

Capacity Requirement of a Mail Sorting Device¹

B. K. Bender and A. J. Goldman

A mathematical model of a sorting device suggested by S. Henig is considered. The relevant parameters are r (the number of destinations for the mail) and k (the number of letters entering the device during each cycle of operation). It is shown that the device will never jam if its capacity is at least $(r-1)(k-1)+k$ letters; this is the lowest possible capacity requirement, and for realistic values of r and k is significantly less than the previously known estimate r^2k .

1. Introduction

We deal with a highly idealized mathematical model of a mail sorting device suggested by S. Henig (NBS Electronic Instrumentation Section). For our purposes, the operation of the device can be described² as follows. Mail (to any of r destinations) enters the system. After k letters have entered, the device "asks itself" to which destination it contains the *most* letters.³ All letters to this *predominant destination* are then *dropped out* of the device, another k letters enter, and the process continues.

The device is said to *jam* if, after a dropout, it is still so "full" that entrance of the next k letters would cause an "overflow". We will determine the minimum capacity required of the device to ensure that no jamming occurs. This is of course equivalent to determining (in terms of r and k) the *maximum possible contents* of the device under the dropout rule described above, and we shall work with this alternative version of the problem. The derivation involves nothing more complicated than counting up letters, so that nonmathematician readers should be able to follow the arguments.

2. Statement of Results

In order to describe our results, it is convenient to define

$x(t)$ = number of letters in the device just before the t th dropout. (1)

A. Bruce Clarke⁴ has shown that

$$x(t) \leq r^2k, \text{ for all } t, \quad (2)$$

which proves that the number of letters in the device never exceeds r^2k , so that the device will never jam if its capacity is r^2k or more. Clarke (see footnote 4) remarks that "this upper bound for $x(t)$ is clearly much too crude to be of any practical use;" for example, for

the Philadelphia-mail-type data (with $r=200$, $k=3$) used for numerical illustration, it only informs us that the device will never jam if its capacity is 120,000 or more.

As was pointed out by L. S. Joel (NBS Computation Laboratory), the estimate (2) can be very much reduced using rather simple arguments. It will in fact be shown that

$$x(t) \leq (r-1)(k-1)+k, \text{ for all } t, \quad (3)$$

which proves that the number of letters in the device can never exceed $(r-1)(k-1)+k$, so that a capacity of at least $(r-1)(k-1)+k$ will ensure that the device never jams. For the Philadelphia-type data, this shows that a capacity of 401 (rather than 120,000) is sufficient to prevent jamming. The proof of inequality (3) will be given in section 3.

In addition, it will be shown that the device *can* jam if its capacity is *less* than $(r-1)(k-1)+k$; that is, it is possible (under the dropout rule described above) for the number of letters in the device actually to reach $(r-1)(k-1)+k$. In other words, the estimate in (3) is not only an upper bound for $x(t)$ but is the maximum possible value for $x(t)$:

$$x(t)_{\max \text{ poss}} = (r-1)(k-1)+k \quad (4)$$

This will also be proved in section 3. Section 4 contains some comments on the probabilistic aspects of the problem.

3. Proofs

3.1. Proof of Inequality (3)

It is assumed, of course, that the device had fewer than $(r-1)(k-1)+k$ letters in it at the start of operation. An "indirect proof" will be used; that is, we can tentatively suppose that (3) is false and show that this supposition is untenable.

If (3) is false, there must be a first value t_0 of t (i.e., a first dropout) for which it fails. Since (3) is false for $t=t_0$ but true for $t=t_0-1$, we have

$$x(t_0) > (r-1)(k-1)+k, \quad (5)$$

$$x(t_0-1) \leq (r-1)(k-1)+k. \quad (6)$$

¹ Part of a project supported by the Post Office Department, Office of Research and Engineering.

² The physical device can also operate under "dropout rules" other than the one described below.

³ A rule for breaking "ties" between destinations is also required.

⁴ A. Bruce Clarke, A mathematical model for the Henig sorting machine, partly abstracted in *Ann. Math. Statistics* **29**, 622 (1958).

We continue the argument by pointing out two facts about the contents of the device *directly after the* (t_0-1) -st dropout:

(a) There are more than $(r-1)(k-1)$ letters in the device, and

(b) There are at least k letters in the device to some one destination.

The truth of (a) follows directly from (5) and the fact that only k letters were added between the (t_0-1) -st and t_0 -th dropouts. All letters to some destination left the device in the (t_0-1) -st dropout, so that if (b) were false the device would contain at most $(k-1)$ letters to each of at most $(r-1)$ destinations, or at most $(r-1)(k-1)$ letters in all, contradicting (a). Thus (b) is true.

From (b) and the dropout rule it can be seen that the (t_0-1) -st dropout consisted of at least k letters, so, examining the contents of the device just *before* the (t_0-1) -st dropout, we have

(number of letters to predominant dest.) $\geq k$,
and by (a),

(number of letters to other dest.'s) $> (r-1)(k-1)$;
the last two inequalities together yield

$$x(t_0-1) > (r-1)(k-1) + k, \quad (7)$$

which contradicts (6). Thus the supposition that (3) is false is untenable, so (3) must be true.

3.2. Proof of Equation (4)

We already know, by (3), that the device can never contain *more* than $(r-1)(k-1) + k$ letters. Thus, in order to prove (4), it suffices to exhibit a sequence of possible events leading to a situation in which the device contains *exactly* $(r-1)(k-1) + k$. There are many such sequences, and the one described below is not necessarily the shortest one.

