

Technical Report No. 144

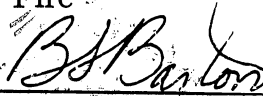
4967-2-T

SCHEDULING WITH RANDOM ARRIVALS  
AND LINEAR LOSS FUNCTIONS

by

D. W. Fife

Approved By:



B. F. Barton

COOLEY ELECTRONICS LABORATORY

Department of Electrical Engineering  
The University of Michigan  
Ann Arbor

Project 04967

Contract No. AF 30(602)-2661  
Rome Air Development Center  
Griffiss Air Force Base  
New York

April 1964

Eugen

UMR

1461

## TABLE OF CONTENTS

| Section |   | Page |
|---------|---|------|
|         | Summary . . . . .                               | 1    |
| 1       | Introduction . . . . .                          | 3    |
| 2       | Decision Problem With Random Arrivals . . . . . | 5    |
| 3       | An Equivalent Criterion . . . . .               | 7    |
| 4       | Optimum Nonpreemptive Scheduling . . . . .      | 11   |
| 5       | Concluding Remarks . . . . .                    | 15   |
|         | References . . . . .                            | 16   |
|         | Distribution List . . . . .                     | 17   |

## SUMMARY

The problem under consideration involves scheduling of the processing of an initial queue of jobs and subsequent Poisson arrivals on a single processor. Each job to be processed incurs a loss which increases linearly with its waiting time. The scheduling algorithm is sought which minimizes the average rate of expected loss over infinite time. It is shown that if statistical equilibrium exists for the total loss of an individual arrival, the optimal schedule minimizes the expected total loss of a single arrival, and is given by the scheduling rule which applies when there are no additional arrivals.



## 1. INTRODUCTION

A scheduling model which has attracted continued interest since its appearance is that introduced by McNaughton (Ref. 5). In this model, there is a finite set of jobs which must be processed in a facility composed of one or more identical processors. Each job can be finished in a single stage of processing by one processor, and the required processing time for job  $i$  is denoted by  $a_i$ . Assuming that the number of jobs in the set is larger than the number of processors, all jobs cannot be processed simultaneously and the elapsed time to completion for the jobs which must wait will be larger than the required processing time. It is assumed that if the time of completion of job  $i$  occurs after a deadline time,  $d_i$ , a loss is incurred for job  $i$  which depends upon the amount of delay beyond  $d_i$ . The loss generally reflects a penalty for failure to finish the job before the deadline, so it is assumed that no loss is incurred if completion occurs before  $d_i$ . The amount of loss for job  $i$ , if completion occurs at time  $t$  after the deadline, is given by a loss function  $L_i(t)$ , assumed to be monotonic nondecreasing.

A schedule for processing the set of jobs gives the order in which to admit jobs to the processing facility. The total loss of a schedule is given by the sum of the losses which result for individual jobs in the set. It is assumed that the processing time, deadline time, and loss function of any job are unaffected by their position in a schedule. The problem of interest here is to determine an optimum schedule which has a minimum value of total loss.

McNaughton (Ref. 5) and Smith (Ref. 8) have determined the optimum schedule in the special case of a single processor facility, with all jobs having linearly increasing, unbounded loss functions and deadlines which have passed when scheduling begins. The loss function is characterized simply by the loss rate, denoted  $p_i$  for job  $i$ . The optimum schedule is achieved by selecting jobs for processing in an order for which the corresponding sequence of values,

$\{p_i/a_i\}$  is monotonic nonincreasing. This procedure will henceforth be referred to as the single processor scheduling rule. Schild and Fredman (Refs. 6 and 7) have recently developed scheduling algorithms for the single processor facility when some jobs can be completed before their deadlines, or when loss functions are nonlinear.

The more general case of linear loss functions and a facility consisting of several identical processors in parallel is much more difficult. If all processing times  $a_i$  are equal and all  $d_i = 0$ , the scheduling rule given above for the single processor facility is sufficient to obtain an optimum schedule for the parallel facility. However, for unequal  $a_i$ , one can only show that every subset of jobs processed on the same processor in the optimum schedule must be ordered according to the single processor rule. No rule is known for optimally partitioning the set into subsets to be done on the same processor. Lacking an optimum algorithm, one is strongly tempted to use the single processor scheduling rule to obtain a schedule for the parallel facility. The resulting total loss appears to closely approximate the optimum when the maximum single job processing time is much less than the total time to complete all jobs. This conjecture has not been conclusively verified however. Partial results on the parallel processor problem have been given by McNaughton (Ref. 5), and recently by Lawler (Ref. 4).

