Can the processes of natural evolution be mimicked to create robots or autonomous agents? This question embodies the most fundamental goals of evolutionary robotics (ER). ER is a field of research that explores the use of artificial evolution and evolutionary computing for learning of control in autonomous robots, and in autonomous agents in general.

In a typical ER experiment, robots, or more precisely their control systems, are evolved to perform a given task in which they must interact dynamically with their environment. Controllers compete in the environment and are selected and propagated based on their ability (or fitness) to perform the desired task. A key component of this process is the manner in which the fitness of the evolving controllers is measured.

In ER, fitness is measured by a fitness function or objective function. This function applies some given criteria to determine which robots or agents are better at performing the task for which they are being evolved. Fitness functions can introduce varying levels of a priori knowledge into evolving populations. Some types of fitness functions encode the important features of a known solution to a given task. Populations of controllers evolved using such functions then reproduce these features and essentially evolve control systems that duplicate an a priori known algorithm. In contrast to this, evolution can also be performed using a fitness function that incorporates no knowledge of how the particular task at hand is to be achieved. In these cases all selection is based only on whether robots/agents succeed or fail to complete the task. Such fitness functions are referred to as aggregate because they combine the benefit or deficit of all actions a given agent performs into a single success/failure term.

Fitness functions that select for specific solutions do not allow for fundamentally novel control learning. At best, these fitness functions perform some degree of optimization, and provide a method for transferring known control heuristics to robots. At some level, selection must be based on a degree of
overall task completion independent of particular behaviors or task solution features if true learning rather than simple optimization or transference is to be achieved.

Aggregate fitness functions measure overall task completion. However, they can suffer from an inability to produce non-random selection in nascent un-evolved populations. If the task is too difficult, it is likely that none of the randomly initialized controllers will be able to make any meaningful progress toward completing the overall task.

This chapter investigates how aggregate fitness functions have been and continue to be used in ER, what levels of success they have generated relative to other fitness measurement methods, and how problems with them might be overcome.

4.1 Introduction

A distinction can be made between what a robot does and how it does it. That is, there is a difference between the task that a robot is to perform, and the manner in which it performs or solves the task. For example, consider a robot that is to be designed to move toward a light source (phototaxis). The robot’s task is phototaxis, but there are many ways in which this task could be performed. For instance, the robot might detect the light source, turn toward it, and then move forward until it collides with the source. Another solution might be for the robot to just wander around in its environment until it detected a threshold magnitude of light indicating it was near the source, at which point it would stop.

In general there are many solutions (of varying quality) to any given task. Determining the relative qualities of different solutions is essential for control learning. In artificial evolution-based forms of learning, fitness functions make this determination in an automatic or algorithmic way. The distinction between task and task solution defines two broad classes of fitness functions, namely behavioral fitness functions and aggregate fitness functions, and these will be a central focus of discussion in the following sections of this chapter.

Most autonomous robot systems are currently programmed by hand to perform their intended tasks. Learning to perform non-trivial tasks remains a largely unsolved problem in autonomous robotics. Evolutionary robotics approaches the problem of autonomous control learning through population-based artificial evolution. ER methods bear a great deal of similarity to other approaches to controller learning in autonomous robots. In particular, most learning methods require an objective function, and although the discussion in this chapter is focused on experimental work that involved population-based learning, most of the issues related to fitness evaluation are directly relevant to any autonomous system that is intended to learn basic control or behavior in a dynamic environment. We should point out here that other applications of machine learning in robots that are not aimed at learning the