The sequence will be constructed in two stages, beginning with the device empty at $t=0$. At the end of stage 1, the device will contain $k-1$ letters to each of 2 destinations and $k-2$ letters to each of the other $r-2$ destinations. At the end of stage 2, the device will contain exactly $(r-1)(k-1) + k$ letters and the proof of equation (4) will have been completed.

Stage 1: We suppose that the r destinations are numbered in some way. The sequence of possible events begins as follows:⁵ The first k letters entering the device consist of one letter to the first destination and $k-1$ letters to the r th destination; these last $(k-1)$ letters then leave the device in the first dropout. The second k letters to enter the device consist of one letter to the second destination and $k-1$ letters to the r th destination, etc. After the $(r-1)$ -st dropout, the device contains no letters to the r th destination and one letter to each other destination. The r th set of k letters to enter the device consists of one letter to the first destination

and $k-1$ letters to the r th destination, and the process continues as before; after the $(r-1)(k-2)$ -nd dropout the device contains no letters to the r th destination and $k-2$ letters to each of the $(r-1)$ other destinations.

The k letters entering the device between the $(r-1)(k-2)$ -nd and $((r-1)(k-2)+1)$ -st dropouts consist of $k-1$ letters to the r th destination and one letter to some other destination; the device now contains $k-1$ letters to each of 2 destinations and $k-2$ letters to each of the other $r-2$ destinations. In other words, if we define

$$n(t) = \left. \begin{array}{l} \text{number of destinations to which there are} \\ k-1 \text{ letters in the device just before the} \\ t\text{th dropout,} \end{array} \right\} \quad (8)$$

then we have

$$n((r-1)(k-2)+1) = 2. \quad (9)$$

The $((r-1)(k-2)+1)$ -st dropout will consist of $k-1$ letters to whichever of the 2 destinations mentioned above is "preferred" by whatever tie-breaking rule is employed. After this dropout the device will contain no letters to some one destination, $k-1$ letters to some other destination, and $k-2$ letters to each of the $r-2$ remaining destinations.

Stage 2: In this stage matters will be arranged so that

$$n(t) = n(t-1) + 1 \quad (10)$$

for $(r-1)(k-2) + 2 \leq t \leq (r-1)(k-1)$;

i.e., so that after each dropout there are $k-1$ letters in the device to *one more* destination than before. If this is done, then by (9) and repeated application of (10) we have

$$n((r-1)(k-1)) = n((r-1)(k-2) + (r-1)) = r,$$

so that just before the $(r-1)(k-1)$ -st dropout the device contains $k-1$ letters to each of the r destinations. Then $k-1$ of these $r(k-1)$ letters are dropped leaving $(r-1)(k-1)$ in all, and k new letters enter; the device now contains $(r-1)(k-1) + k$ letters, and equation (4) is proved.

To achieve (10) we choose the k letters entering the device between the $(t-1)$ -st and t th dropouts, for $t = (r-1)(k-2) + 2, (r-1)(k-2) + 3, \dots, (r-1)(k-1)$, to consist of $k-1$ letters to the destination just previously dropped out, and one letter to a destination which previously had $k-2$ letters to it in the device.

4. Probabilistic Aspects of the Capacity Problem

We have been concerned above with choosing the capacity of the device so that jamming *never* occurs. In practice we might well have the more modest aim of keeping the frequency of jamming down to some tolerable level, and so could get by with a

⁵ For a more formal description of the following process, let $R(t)$ denote the remainder when t is divided by $r-1$, except that $R(t) = r-1$ when t is a multiple of $r-1$. The for $1 \leq t \leq (r-1)(k-2)$, the k letters entering the device between the $(t-1)$ -st and t th dropouts are to consist of one letter to the $R(t)$ -th destination and $k-1$ letters to the r th destination.

smaller capacity than that indicated by eq (4). On the one hand this leads to the "policy" question of what jamming frequencies are to be considered intolerable; on the other hand, it leads to the purely mathematical problem of what capacity is needed in order to keep jamming below these frequencies.

This last problem seems very difficult. As was pointed out (unpublished memorandum, July 1956) by J. R. Rosenblatt (NBS Statistical Engineering Laboratory), the behavior of the device constitutes a stochastic process (indeed, a Markov chain) governed by the probabilistic distribution of mail by destinations. The possible number of "states" of the device is very large (for realistic values of r and k), and the creation of analytic techniques for the treatment of Markov chains with a great many states appears to be one of the "underdeveloped areas" of applied mathematics. We know that the specific Markov chain in the current problem has a probabilistic "steady state" toward which the device "settles down," and the *average* number $E(x)$ of letters in the device before dropouts, in this steady

state, might serve as a very rough initial guide⁶ to the capacity required of the device. Even the determination of $E(x)$ seems far from easy; we may note that the upper bound derived for $E(x)$ by a complicated probabilistic argument (see reference cited in footnote 4) yields the result $E(x) \leq 545$ for the particular illustrative data used, whereas our eq (4) (which was not aimed at estimating $E(x)$ and does not use the probabilistic distribution of mail by destinations) can be combined with the inequality

$$E(x) \leq x(t)_{\max \text{ poss}}$$

to yield the sharper result $E(x) \leq 401$ for the same data.

⁶ The frequency of large-amplitude oscillations around this average must also be considered; at the very least, some information is required on the standard deviation of the number of letters in the device.

WASHINGTON, January 5, 1959.