The purpose of this paper is to extend the applicability of the single processor scheduling rule to a case in which additional jobs to be scheduled may arrive at random before a schedule is completed. Specifically, we consider Poisson arrivals of jobs to be processed on a single processor with the jobs having linearly increasing loss functions and deadlines occurring at the time of arrival. The total loss incurred in any time interval of operation is now a random variable, and is unbounded when the interval becomes arbitrarily large. Hence, the objective considered is to minimize the average rate of expected loss over an arbitrarily large time interval.

With nonpreemptive processing, in which processing of any job cannot be interrupted once it has begun, arrivals occurring while the processor is occupied cannot be scheduled until the job in process is completed. Thus the possibility exists that significant losses are incurred because the queue cannot be rescheduled when arrivals occur. Conceivably, a more complex scheduling procedure than the single processor scheduling rule might allow hedging against such losses. The scheduling decisions which must be considered in nonpreemptive processing are described in the next section. Then, it is shown that the optimum nonpreemptive scheduling policy gives a rule for selecting from the queue the next job to be processed when the processor is idle, and that the optimum rule is to select the job in queue having the largest value of the ratio of loss rate to processing time. Thus, the optimum rule for scheduling the single processor with Poisson arrivals is identical to that with no additional arrivals, under the constraint of nonpreemptive processing. The proof uses a general result from priority queueing theory, and is therefore interesting as a synthesis of this area with the rather distinct area of scheduling theory.

The case of preemptive-resume processing, in which processing of any job can be instantaneously interrupted and later resumed without increase of total processing time, cannot be treated by the method used here. But as a concluding remark, we express our conjecture as to the optimum scheduling rule which applies.

Our interest in this problem, as McNaughton's, stems from its application to the scheduling of a data processing machine.

## 2. DECISION PROBLEM WITH RANDOM ARRIVALS

First, we discuss the single processor scheduling decision problem in general, leading to a definition of the alternatives which must be considered in seeking the optimum scheduling rule.



The stochastic process of interest is  $\{C(T), 0 < T < \infty\}$ , the total loss incurred in service of independent Poisson arrivals in the time interval  $0 \leq t \leq T$  according to a specified scheduling algorithm. An optimal scheduling algorithm minimizes the asymptotic average rate of expected loss,

$$V = \lim_{T \rightarrow \infty} E(C(T)/T) \quad (1)$$

where  $E(\cdot)$  denotes the expectation over the ensemble of arrival sequences. As we will show, there is at least one policy for which  $V$  exists.

The underlying sample space of the stochastic arrival process in an ensemble of sequences  $\{(\tau_n, p_n, a_n), n = 1, 2, \dots\}$  where  $\tau_n$  is the time interval between the  $(n - 1)$ th and  $n$ th arrival. For  $n = 1$ , the term  $\tau_1$  is defined to be the time of the first arrival measured from  $t = 0$ . The terms  $\tau_n$ ,  $p_n$ , and  $a_n$  are independent, non-negative random variables for all  $N$ . The arrival process is assumed stationary, so the probability distribution functions of  $\tau_n$ ,  $p_n$ , and  $a_n$  do not change with  $n$ . The expectations are denoted  $E(\tau)$ ,  $E(p)$ , and  $E(a)$ , all assumed finite. In particular,  $E(\tau) = 1/\lambda$  where  $\lambda$  is the mean arrival rate.

We assume that  $p_n$  and  $a_n$  can have only a finite number of different values. Thus these random variables naturally divide the population of arrivals into a number  $R$  of what are called description classes; all jobs of a given description classes; all jobs of a given description class have the same values of loss rate  $p$  and processing time  $a$ . (All results here hold also if the number of classes is countably infinite, as when  $p_n$  and  $a_n$  assume only rational values.) Since we assume that all deadlines occur upon arrival, jobs of the same description class in queue are indistinguishable.

Because all jobs must eventually be processed, and because processing is assumed to be nonpreemptive, all scheduling decisions occur when the queue is not empty, and can be characterized as among the following:

- (a) Select any job of description class  $j$ , and begin processing it  
( $j = 1, 2, 3, \dots, R$ ).

