WHY SOMETIMES PROBABILISTIC ALGORITHMS
CAN BE MORE EFFECTIVE

Farid M. Ablaev
Department of Mathematics
Kazan State University
ul. Lenina 18
420000 Kazan
USSR

Rūsiņš Freivalds
Computing Center
Latvian State University
Blvd. Raiņa 29
226250 Riga
USSR

For several problems there exist probabilistic algorithms which are more effective than any deterministic algorithms solving these problems. For other problems probabilistic algorithms do not have such advantages. We are interested in understanding, why it is so and how to tell one kind of the problems from another.

Of course, we are not able to present a final answer to these questions. We have just chosen a sequence of examples that can increase our ability to make right predictions whether or not the advantages of probabilistic algorithms can be proved for a given problem.

Since probabilistic algorithms are widely used, the question arise naturally, whether probabilistic algorithms can solve a problem such that no deterministic algorithm can solve it. The very first answer to such a question was found by de Leeuw et al. [20], and it was negative. They proved that under reasonable restrictions on the probabilistic parameters of the machine, probabilistic machines can compute only recursive functions, i.e. functions computable by deterministic machines. Later positive answers were found as well [28, 4, 7].

Probabilistic machines used in our paper are the same as their deterministic counterparts but the probabilistic machines can in addition make use at each step of a generator of random numbers with a finite alphabet, yielding its output values with equal probabilities and in accordance with a Bernoulli distribution, i.e. independently of the values yielded at other instants.

We say that the probabilistic machine M recognizes language L with probability p if M, when working on an arbitrary x, yields a 1 if x ∈ L and yields a 0 if x /∈ L, with a probability no less than p. We say in the case p > 1/2 that L is recognized with an isolated cut-point.

1. PROBLEMS ALLOWING MULTIPLE-VALUED RESULTS

The very first results on advantages of probabilistic algorithms over deterministic ones were proved for problems where the result is not determined uniquely.
but can be chosen from a certain set. Let us consider some examples.

S.V. Yablonskij [28] proved that a probabilistic Turing machine can produce an infinite sequence of Boolean functions such that: a) the running time to produce the n-th element of the sequence is polynomial, and b) probability of the following event equals 1: all but a finite number of the elements in the sequence are Boolean functions with nearly maximal circuit complexity. (Different performances of the probabilistic machine produce different sequences of the Boolean functions).

J.M. Barzdin [4] constructed a recursively enumerable set such that no deterministic Turing machine can enumerate an infinite subset of its complement but there is a probabilistic Turing machine which can enumerate with arbitrarily high probability 1-e infinite subsets in the complement of the given set. (Different performances of the probabilistic machine produce different subsets).

A.V. Vaiser [27] proved that the function log log log x can be approximated by a probabilistic Turing machine in time x log log x, i.e. in less time than any deterministic Turing machine can do this. R. Freivalds [10] improved this upper bound and showed that log log x can be approximated by a probabilistic Turing machine in time x log log x, the function log log log x can be approximated in time x log log log log x, etc. (Different performances of the probabilistic machine can produce different results but with high probability they are nearly the same).


Let A \in \mathbb{N}, B \in \mathbb{N}. We say that A is m-reducible to B (A \leq_m B) if there is a total recursive function f of one argument such that x \in A \iff f(x) \in B holds for all x \in \mathbb{N}.

We say that A is probabilistically m-reducible to B (A \leq_{m-prob} B) if there is a probabilistic Turing machine such that for arbitrary x and i it produces results y so that the probability of the event (x \in A \iff y \in B) exceeds 1-1/i. (Different performances of the probabilistic machine may produce different y's for the same x and i).

Let z \in \{m, m-prob\}. We say that a set L is z-complete if: 1) L is recursively enumerable, and 2) A \leq_z L for arbitrary recursively enumerable set A.

Since A \leq_m B implies A \leq_{m-prob} B, every m-complete set is also m-prob-complete but not vice versa.

THEOREM 1.1. (R. Freivalds, [13]). There is an m-prob-complete set which is not m-complete.

For readers familiar with the notions and the standard notation of the recursive function theory (see [24]) we additionally note a new result by R. Freivalds: m-prob-complete sets always are tt-complete but they can be not btt-complete.

Putting the matter in a highly imprecise way, we claim: if your kind of problems is such that the result is not determined uniquely and it can be chosen from an infinite set then there is a problem of your kind which can be solved by a probabilistic algorithm more easily than by any deterministic algorithm.