50383 EXHAUSTIVE MOTION FOR A CCM, 11 JANUARY 1956 (S-070,038) 13,199 OF 7pp

UNCLASSIFIED

~~SECRET~~

EXHAUSTIVE MOTION FOR A CCM

S. S. Cairns
NSA-34
11 January 1956

For an N-wheel CCM machine a procedure is obtained for an efficient exhaustive motion, one which runs through each setting of the machine with as few repeats as possible. The paper is related to Forrest S. Goepper's "'M'-Motion for CCM" (AFSA-412-B6, 6 June 1952, C 25.221.1-503).

EXHAUSTIVE MOTION FOR A CCM

Let $(w_1, \cdots, w_N)$ be the wheels of a CCM, with w_1 fast and w_j stepping whenever a notch is in active position on w_{j-1} $(j = 2, \cdots, N)$. The graph of the machine is a set of cycles. Let m_j be the length of w_j and let the settings on w_j be cyclically numbered $(1, 2, \cdots, m_j)$. A <u>setting of the machine</u> is then a set of integers $(i_1, i_2, \cdots, i_N)$ where i_j is an integer mod m_j. There exists one and only one cycle $C(i_1, \cdots, i_N)$ of the machine through each setting $(i_1, \cdots, i_N)$.

We commence by briefly stating some relevant facts deducible from earlier reports, notably from "A Simplified Statement of the Solution of an N-Wheel Cyclic Structure Problem" by the writer (20 June 1950). Subsequent reports contain computational procedures for certain cases.

The method of analysis in these reports is recurrent with regard to the number of wheels in the machine. Let a <u>C^n-cycle</u> be a cycle of the "submachine" consisting of $(w_1, \cdots, w_n)$ $(n = 1, 2, \cdots, N)$, so that the C^N-cycles constitute

the graph of the entire machine. In passing from the C^n-cycles to the C^{n+1}-cycles, one regards any C^n-cycle C as a wheel with a notch at each point* of C for which a notch on w_n is in active position. This reduces the problem to a sequence of 2-wheel problems. In the passage from C^n-cycles to C^{n+1}-cycles, a central role is played by the following fact.

(A) If there are λ points and V notches on C, then the number of C^{n+1}-cycles arising from C and w_{n+1} is the greatest common divisor of V and m_{n+1}, denoted by $d(V, m_{n+1})$.

The initial step of the recurrency is for $n=1$. The only C^1-cycle is w_1 itself, for which $\lambda = m_1$ and $V = U_1$ (the number of notches on w_1). Hence the number of C^2-cycles is

$$\nu = d(U_1, m_2) . \tag{1.1}$$

and each is of length

$$\lambda_2 = \frac{m_1 m_2}{\nu} . \tag{1.2}$$

In order to take the next step, passing to the C^3-cycles, it is necessary to know the number of notches on each of the

*A point of C is a set $(i_1, \cdots, i_n)$ of numbers of wheel settings for $w_1, \cdots, w_n$ respectively.

ν C^2-cycles. There exist methods for computing the necessary data on the basis of a precise knowledge of the notch patterns of w_1 and w_2. For the next stage, we should need to figure out the exact notch pattern on each C^2-cycle and to use it with the known pattern on w_3. General methods for such computations exist, but practicable procedures have been devised only for cases where restrictions are placed on wheel lengths and patterns.

In the reports cited above, a special numbering system is employed to facilitate the statement of a computational program. The nature of this numbering is irrelevant here. We change it, for simplicity, with the remark that the present notation is different from and not well adapted to the earlier purposes.

Lemma 1. Let $C(i_1, i_2)$ denote the C^2-cycle containing the setting (i_1, i_2) on the 2-wheel machine consisting of w_1 and w_2. Then the ν cycles of this machine are

$$C_i = C(1, i) \qquad (i = 1, \cdots, \nu) . \tag{1.3}$$

This is a direct consequence of the earlier work, save that the j^{th} cycle in the earlier numbering was $C(1, \nu+2-j)$.

SECRET

(B) Lemma 1 means that there is just one of the points (1, i) on each C^2-cycle and that every C^2-cycle goes through just one such point. At the next stage, there is a set of C^3-cycles arising from each of the 2-wheel machines (C_i, w_3).

Corollary. Let n(i) denote the number of C^3-cycles arising from C_i and w_3. These n(i) cycles are

$$C_{ij} = C(1, i, j) \qquad (i = 1, \cdots, \nu;\ j = 1, \cdots, n(i)) \tag{1.4}$$

where C(1, i, j) is the C^3-cycle of the machine (w_1, w_2, w_3) which contains the setting (1, i, j). The number of C^3-cycles is $\sum_{i=1}^{\nu} n(i)$.

The above commences an obvious recurrent process, the ultimate conclusion of which is as follows.

Theorem. Let $C(j_1, \cdots, j_N)$ be the cycle of the entire machine through the setting $(j_1, \ldots, j_N)$. Then the cycles of the machine are precisely the following, each appearing once and only once in this list, where the n-functions are theoretically computable*.

*One can conduct an exhaustive motion (see below) without computing these functions.

$C(1, j_2, \cdots, j_N)$ as the j's range over

$j_2 = 1, 2, \cdots, \nu$

$j_3 = 1, 2, \cdots, n(j_2)$ for each j_2

$j_4 = 1, 2, \cdots, n(j_2, j_3)$ for each pair (j_2, j_3)

.
.
.

$j_N = 1, 2, \cdots, n(j_2, j_3, \cdots, j_{N-1})$ for each set $(j_i, j_3, \cdots, j_{N-1})$

An exhaustive motion, one which runs through each setting of the machine with few repetitions is as follows:

(1) Start with the setting $(1, 1, \cdots, 1)$. Follow CCM motion. When the original setting reappears, let all wheels stand save w_2, which steps once.

(2) Proceed with CCM motion till $(1, 2, 1, \cdots, 1)$ reappears, then again let w_2 alone step

.
.
.

(n) After the second appearance of $(1, n, 1, \cdots, 1)$, let w_2 step again, restoring $(1, 1, \cdots, 1)$

(n+1) Now let w_3 alone step, to get $(1, 1, 2, 1, \cdots, 1)$.

This commences an obvious process taking us through all the cycles in the order

$C(1, 1, 1, \cdots, 1)$
$C(1, 2, 1, \cdots, 1)$
.
.
.
$C(1, n, 1, \cdots, 1)$

$C(1, 1, 2, \cdots, 1)$
$C(1, 2, 2, \cdots, 1)$
.
.
.
$C(1, n, 2, \cdots, 1)$

$C(1, 1, 3, \cdots, 1)$
$C(1, 2, 3, \cdots, 1)$
etc.

This is a lexicographic order in terms of the symbols

$(j_N, j_{N-1}, \cdots, j_2, 1)$

If the n's are known, counters could be used to shift the type of motion. If the n's are not known, devices to recognize an earlier setting might be employed. In the latter case, counters could evaluate the n's .