- (b) Leave the processor idle until a specified time  $\tau$  has elapsed ( $0 < \tau < \infty$ ).
- (c) Leave the processor idle until the next arrival occurs.

Of course, one cannot decide to process a class  $j$  job if none are available at the decision time. This information is provided by knowledge of the state of the system, which at any decision time is given by a vector,  $s = (n_1, n_2, n_3, \dots, n_R)$  where  $n_j$  is the number of uncompleted class  $j$  jobs present ( $n_j \geq 0$  for all  $j$ ). A scheduling policy is then defined when the decision to be made from among the available alternatives is specified for every observable state and time at which a decision may occur.

We will limit consideration to stationary policies, for which the decision in any state is independent of the time at which the decision is made. Such a restriction is often desirable because of the difficulty of implementing a nonstationary policy. Furthermore, although we cannot show it, the optimal policy may indeed be stationary.

### 3. AN EQUIVALENT CRITERION

The criterion for optimality of a scheduling policy is that it minimizes  $V$  defined in (1). Whenever statistical equilibrium exists for the loss of an individual arrival as  $T \rightarrow \infty$ , another criterion is applicable. We now show that this equivalent criterion is to minimize the expected loss of a single arrival taken over the population of arrivals when equilibrium has been reached. Later results are obtained by considering only policies to which this equivalent criterion may be applied. In order to establish equivalence of the two criteria, the scheduling process must be treated with greater detail.

Let  $\xi$  denote the state of the system at  $t = 0$ , and assume the number of jobs in queue is finite, and all have finite loss rates and processing time values. For any arrival sequence, the integer random variable  $N(T, \xi)$  is defined to be the total number of jobs appearing for scheduling during the interval  $0 \leq t \leq T$

as random arrivals and as members of the initial queue, when the initial state is  $\xi$ . Clearly this variable is unaffected by the scheduling policy.

Let a particular scheduling policy be denoted by  $\pi$  and define the continuous random variable  $\ell_i(T, \pi, \xi)$  to be the loss over the interval  $0 \leq t \leq T$  of the  $i$ th job appearing for scheduling with any arrival sequence. Without loss of generality, the jobs initially in queue are considered as "arriving" at  $t = 0$  and indexed arbitrarily, with later arrivals subsequently numbered in order of time of arrival. Because the loss of any job during its processing time  $a_i$  is unaffected by scheduling, this constant loss term can be neglected in seeking the policy minimizing  $V$ . So, defining  $t_a^{(i)}(\xi)$  as the time of arrival of the  $i$ th job, and  $t_s^{(i)}(\pi, \xi)$  as the time of the start of its processing, we obtain

$$\ell_i(T, \pi, \xi) = p_i(\xi) \min \left\{ t_s^{(i)}(\pi, \xi) - t_a^{(i)}(\xi), \quad T - t_a^{(i)}(\xi) \right\} \quad (2)$$

where  $0 \leq t_a^{(i)}(\xi) \leq T$ . Neither the loss rate  $p_i(\xi)$  nor arrival time  $t_a^{(i)}(\xi)$  is dependent upon the policy  $\pi$ . Then,

$$C(T, \pi, \xi) = \sum_{i=1}^{N(T, \xi)} \ell_i(T, \pi, \xi) \quad (3)$$

is the total loss incurred in  $0 \leq t \leq T$  with policy  $\pi$  and initial state  $\xi$ .

For every interval  $T$ , policy  $\pi$ , and initial state  $\xi$ , there is assumed to exist a probability measure on values of  $N(T, \xi)$ , denoted  $P_n$ , and a conditional probability measure on values of the set  $\{\ell_i(T, \pi, \xi), i = 1, 2, \dots, N(T, \xi)\}$  for a specified value of  $N(T, \xi)$ . Let  $Q_n(x_1, x_2, \dots, x_n)$  denote the latter. Suppressing the arguments denoting conditionality on  $T, \pi, \xi$ , we can write

$$E\left(\frac{C}{T}\right) = \frac{1}{T} \sum_n P_n \int_{x_1} \cdots \int_{x_n} \sum_{i=1}^n x_i dQ_n(x_1, \dots, x_n).$$

