Chapter 4
Measurement and Fitness Function

Martin A. Trefzer and Andy M. Tyrrell

4.1 Introduction

One of the greatest challenges in Evolvable Hardware—in particular when dealing with intrinsic evolution—is to monitor and assess the performance of an (evolving) circuit or system in order to arrive at a solution that ultimately meets the design specifications. In order to be able to accurately assess performance it is necessary to accurately measure not only input/output characteristics, but also properties such as power consumption, resource consumption and temperature. From these measurements desired performance characteristics can then be calculated; the formula (or algorithm) that achieves this is called the fitness function. Once more borrowing terminology from biology, the fitness of a design under test represents its performance on one or more previously defined objectives. The fitness value is then used by the evolutionary algorithm to rank a population of candidate solutions (individuals), hence to decide which ones shall survive and produce offspring, and which ones shall be discarded. Note that although this chapter is mostly about electronic systems, many of the problems addressed are generic to all evolved physical systems and just the way certain properties manifest themselves changes. For example, in electronic circuits “device variability” occurs as a result of the manufacturing process and the equivalent in biological organisms would be “cell diversity”, which occurs due to how growth takes place.

From this, we can identify a number of areas where the designers of evolutionary algorithms often face major challenges:

1. **Measuring Hardware**: It is often not trivial to measure the performance characteristics of a device with sufficient speed and accuracy, and often there is a trade-off to be made between the two. Measurement circuitry or instruments must be included in the evolution system in order to be able to feed information into the evolutionary optimisation loop. Another dimension of complexity added to measurement is the input pattern order problem (discussed in Section 4.3.1).
Particularly when testing more complex circuits with hidden internal states, e.g. memory, capacity or registers, and/or a larger number of inputs and outputs, it becomes an intractable problem to perform an exhaustive measurement of all possible states of a device or system in each evolutionary step.

2. **Defining Objectives:** Assuming all performance metrics can be measured, it is then necessary to decide which ones to take into account during optimisation, which ones to combine into a single fitness value (and how) and which ones to consider separately. The selection of appropriate objectives is not an easy task either, because how to combine objectives which require a trade-off, such as *minimise power consumption* and *maximise speed*, is not obvious.

3. **Calculating Fitness:** Once it has been decided which objectives, or combinations of objectives, shall be optimised, the next challenge of designing suitable fitness functions arises. The first big question is how to combine and weigh the values measured for all objectives in order to provide a search landscape that is sufficiently smooth and does not feature more local optima than necessary. In the worst case, it is even possible to misguide the search algorithm. For example, when adding up (or multiplying) two differently weighted objective values (Fitness = $\alpha_1 \times A + \alpha_2 \times B$) the resulting fitness value does not distinguish, for instance, between the two cases where (i) $\alpha_1 = 0.5$, $\alpha_2 = 2.0$ and $A = 2.0$, $B = 0.5$, and (ii) $\alpha_1 = 2.0$, $\alpha_2 = 0.5$ and $A = 0.5$, $B = 2.0$.

Unfortunately, evolution is particularly good at finding specific solutions which are only valid for exactly the set of inputs used and the environment presented during the evolutionary optimisation process. Even when certain parameters are varied, e.g. the location on a given hardware substrate where the candidates are tested, or the order of the input vectors, evolution is likely to produce circuits that only meet these minimal requirements. In the worst case, an evolved circuit can be just a pattern generator that always generates the desired output irrespective of the applied inputs. Of course, resulting circuits with these properties are far from being fully functional and usable.

Due to the fact that the structure and behaviour of evolving designs are generally unknown, since EAs are usually designed to work on a black box (Imamura et al., 2000), the only truly “safe” option is to include randomness in all aspects where evolution is expected to converge towards generic solutions, e.g. inputs and environment, while the evolutionary optimisation process takes place. This is because outputs of different types of circuits are prone to be misjudged in their own ways as repeatedly performing measurements using the same fixed input test patterns yields different results: digital sequential circuits possess hidden internal states; digital combinational circuits comprise critical paths and delays; and analogue circuits contain internal (parasitic) capacitances and resistances and are affected by variability of device characteristics.

Therefore, this chapter will discuss the challenges highlighted with respect to analogue and digital hardware and will provide some solutions that are proven to—to a certain extent—resolve issues related to these challenges: measuring hardware, defining objectives and calculating fitness.