Cybernetics and Control Theory

I. E. Maizlin

1964

SovietRxiv

Abstract

Full Text

Cybernetics and Control Theory

I. E. Maizlin

On One Method of Information Retrieval and Its Application in Implementing on a Computer an Algorithm for Finding the Critical Path

(Presented by Academician A. I. Berg on 27 III 1964)

1. Suppose there are n mutually distinct k-digit binary codes $X_0, X_1, \dots, X_{n-1}$. To each code X_i $(i = 0, 1, \dots, n-1)$ we assign a "pseudonumber" —a number N_i^* satisfying the following conditions:

a) $N_0^*, N_1^*, \dots, N_{n-1}^*$ are independent random variables, each of which assumes with equal probability any integer value on the interval $[0, n-1]$.

b) The value N_i^* is uniquely determined by the code X_i $(i = 0, 1, \dots, n-1)$.

The following process of obtaining N_i^*, while satisfying condition b), ensures with sufficient accuracy the fulfillment of condition a). Let $y_0, y_1, \dots, y_{r-1}$ be a sample from a set of numbers distributed uniformly on the interval $[0, 1]$, $r = 2^m$, $m = k/q$, where q is an integer. We divide the k-digit code X_i $(i = 0, 1, \dots, n-1)$ into q m-digit codes $x_i^1, x_i^2, \dots, x_i^q$, and, considering x_i^j $(j = 1, \dots, q)$ as an integer, set

$$N_i^* = \left[n \left\{ \sum_{j=1}^{q} y_{x_i^j} \right\} \right], \tag{1}$$

where $\{\}$ denotes the fractional part of a number, and $[\ ]$ denotes the integer part.

Suppose that in the sequence $N_0^*, N_1^*, \dots, N_{n-1}^*$ there are ν distinct numbers $N_{\alpha_1}^*, N_{\alpha_2}^*, \dots, N_{\alpha_\nu}^*$, and that the pseudonumber $N_{\alpha_j}^*$ $(j = 1, \dots, \nu)$ is possessed by β_j codes of the sequence $X_0, X_1, \dots, X_{n-1}$. The quantity

$$\beta = \frac{1}{n} \sum_{j=1}^{\nu} \beta_j^2$$

may naturally be called the average number of codes that have received the same pseudonumber; β is a random variable.

We shall show that $M\beta = 2 - 1/n$. Indeed:

$$\beta = \frac{1}{n}\sum_{j=1}^{\nu}\beta_j^2 = \frac{1}{n}\sum_{l=0}^{n-1}\gamma_l^2,$$

where γ_l $(l = 0, 1, \dots, n-1)$ is the number of elements of the sequence $N_0^*, N_1^*, \dots, N_{n-1}^*$ equal to l; γ_l is a random variable with the binomial distribution $\left(p = \frac{1}{n}\right.$ by virtue of condition a)$\left.\right)$. Therefore

$$M\beta = \frac{1}{n}\sum_{l=0}^{n-1} M\gamma_l^2 = \frac{1}{n}\sum_{l=0}^{n-1}\left[D\gamma_l + (M\gamma_l)^2\right] = \frac{1}{n}\sum_{l=0}^{n-1}\left[n\frac{1}{n}\left(1-\frac{1}{n}\right) + \left(n\frac{1}{n}\right)^2\right] = 2-\frac{1}{n}.$$

Let us formulate the information problem. There are n mutually distinct k-digit binary codes $X_0, X_1, \dots, X_{n-1}$, arranged consecutively in the memory of a computer, with $n \leqslant 2^k$. It is required to find the numbers of those codes

sets $\{X_i\}$ $(i = 0, 1, \dots, n-1)$, which are equal to the given codes $Y_1, \dots, Y_\theta$. The proposed algorithm for solving this problem consists of two parts (let us call them the A-algorithm and the B-algorithm).