By introducing a conditional expectation we can set

$$\int_{x_1} \cdots \int_{x_n} \sum_{i=1}^n x_i dQ_n(x_1, \dots, x_n) = \sum_{i=1}^n E(\ell_i | n)$$

where  $E(\ell_i | n)$  is the expected loss of job  $i$  over the interval  $[0, T]$ , given  $N(T, \xi) = n$ . Then,

$$E\left(\frac{C}{T}\right) = \frac{1}{T} \sum_n P_n \sum_{i=1}^n E(\ell_i | n). \quad (4)$$

As defined in (1),  $V$  is the limit as  $T \rightarrow \infty$  of the expression in Eq. 4. We assume that  $V$  exists and has the same value for every initial state  $\xi$ . Both  $P_n$  and  $E(\ell_i | n)$  are continuous functions of  $T$  for every value of  $n$ . So, assuming  $E(\ell_i | n)$  is bounded for all  $i, n, \xi$ , and  $T$ ,  $V$  can be expressed as

$$V = \sum_n \left[ \lim_{T \rightarrow \infty} \frac{P_n}{T} \right] \left[ \lim_{T \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n E(\ell_i | n) \right]. \quad (5)$$

From the stationarity of the Poisson arrival process and the restriction to finite initial states,  $N(T, \xi)/T \rightarrow \lambda$  with probability one as  $T \rightarrow \infty$ . The left limit in (5) is therefore nonzero only for  $n \rightarrow \infty$ . Thus

$$V = \lambda \lim_{n \rightarrow \infty} \lim_{T \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n E(\ell_i | n). \quad (6)$$

Now define  $c_i(\pi, \xi)$  as the total loss of job  $i$  as  $T \rightarrow \infty$ . Corresponding to (2) is the definition

$$c_i(\pi, \xi) = p_i(\xi) w_i(\pi, \xi) \quad (7)$$

where  $w_i(\pi, \xi)$  is the total waiting time of job  $i$  before start of processing. Then

$$\lim_{n \rightarrow \infty} \lim_{T \rightarrow \infty} E(\ell_i | n) = E(c_i)$$

and

$$V = \lambda \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n E(c_i). \quad (8)$$

Finally, assume that  $E(c_i)$  converges to a common value  $E(c)$  as  $i \rightarrow \infty$  for every initial state  $\xi$ . Then it follows that  $V = \lambda E(c)$  for every state  $\xi$ . To minimize  $V$ , one must minimize  $E(c)$ , which can be interpreted as the expected total loss of a single arrival as  $T \rightarrow \infty$ .

Summarizing, the expected loss  $E(c)$  will serve to evaluate any scheduling policy whenever:

- (a) The expected total loss of any job is bounded for every initial state.
- (b) The expected total loss of a single arrival at  $T \rightarrow \infty$  converges to a value independent of the initial state.

Both properties hold when all jobs are processed and a stationary probability distribution, independent of the initial state, exists for the total loss of a single arrival at  $T \rightarrow \infty$ . The latter describes the condition of statistical equilibrium, familiar in queueing theory.<sup>1</sup>

Thus, an example of a policy satisfying these conditions is the "first come, first served" queue discipline. In this case, condition (a) above is satisfied if  $\lambda E(a) < 1$ . Because the waiting time,  $w_i(\pi, \xi)$  is independent of  $p_i(\xi)$

$$E(c_i) = E(p_i)E(w_i).$$

For  $i$  sufficiently large that the  $i$ th job is a random arrival, rather than one of the initial queue members,  $E(p_i) = E(p)$ . Thus one has only to show convergence of  $E(w_i)$  for every initial state to demonstrate condition (b). A detailed proof of that has been given by Kiefer and Wolfowitz (Ref. 3).

Using subscripts to identify description classes, one can write, for any policy for which equilibrium exists,

---

<sup>1</sup>R. Syski, Introduction to Congestion Theory in Telephone Systems, Oliver and Boyd, London, 1960, p. 179.

$$E(c) = \sum_{j=1}^R p_j q_j E(w_j) \quad (9)$$

where  $q_j$  is the probability that an arrival is a class  $j$  job. Then it easily follows that

$$\sum_{j=1}^R p_j E(n_j) = \lambda E(c) + \text{constant} \quad (10)$$

where the constant is independent of the policy. The left side is the expected total loss rate in equilibrium, which is therefore minimized by the optimal policy.

