УДК

L. Bokati, L. Glaese, B. Molina, and V. Kreinovich
University of Texas at El Paso, El Paso, Texas, USA

A POSSIBLE ALTERNATIVE DEFINITION OF AVERAGE-CASE FEASIBILITY

Some algorithms are feasible and some are not: reminder. From the practical viewpoint, some algorithms are feasible, while some are not: algorithms that require an unrealistic amount of computation time are clearly not practically feasible.

Two ways to gauge computation time. There are two ways to gauge how the computation time T(n) depends on the length n of the input:

- we can use worst-case complexity, i.e., the longest time $t(n) = \max \{t(x): \text{len}(x) = n\}$ that the algorithm takes on all the inputs of length n, or
- we can use average-case complexity, i.e., the mean value $m(n) = E_n[t(x)]$ of the computation time on all the inputs of length n.

How feasibility is defined for worst-case complexity. For worst-case complexity, feasibility is defined as polynomial boundedness: an algorithm is feasible if there exists a polynomial P(n) such that for all n, the time t(n) does not exceed P(n).

Comment. This is not a perfect definition. For example, according to this definition, an algorithm that requires time 10^{100} for all inputs is feasible -- since its computation time is a constant, and every constant is a polynomial.

However, such cases are rare. In most practical situations, this definition correctly reflects the intuitive idea of feasibility.

The usual definition of worst-case feasibility does not change if we switch to a feasibly equivalent device. The above definition has the following useful property: whether an algorithm is feasible or not does not change if we replace the computational device with another one which is feasibly (= polynomial) related to the original one – i.e., for which the computation times s(x) and t(x) are mutually polynomially bounded: s(x) does not exceed P(t(x)) and t(x) does not exceed P(s(x)) for some polynomial P(n).

A seemingly natural definition of average-case feasibility. At first glance, it may seem natural to expect that average-case feasibility be defined as the existence of a polynomial bound on the average-case complexity m(n). Let us call such algorithms *average-case feasible on a given computational device*.

It may indeed sound reasonable to call such algorithms average-case feasible, but if we make this a definition of average-case feasibility, then we have a problem: what is feasible on one computational device is not necessarily feasible on a feasibly equivalent computational device.

Current definition of average-case feasibility. To retain the invariance property, computer science uses a different definition of average-case feasibility: namely, average-case feasibility is defined as the existence of a polynomial bound on the expected value $m_a(n) = E[(t(x))^a]$ of some power $(t(x))^a$ of computation time t(x), for some $a > 0$.

An additional useful property of the current definition. One of the useful properties of the current definition of average-case feasibility is that every algorithm which is feasible in the usual worst-case sense is also average-case feasible in the sense of this definition.

Towards an alternative definition. In terms of the above definition of average-case feasibility of a computational device, the current definition of average-case feasibility means the following: an algorithm is average-case feasible if it is average-case feasible on *some* feasibly equivalent computational device.

This reformulation naturally leads to the following alternative definition of average-cases feasibility: an algorithm is average-case feasible if it is average-case feasible on *all* feasibly equivalent computational devices.

Relation between the two definitions. The new definition describes a narrower class of algorithms: the requirement to be feasible on *all* feasibly equivalent devices is clearly stronger than the requirement to be feasible on one of the possible feasibly equivalent devices.

Properties of the new definition. Similarly to the current definition of average-case feasibility, the new definition that also does not change if we replace the original computational device by a feasibly equivalent one

The new definition also has the same additional property as the current one: every algorithm which is feasible in the usual worst-case sense is also average-case feasible in the sense of this new definition.

Our hope. We hope that the study of this stronger alternative definition of average-case feasibility can help us better understand what is feasible and what is not.