - \alpha I \nabla f(x^{(n)}). \tag{2.6}$$

In essence, this is the steepest descent method.