#### 4. OPTIMUM NONPREEMPTIVE SCHEDULING

Confining consideration now to policies for which statistical equilibrium exists, we proceed to a specification of the optimal nonpreemptive scheduling policy. In doing so, we use some general results on nonpreemptive priority queues obtained by Cobham (Ref. 1) and Holley (Ref. 2).

First, it can easily be argued that no decision which leaves the processor idle when jobs are waiting to be processed is included in an optimal policy. Let  $E(w_j)$ ,  $j = 1, 2, \dots, R$ , in (9) denote the values obtained by a policy in which the processor is not left idle. A decision to leave the processor idle must reduce  $E(w_j)$  for some  $j$  if it is to be an optimal decision. We can consider  $w_j$  as composed of two parts:  $w'_j$ , the time a job waits after its arrival until any job then in process is completed; and  $w''_j$ , the subsequent time spent waiting for processing of other jobs in queue before its admission to the processing facility. If all jobs are to be eventually processed and equilibrium exists, any decision to leave the processor idle when the queue is not empty cannot change the expected fraction of time the processor is occupied processing jobs of any class. There-

fore,  $E(w'_j)$  is unaffected by these decisions for any  $j$ . In fact, following Cobham's discussion (Ref. 1), one can show that

$$E(w'_j) = E(T_0) = \frac{\lambda}{2} \sum_{j=1}^R a_j^2 q_j \quad (11)$$

for all  $j$ . On the otherhand,  $w''_j$  will be unchanged or increased for every  $j$  by a policy which allows the processor to be idle when jobs are waiting. Thus  $E(w_j)$  cannot be reduced for any  $j$  by such a policy.

The decisions to be determined are now seen to occur at the times of job completions where the queue is not empty, and constitute a selection of the class of the job to be processed next from the description classes represented in the queue. Because the optimum scheduling rule for the single processor without additional arrivals remains the same for any state, and because the random arrival process here is independent of the state of the system, it is sufficient to consider a selection criterion applicable to any state and achieved by assigning priorities to the description classes based on the loss rate, arrival rate, and processing time values of each. The next job to be processed is any job from the highest priority class present in queue, and without introducing any restriction we can assume the earliest arrival is chosen. Hence, we seek the rule for assigning nonpreemptive priorities which minimizes  $E(c)$ .

Let subscripts on  $p_i$ ,  $a_i$ , and  $q_i$  now refer to the values for the description class assigned to priority  $i$  in an  $R$  priority class system, where priorities are numbered  $i = 1, 2, \dots, R$  from highest to lowest. To evaluate  $E(c)$  for any priority assignment requires an expression for the expected waiting time of a priority  $i$  arrival in statistical equilibrium. From Cobham (Ref. 1) and Holley (Ref. 2), the required result, specialized to the case where the processing time for each priority class is a known constant, is

$$E(w_i) = \frac{E(T_o)}{\left(1 - \lambda \sum_{j=1}^{i-1} q_j a_j\right) \left(1 - \lambda \sum_{j=1}^i q_j a_j\right)} \quad (12)$$

$E(T_o)$  is defined in (11). With  $\rho_i = \lambda q_i a_i$ , the condition for equilibrium to exist is

$$\sum_{i=1}^R \rho_i < 1. \quad (13)$$

Then, for an arbitrary priority assignment,

$$E(c) = \sum_{i=1}^R \frac{p_i q_i E(T_o)}{\left(1 - \sum_{j=1}^{i-1} \rho_j\right) \left(1 - \sum_{j=1}^i \rho_j\right)} \quad (14)$$

The optimum priority assignment can be obtained from an arbitrary initial assignment by successively interchanging classes assigned to adjacent priority levels, each interchange reducing the loss measure  $E(c)$ . The conditions under which an interchange of any pair reduces loss can be determined by a comparison of  $E(c)$  with the loss measure  $E_m(c)$  of the assignment in which the two classes assigned to levels  $m, m + 1$  are interchanged ( $m = 1, 2, \dots, R - 1$ ). When the subscripts identifying classes by their priority level before the interchange, are maintained,