The A-algorithm processes information about the codes $X_0, X_1, \dots, X_{n-1}$. For its operation $2n$ auxiliary quantities $u_0, u_1, \dots, u_{n-1}$ and $p_0, p_1, \dots, p_{n-1}$ are needed, each of which will take values from 0 to $n-1$. Initially $u_0 = u_1 = \dots = u_{n-1} = 0$, and $p_i = N_i^*$ $(i = 0, 1, \dots, n-1)$, where N_i^* is the pseudonumber of the code X_i, obtained by formula (1). Next a recurrent process is carried out, at whose $(n-k)$-th step $(n-k = 1, 2, \dots, n)$ p_k is equated to $u_{N_k^*}$, and the new value of $u_{N_k^*}$ is set equal to k. Here p_k is the number of the nearest code with pseudonumber equal to N_k^*. The equality $p_k = 0$ means that $N_j^* \neq N_k^*$ $(j > k)$. In implementing the A-algorithm on a computer, cn (c is a constant number) machine operations are required.

The B-algorithm solves the stated problem successively for each code Y_j $(j = 1, \dots, \theta)$. For the code Y_j, by formula (1) we determine the pseudonumber $N^*(Y_j)$. We compare the code X_{z_1}, where $z_1 = u_{N^*(Y_j)}$, with the code Y_j. If $X_{z_1} = Y_j$, then the required number is z_1. If $X_{z_1} \neq Y_j$, we compare the code X_{z_2}, where $z_2 = p_{z_1}$, with the code Y_j, and so on until the equality $X_{z_k} = Y_j$ $(z_k = p_{z_{k-1}}$, if $k > 1)$ is satisfied, from which we conclude that the required number is z_k. The number of operations in implementing the B-algorithm on a computer turns out to be equal to $c\theta\beta$, where c is a constant and β is the random variable defined above, i.e.

$$M[c\theta\beta] = c\theta\left(2-\frac{1}{n}\right) < c_1\theta.$$

Let us note that the B-algorithm realizes Johnson' s proposed ([3]) method of indirect chained search for information in a card file.

2. The problem of finding the time characteristics of a plan in the planning of the development of large systems by the critical path method (pert-time) is solved. The statement of the problem, the reduction to logical networks, and some of the terms used in the note are described in detail in ([1]). In the algorithm described below, in contrast to ([1]), the initial information is information about activities, not about events. This eliminates the renumbering process, which requires either a large number of machine operations or substantial restrictions on the independence of information about different parts of the system.

We introduce two "fictitious" activities of zero duration d_{α_0} and $d_{\hat{\alpha}}$. Completion of activity d_{α_0} corresponds to the event "beginning of activities," and for the beginning of activity $d_{\hat{\alpha}}$ the occurrence of the event "end of activities" is necessary.

In the information about activity d_α its code X_α, duration l_α, and the number k_α of activities whose completion is necessary for the beginning of activity d_α, as well as their codes $X_{i_1}, \dots, X_{i_{k_\alpha}}$, are specified. The totality of the codes $X_{i_1}, \dots, X_{i_{k_\alpha}}$ for all α will be called the I-information.

The process of finding the critical activities is divided into a forward-code algorithm and a reverse-code algorithm. In the forward-code algorithm we associate with each activity d_α, in addition to its duration l_α, three more quantities π_α, q_α, and t_α, whose values will change in the course of the algorithm. Initially $t_\alpha = 0$, $q_\alpha = k_\alpha$, and $\pi_\alpha = 0$ for all activities except d_{α_0}, for which $\pi_{\alpha_0} = 1$. After applying the A-algorithm to the I-information, we pass to the forward-code algorithm, one cycle of which is as follows. We take any activity d_β for which $\pi_\beta = 1$. We consider all activities $d_{\beta_1}, \dots, d_{\beta_p}$ for whose start the completion of activity d_β is necessary. To find these activities, the B-algorithm is applied p times over the I-information. For all $j = 1, \dots, p$, if $t_{\beta_j} < t_\beta$, we set $t_{\beta_j} = t_\beta$. Then we decrease k_{β_j} $(j = 1, \dots, p)$ by one and, if k_{β_j} has become zero, increase t_{β_j} by l_{β_j} and set $\pi_{\beta_j} = 1$. After this we set $\pi_\beta = 2$ and begin the next cycle by considering the next activity d_γ, for whi-

