5

Introduction to Foraging-Inspired Algorithms

The observation that the foraging processes of animals often entail a search for resources whose location is not known with certainty in advance bears a clear parallel with ‘a search for good locations on a partially observable terrain’. This has led to the development of a significant literature which takes metaphorical inspiration from the foraging strategies of various organisms in order to design powerful search algorithms. In the last three chapters we have introduced elements from various literatures which are relevant to the design of these algorithms.

In this chapter we aim to set up the following sections of the book, which discuss the specifics of a wide range of foraging-inspired algorithms. As noted in Chap. 1, we restrict our attention to optimisation algorithms, so we begin in this chapter by providing a brief introduction to typical optimisation problem settings.

A drawback of many discussions of foraging-inspired algorithms is that they are usually presented in isolation, with no attempt being made to place the algorithms in a coherent taxonomy. A problem with this approach is that it makes it difficult for a reader to fully appreciate the breadth of algorithmic approaches available, or to critically compare and contrast algorithms which may on the surface have very different metaphorical inspirations. In this chapter we illustrate a number of taxonomies which can be used to group the various foraging algorithms. This leads naturally into a discussion of algorithmic design issues, and we demonstrate that most foraging-inspired algorithms can be described in a relatively compact metaframework. Finally, we discuss the phenomenon of swarm intelligence in the context of foraging-inspired algorithms.

5.1 Characterising an Optimisation Problem

In this book we characterise optimisation as being the search in a solution space for the best solution to a problem of interest, where the quality of a solution can be determined according to some predefined criteria. Below we describe a number of common optimisation settings.
Continuous Optimisation

Perhaps the best-known example of an optimisation problem is where we seek to find the input values which maximise or minimise the output value of a real-valued function. Given a function \( f : X \to \mathbb{R} \), we wish to determine the vector \( x_0 \in X \) such that \( f(x_0) \leq f(x) \) for all \( x \in X \) (for a minimisation problem) or, alternatively, \( f(x_0) \geq f(x) \) for all \( x \in X \) (for a maximisation problem).

Constrained Optimisation

Frequently, optimisation problems will be subject to constraints and not all solutions will be considered feasible. A constrained optimisation problem (assuming that the objective function is to be maximised) can be stated as follows. Find a vector \( x = (x_1, x_2, \ldots, x_d)^T, x \in \mathbb{R}^d \), so as to:

\[
\text{Maximise: } f(x) \tag{5.1}
\]

subject to

- inequality constraints: \( g_i(x) \leq 0, \quad i = 1, \ldots, j \) \tag{5.2}
- equality constraints: \( h_i(x) = 0, \quad i = 1, \ldots, k \) \tag{5.3}
- boundary constraints: \( x_i^{\text{min}} \leq x_i \leq x_i^{\text{max}}, \quad i = 1, \ldots, d \) \tag{5.4}

where \( j \) and \( k \) are the numbers of inequality and equality constraints, respectively. Not all constrained optimisation problems will necessarily have all three of the above categories of constraints.

Multiobjective Optimisation

Another common scenario in optimisation problems is that they may be multiobjective. In this case the aim is to optimise over several, possibly conflicting, objectives. The solution space may also be constrained in that all solutions may not be feasible. A multiobjective problem can be generally formulated as follows.

Assume that there are \( n \) objectives \( f_1, \ldots, f_n \) and \( d \) decision variables \( x_1, \ldots, x_d \) with \( x = (x_1, \ldots, x_d) \), and that the decision maker is seeking to minimise the multiobjective function \( y = f(x) = (f_1(x), \ldots, f_n(x)) \). The problem, therefore, is to find the set (region) \( R \subseteq \mathbb{R}^d \) of vectors \( x = (x_1, x_2, \ldots, x_d)^T \), where \( x \in \mathbb{R}^d \), in order to:

\[
\text{Minimise: } y = f(x) = (f_1(x_1, \ldots, x_d), \ldots, f_n(x_1, \ldots, x_d)) \tag{5.5}
\]

subject to

- inequality constraints: \( g_i(x) \leq 0, \quad i = 1, \ldots, j \) \tag{5.6}
- equality constraints: \( h_i(x) = 0, \quad i = 1, \ldots, k \) \tag{5.7}
- boundary constraints: \( x_i^{\text{min}} \leq x_i \leq x_i^{\text{max}}, \quad i = 1, \ldots, d \). \tag{5.8}