$$\begin{aligned} E(c) - E_m(c) &= \frac{p_m q_m E(T_o)}{\left(1 - \sum_{j=1}^{m-1} \rho_j\right) \left(1 - \sum_{j=1}^m \rho_j\right)} \\ &+ \frac{p_{m+1} q_{m+1} E(T_o)}{\left(1 - \sum_{j=1}^m \rho_j\right) \left(1 - \sum_{j=1}^{m+1} \rho_j\right)} - \frac{p_{m+1} q_{m+1} E(T_o)}{\left(1 - \sum_{j=1}^{m-1} \rho_j\right) \left(1 - \sum_{j=1}^{m-1} \rho_j - \rho_{m+1}\right)} \end{aligned}$$



$$-\frac{p_m q_m E(T_0)}{\left(1 - \sum_{j=1}^{m-1} \rho_j - \rho_{m+1}\right) \left(1 - \sum_{j=1}^{m+1} \rho_j\right)} \quad (15)$$

From (15) one can show  $E_m(c) < E(c)$  if

$$\frac{1 - \sum_{j=1}^{m-1} \rho_j - \rho_m}{1 - \sum_{j=1}^{m-1} \rho_j - \rho_{m+1}} < \frac{1 - \sum_{j=1}^{m-1} \rho_j - \frac{p_m q_m}{\bar{p}} (\rho_m + \rho_{m+1})}{1 - \sum_{j=1}^{m-1} \rho_j - \frac{p_{m+1} q_{m+1}}{\bar{p}} (\rho_m + \rho_{m+1})} \quad (16)$$

where  $\bar{p} = p_m q_m + p_{m+1} q_{m+1}$ . Then

$$\frac{p_m q_m}{\bar{p}} (\rho_m + \rho_{m+1}) < \rho_m$$

holds whenever

$$\frac{p_m}{a_m} < \frac{p_{m+1}}{a_{m+1}} \quad (17)$$

But, (17) is also required in order for

$$\frac{p_{m+1} q_{m+1}}{\bar{p}} (\rho_m + \rho_{m+1}) > \rho_{m+1}$$

Hence, (17) is both necessary and sufficient that (16) be true, and that the interchange of any two adjacent priority assignments reduces loss.

If the inequality in (17) becomes an equality, the interchange of priority assignments does not affect  $E(c)$  and hence the two description classes can be assigned to one priority. By a succession of interchanges and combinations of classes having equal ratios,  $p/a$ , one arrives at a priority system of some number  $K$  of levels, with

$$\left(\frac{p}{a}\right)_1 > \left(\frac{p}{a}\right)_2 > \dots > \left(\frac{p}{a}\right)_K.$$

The optimum schedule for a single processor without additional arrivals or with Poisson arrivals is therefore obtained by the same scheduling rule.

## 5. CONCLUDING REMARKS

Cobham's result, from which (12) is obtained, holds also in the more general case where the processing time of any job in the description class assigned to priority  $i$  is not known with certainty, but has a probability distribution function  $F_i(\tau)$ , with expected value  $\mu_i$ . In this case,

$$E(T_0) = \frac{\lambda}{2} \sum_{j=1}^R q_j \int_0^{\infty} \tau^2 dF_j(\tau)$$

and  $E(w_1)$  can be obtained by replacing  $a_j$  in (12) by  $\mu_j$ , for every value of  $j$ . It follows from the preceding discussion that the optimal scheduling policy is to assign description classes to priority levels so that

$$\frac{p_1}{\mu_1} > \frac{p_2}{\mu_2} > \dots > \frac{p_R}{\mu_R}$$

The expected processing time is the only parameter of the processing time probability distribution that need be known to achieve optimal scheduling.

The constraint of nonpreemptive service is rather severe in terms of the resulting average expected loss rate. In some applications, it may be possible to preempt any job in process and begin processing of another with negligible delay and without losing any of the processing time accomplished on the preempted job. Intuitively, it seems that since random arrivals did not introduce any modification of the single processor scheduling rule with nonpreemptive processing, the optimal preemptive-resume policy would require that at any time the queue was not empty the job which has the largest ratio of loss rate

to remaining processing time should be the one in service. It does not appear feasible to establish this rigorously by the method employed here, for it requires that the priority of a job be continuously changing while it is in process. A priority queueing model with such generality would be very difficult to solve.