$\pi_\gamma = 1$. Under the assumption that the initial logical network is a graph without circuits, with a single initial and a single terminal vertex[2], it is not difficult to show that, if $\pi_{\hat{\alpha}} \neq 1$, there will be found an operation d_γ with $\pi_\gamma = 1$. If $\pi_{\hat{\alpha}} = 1$, then for all operations d_γ $(\gamma \neq \hat{\alpha})$ we have $\pi_\gamma = 2$, and t_γ is equal to the earliest completion time of operation d_γ. In this case $t_{\hat{\alpha}}$ is equal to L, the length of the critical path being sought.

Next the backward-pass algorithm is carried out. To each operation d_α there correspond, besides its duration l_α, three numbers P_α, Q_α, and T_α. First, the initial information about the operations is processed using the A- and B-algorithms, as a result of which the information about operation d_α is supplied with the numbers of the computer memory cells in which information is stored about the operations $d_{i_1}, \dots, d_{i_{K_\alpha}}$ needed for the execution of operation d_α, and the number K_α is determined—the number of operations for whose start the execution of operation d_α is necessary. Before the algorithm starts, $T_\alpha = 0$, $Q_\alpha = K_\alpha$, and $P_\gamma = 0$ for all operations except $d_{\hat{\alpha}}$, for which $P_{\hat{\alpha}} = 1$. One cycle of the backward-pass algorithm consists in the following. We take an operation d_β for which $P_\beta = 1$. Consider the operations $d_{\beta_1}, \dots, d_{\beta_r}$ needed for the start of operation d_β. If $T_{\beta_j} < T_\beta + l_\beta$ $(j = 1, \dots, r)$, set $T_{\beta_j} = T_\beta + l_\beta$. Next decrease the value Q_{β_j} $(j = 1, \dots, r)$ by one and, if Q_{β_j} has become equal to zero, set $P_{\beta_j} = 1$. Then set $P_\beta = 2$ and begin a new cycle by considering the next operation d_γ with $P_\gamma = 1$. Under assumptions analogous to those indicated above, it can be proved that, if $P_{\alpha_0} \neq 1$, there will be found an operation d_γ with $P_\gamma = 1$. If $P_{\alpha_0} = 1$, then for all operations d_α $(\gamma \neq \alpha_0)$ $P_\gamma = 2$, and $L - T_\gamma$ is equal to the latest execution time of operation d_γ, provided that the entire complex of operations is executed in time L.

The operations $d_{i_1}, \dots, d_{i_g}$, for which $t_{i_k} = L - T_{i_k}$ $(k = 1, \dots, g)$, are critical. Any critical path consists of critical operations.

If the size of the computer's main memory is sufficient to store all the information about the operations, then the total number of operations in the implementation of the described algorithm is a random variable with mathematical expectation less than CMn, where C is a constant depending on the type of machine, n is the number of operations in the system, and M is the greatest degree of a vertex[2] of the corresponding logical network.

A program has been written that implements the described algorithm on a computer. The author expresses deep gratitude to Corresponding Member of the Academy of Sciences of the USSR L. A. Lyusternik for his guidance in carrying out this work.

Received
24 III 1964

CITED LITERATURE

[1] G. S. Pospelov, A. I. Teiman, *Izv. AN SSSR, Technical Cybernetics*, No. 4, 60 (1963). [2] C. Berge, *Theory of Graphs and Its Applications*, Moscow, 1962. [3] L. R. Johnson, *Commun. Assoc. Computing Machinery*, 4, No. 5, 218 (1961).

Note: Figure translations are in progress. See original paper for figures.