APPLYING FAST SIMULATION TO FIND THE NUMBER OF GOOD PERMUTATIONS

N. Yu. Kuznetsov

UDC 519.21

A permutation \((s_0, s_1, \ldots, s_{N-1})\) of symbols 0, 1, \ldots, \(N-1\) is called good if the set \((t_0, t_1, \ldots, t_{N-1})\) formed according to the rule \(t_i = i + s_i \mod N\), \(i = 0, 1, \ldots, N-1\), is also a permutation. A fast simulation method is proposed. It allows the number of good permutations to be evaluated with high accuracy for large \(N\) (in particular, \(N > 100\)). Empirical upper and lower bounds for the number of good permutations are presented and verified against numerical data.

Keywords: permutation, fast simulation method, unbiased estimate, sample variance, relative error.

INTRODUCTION

We propose here a new and quite efficient approach to solving one of the most difficult combinatorial problems of discrete mathematics [1]. This problem was proved to be complicated once it was related to the classical problem of calculating permanents [2].

Let \((s_0, s_1, \ldots, s_{N-1})\) be an arbitrary permutation of symbols \((0, 1, \ldots, N-1)\). Let us generate a new set \((t_0, t_1, \ldots, t_{N-1})\) according to the rule \(t_i = i + s_i \mod N\), \(i = 0, 1, \ldots, N-1\). If this set is also a permutation, then the initial permutation \((s_0, s_1, \ldots, s_{N-1})\) is called good. The purpose of our study is to determine the number of good permutations for as many values of \(N\) as possible.

The term “good” permutation is introduced in [3], where the application of good permutations in cryptography is detailed (see also [4], where the principles of using good permutations in rotor encryption systems are described).

The total number of permutations is equal to \(N!\); therefore, the problem formulated is equivalent to finding the probability that a random permutation is good. Denote this probability by \(P_N\).

It is generally known that \(P_N = 0\) for even values of \(N\). In what follows, we consider only odd values of \(N\). Until recently, exact values of \(P_N\) have been calculated in [3] only for \(N < 19\). Rapid computer progress has made it possible to determine \(P_N\) also for \(N = 21, 23, 25\) (http://www.research.att.com/~njas/sequences/A003111). To calculate \(P_N\) with increase in \(N\) by at least 2 is a challenge and depends primarily on the computer development. Therefore, the main attention is paid to approximate calculation methods, in particular, to asymptotic and statistical ones.

Cooper and Kovalenko assumed in [5, 6] that \(P_N \leq e^{-cN}\), where \(c\) is a constant to be estimated. They proved in [5] that \(c \approx 0.0885\). A stronger estimate is obtained in [6]: \(c \geq \frac{\ln 2}{2} \approx 0.3466\).

Of primary interest are the values of \(P_N\) for large \(N\). Cooper et al. put forward a hypothesis in [3] that

\[ P_N \sim ae^{-cN} \quad (1) \]

as \(N \to \infty\), and \(c \in [0.5; 1]\), and proposed an approximation that allowed generating the predictive estimates of \(P_N\) for \(N = 25, 35, 45, 55\) (though no estimates of the approximation error were given). They also concluded that \(c \approx 0.9538\).

Of few studies that estimate the number of good permutations, noteworthy is [7], where the properties of good permutations are studied and a series of important theorems are proved, which substantially reduces the search area and thus
increases the efficiency of algorithms being created. Among the methods finding good permutations, noteworthy are the telephone-dial algorithm, symmetric algorithm, weighted algorithm, fair-distribution algorithm, sampling method \cite{3}, and the approximating-curve method \cite{3}. The ideas of \cite{7} underlie the symmetric and weighted algorithms. The first four algorithms are deterministic, and the last two are based on statistical methods.

Note also that there are very few attempts to find at least approximate values of $P_N$ for $N > 25$ and to estimate the associated errors. The purpose of our study is to bridge this gap. We propose to use the fast simulation method to estimate the number of good permutations. Such an approach has already been applied to solve some combinatorial problems in discrete mathematics, in particular, to find the permanent of a matrix \cite{8} and to solve the knapsack problem \cite{9}. Although the method we propose is quite simple, it allows generating unbiased estimates and the associated confidence intervals for $P_N$ for rather large $N$ (we present here the estimates of $P_N$ up to $N = 155$) with relatively small time expenditures. Moreover, this algorithm proves relation (1) using practical calculations and specify more exact boundaries for the constant $c$, namely, $0.9825 \leq c \leq 0.9883$. In the last section of the paper, we present the upper and lower estimates for $P_N$ for $N \geq 75$ confirmed with statistical data. These estimates can be used also for $N$ so large that even the fast simulation does not allow generating adequate-accuracy estimates in real time.

**FAST-SIMULATION ALGORITHM**

It is required to arrange the symbols $0, 1, \ldots, N - 1$ in $N$ places. The arrangement algorithm should be organized so as to substantially increase the probability of obtaining a good permutation. To this end, it is necessary to establish the necessary conditions of obtaining a good permutation. In other words, we need criteria that allow determining, at each step of the algorithm, which symbols can be put in a certain place in order that the permutation has chances to be good.

The algorithm consists of two stages:

**Stage I** (telephone-dial algorithm).
1. Let $p_1 = 1$ be the initial value of the estimate, $v_i = 0$, $\mu_j = 0$, $i, j = 0, \ldots, N - 1$.
2. Assume that $r$ takes on the values $m + 1, \ldots, N$ (place number) sequentially. Let us find the set of symbols that can be put in the place $r$:

$$A_r = \{ i : v_i = 0, \mu_k = 0, \text{ where } k = r + i \mod N \}. \quad (2)$$

Denote by $|A_r|$ the number of symbols in the set $A_r$. If $|A_r| = 0$, then Stage I (together with Stage II) is completed and we have $p_1 = 0$ as the estimate (it occurred impossible to generate a good permutation in this realization). If $|A_r| > 0$, then we choose, with the same probability $\frac{1}{|A_r|}$, one of symbols of the set $A_r$. If this symbol is $i$, then assume

$$\hat{p}_1 := \hat{p}_1 - \frac{|A_r|}{N + m + 1 - r}; \quad v_i = 1, \mu_k = 1,$$