## REFERENCES

1. A. Cobham, "Priority Assignment in Waiting Line Problems," Operations Research, 2, 70-76, 1954.
2. J. Holley, "Waiting Line Subject to Priorities," Operations Research, 2, 341-343, 1954.
3. J. Kiefer and J. Wolfowitz, "On the Characteristics of the General Queueing Process with Applications to Random Walk," Annals of Mathematical Statistics, 27, 147-161, 1956.
4. E. Lawler, "Computational Techniques for Scheduling Problems with Deferral Costs," Cooley Electronics Laboratory TR-140, The University of Michigan, Ann Arbor, Michigan, July, 1963. (Also presented at the National meeting of the Association for Computing Machinery, Denver, Colorado, August 27-30, 1963).
5. R. McNaughton, "Scheduling with Deadlines and Loss Functions," Management Science, 6, 1-12, 1959.
6. A. Schild and I. Fredman, "On Scheduling Tasks with Associated Linear Loss Functions," Management Science, 7, 280-285, 1961.
7. A. Schild and I. Fredman, "Scheduling Tasks with Deadlines and Non-linear Loss Functions," Management Science, 9, 1962.
8. W. E. Smith, "Various Optimizers for Single Stage Production," Naval Research Logistics Quarterly, 3, 59-66, 1956.

DISTRIBUTION LIST

- |    |  |     |
|----|--|-----|
| 1. | John McLean, EMIIF<br>RADC<br>Griffiss AFB<br>Rome, New York               | (1) |
| 2. | Lt. L. W. Odell, EMIICA<br>RADC<br>Griffiss AFB<br>Rome, New York          | (2) |
| 3. | Pat Langendorf, EMIIT<br>RADC<br>Griffiss AFB<br>Rome, New York            | (1) |
| 4. | Fred Dion, EMIIT<br>RADC<br>Griffiss AFB<br>Rome, New York                 | (1) |
| 5. | Rome Air Development Center<br>Griffiss AFB<br>Rome, New York<br>Attn: RAT | (1) |
|    | RAALD  | (1) |
|    | RAAPT  | (1) |
|    | RAL  | (1) |
|    | RALC   | (1) |
|    | RALS   | (1) |
|    | RALT   | (1) |
|    | RAO  | (1) |
|    | RAD  | (1) |
|    | RAS  | (1) |
|    | RASG   | (1) |
|    | ROAMA (ROAEPP-1)   | (4) |
|    | EMIIH  | (5) |
|    | EMIIT  | (5) |
|    | EMIIF  | (5) |
|    | EMIID  | (5) |
|    | RASH   | (1) |
|    | RASS   | (1) |
|    | RAU  | (1) |
|    | RAUA   | (1) |
|    | RAUE   | (1) |
|    | RAUM   | (1) |
|    | RAUO   | (1) |
|    | RAWC   | (1) |
|    | RAWE   | (1) |
|    | RAWI   | (1) |
|    | RAYA   | (1) |
|    | RAI  | (1) |

DISTRIBUTION LIST (Continued)

- |     |   |                                 |
|-----|---|---------------------------------|
| 6.  | GEEIA<br>Griffiss AFB<br>Rome, New York<br>Attn: ROZMA<br>ROZMC<br>ROZME<br>ROZMN<br>ROZMCAT                                    | (1)<br>(1)<br>(1)<br>(1)<br>(1) |
| 7.  | RAOL (Maj. Shields)<br>Griffiss AFB<br>Rome, New York   | (1)                             |
| 8.  | RAOL (S/ L Tanner)<br>Griffiss AFB<br>Rome, New York  | (2)                             |
| 9.  | AFSC<br>Andrews AFB<br>Washington 25, D. C.<br>Attn: SCSE<br>SCLDE<br>SCFRE   | (1)<br>(1)<br>(1)               |
| 10. | Redstone Scientific Information Center<br>U. S. Army Missile Command<br>Redstone Arsenal, Ala.<br>Attn: Chief, Document Section | (2)                             |
| 11. | Bureau of Naval Weapons<br>Main Navy Building<br>Washington 25, D. C.<br>Attn: Tech. Lib. DL1-3                                 | (2)                             |
| 12. | Chief, Bureau of Ships<br>Code 454 F<br>Main Navy Building<br>Washington 25, D. C.  | (1)                             |
| 13. | Central Intelligence Agency<br>2430 E Street NW<br>Washington 25, D. C.<br>Attn: OCR Mail Room                                  | (1)                             |
| 14. | U. S. Army Material Command<br>Harry Diamond Laboratories<br>Washington 25, D. C.<br>Attn: AMXDO-TIB                            | (1)                             |
| 15. | Scientific and Technical Information Facility<br>P. O. Box 5700<br>Bethesda, Maryland<br>Attn: NASA Representative (S-AK/DL)    | (2)                             |

DISTRIBUTION LIST (Continued)

16. Director (1)  
National Security Agency  
Ft. George G. Meade, Maryland  
Attn: C3/TDL
17. Commander (1)  
Naval Missile Center  
Tech Library (Code No. 3022)  
Pt. Mugu, California
18. Commanding Officer and Director (1)  
U. S. Navy Electronic Lab (Lib)  
San Diego 52, California  
92152
19. Office of Chief of Naval Operations (Op-723 E) (1)  
Rm. 1602  
Building T-3, Navy Department  
Washington 25, D. C.
20. Office of Naval Research (1)  
Chief Scientist (Code 427)  
Washington 25, D. C.
21. The Rand Corp. (Tech Lib) (1)  
1700 Main Street  
Santa Monica, California
22. Hq TAC (DORQ-S) (1)  
Langley AFB  
Virginia
23. Hq TAC (OA) (1)  
Langley AFB  
Virginia
24. Commander, TAC Comm Region (1)  
Langley AFB  
Virginia
25. USA FSS (ECD) (1)  
San Antonio  
Texas
26. Commanding General (2)  
US Army Electronic Proving Ground  
Attn: Tech Library  
Fort Huachuca, Arizona
27. Commanding Officer (1)  
US Army Electronics Rand D Lab  
Attn: SELRA/ADT  
Ft. Monmouth, New Jersey

DISTRIBUTION LIST (Continued)

- |     |  |                   |
|-----|--|-------------------|
| 28. | Institute of Technology Library<br>MCLI-LIB, Bldg 125 Area "B"<br>Wright-Patterson AFB<br>Ohio                             | (1)               |
| 29. | NAFEC Library<br>Bldg. 3<br>Atlantic City, New Jersey  | (1)               |
| 30. | Commandant<br>US Army War College (Lib)<br>Carlisle Barracks, Pa.  | (1)               |
| 31. | Commanding Officer<br>US Army Electronic Research Unit<br>P O Box 205<br>Mountain View, California                         | (1)               |
| 32. | Electronic Defense Labs<br>Attn: Library<br>P O Box 205<br>Mountain View, California                                       | (1)               |
| 33. | U S Naval Avionics Facility (Library)<br>Indianapolis 18<br>Indiana  | (1)               |
| 34. | Commander<br>US Naval Air Dev Cen (NADC Lib)<br>Johnsville, Pa.  | (1)               |
| 35. | Director<br>US Naval Research Lab (Code 2027)<br>Washington 25, D. C.  | (2)               |
| 36. | Hq USAF<br>Attn: AFRSTE<br>AFRDPC<br>AFGOA<br>Washington 25, D. C.   | (1)<br>(2)<br>(1) |
| 37. | National Aeronautics and Space Admin<br>Langley Research Center<br>Langley Station<br>Hampton, Virginia<br>Attn: Librarian | (3)               |
| 38. | RTD (RTH)<br>Bolling AFB 25 D. C.  | (1)               |
| 39. | Federal Aviation Agency<br>Information Retrieval Br Hq-630<br>Washington 25, D. C.   | (1)               |

DISTRIBUTION LIST (Continued)

- 40. RTD (RTHR/ Capt. K. O. Malkemes (2)  
Bolling AFB, D. C.  
20332
- 41. OSD (DDR and E/ Mr. Willie Moore (2)  
Washington D C  
20330
- 42. Hq USAF (AFRST/ Lt. Col. A. H. Grinsted, Jr.) (1)  
Washington D C  
20330
- 43. Advanced Research Projects Agency (1)  
Command and Control Research  
(Dr. J. C. R. Licklider)  
Washington, D. C.
- 44. National Security Agency (1)  
Engineering Research Div, Code R42  
(Mr. O. H. Bartlett, Jr.)  
Fort Meade, Maryland



UNIVERSITY OF MICHIGAN



3 9015 02826 3666