4967-3-T

Technical Report No. 145

ECONOMIC CONTROL AND OPTIMIZATION OF A MULTICONSOLE SHARED DATA PROCESSING SYSTEM

by

V. L. Wallace

Approved by: B F Barton

B. F. Barton

for

COOLEY ELECTRONICS LABORATORY

Department of Electrical Engineering
The University of Michigan
Ann Arbor

Air Force Systems Command
Rome Air Development Center
AF 30(602)-2661
Griffis Air Force Base, New York

April 1964

ACKNOWLEDGMENT

The author wishes to acknowledge the valuable aid and criticism provided by Mr. D. W. Fife during the course of this research.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

SUMMARY

In shared, multiple-console data processing systems, individual determination of optimal arrival times and priorities for each job by a central authority is clearly impossible. This report shows that a "free market" type of decentralized control, with certain traffic information displayed to users continuously and with prices for service controlled in a particular manner, results in an optimum selection of arrival time and priority by the users themselves.

The report develops an economic model for the description of the word "optimum" in this context, and determines the optimum policy through theoretical investigation. The optimum policy is then shown to result from the "free market" control.

1. GENERAL

A multiple-console data processing system which is shared by many users has much in common with a public utility. People who have computing to be done simply seek out an available console and obtain service with almost no knowledge of competing demands by other people. The advantages of such service have been enumerated many times (Refs. 1-4), but the one essential feature which such a system has in common with public utilities is that although demands appear in an apparently random manner, the statistical averaging of large numbers of jobs permits the system to give economical service on short notice.

When it comes to the question of what constitutes acceptable service, however, the analogy to typical utility systems breaks down. Unlike telephone systems, the value of fast service on a shared computer varies very widely from user to user and job to job. Sometimes milliseconds are important; on other occasions and for other users, delays of an hour or more may be truly minor.

This, implicitly, is the reason why most scheduling algorithms suggested for shared systems (Refs. 4-6) allow for multiple priority levels for the arriving jobs. Nevertheless, the criticality of priority depends directly on the amount of pressure which exists forcing maximum use of the computer resource. The problem is avoided only if either the utilization by urgent jobs is low relative to total demand allowing almost instantaneous service, or if the optimization of service on a particular machine is unimportant, as when many other alternatives are available to the users.

On the other hand, the potential number of jobs entering the system is tremendous. The shared system of Corbato at MIT using the IBM 7090, no multiprocessing,[1] and rather slow input-output typewriters encounters jobs averaging perhaps a few seconds in duration. It can be expected that with growing user experience in using such systems, more developed system routines and consoles, and by multiprocessing, an arrival rate of several hundred

[1] Multiprocessing is defined here as the existence and activity of more than one job in core memory simultaneously.

jobs per minute is possible, even for a 7090. The individual determination of priorities for so many jobs by a central authority is clearly impossible.

Thus it is seen that not only must the priorities be established, they must also be either established automatically or by placing authority for the selection of priorities in the hands of each individual user. In an open environment automatic priority assignment is rarely practical because the exception is usually the rule. Hence, the individual user is the one in the best position, with suitable supervision, to evaluate his priority requirements.

The obvious danger in decentralizing priority assignments is possible abuse by the user. It is the purpose of this report to provide a procedure for making every incentive toward improvement of _personal_ service an incentive toward the maximization of value _to an entire enterprise_ by appropriate manipulation of the charges for the service. There are clearly three major matters which must be studied in order to accomplish this purpose: (1) the nature of the economic quantity loosely referred to as "value," (2) the optimum schedule toward which the normal individual incentives are to be directed, and (3) the practical mechanism for manipulation of the incentives. Each of these matters will be the subject of a chapter in this report.

The fundamental control concept of this work, the idea of using prices for service to manipulate incentives of users toward maximizing value to an enterprise, presupposes that the prices themselves are not necessary to other functions. While this will be true in general for a computing system which is a service operation for a single economic unit (enterprise) as in the military or large corporations having sole use, it will not be the case for a service bureau computer providing service at a profit. In the latter case, the prices represent a component of profit and are themselves a part of the quantity to be maximized, while in the former case the prices are merely a means of transfering funds from one pocket to another and can be compensated for collectively by budgets. It is here that such control can be used to great advantage in the control of user decisions, and to which purpose this work is directed.

It should, however, be made clear at the outset that this work is not based upon the fallacious concept of "rational man" in the person of the user. This image of man as a being who always makes the choice which is best for him based on complete information concerning his environment has been thoroughly refuted in economic and philosophical literature.

Rather, man is here assumed to be governed by a process of "bounded rationality," to borrow a term from Herbert Simon (Ref. 7). Such a man bases his decisions upon limited information he can gather in a reasonable period of time, using educated guesses to fill in missing detail,[1] and seeking primarily a choice which is "good enough" rather than "best." This model, which is much more acceptable, is useful in describing behavior when the person is sufficiently motivated. At the very least it should describe the tendency or statistical average of a large number of operator decisions in the shared computer system.

Finally, let it be emphasized that this report represents but one part of a very large and important problem: that of establishing a complete and usable supervisory control for the shared computer which allows maximum flexibility and utility. There are many system organization problems which are not discussed here which have larger effect on the value of the system. For example: memory and I/0 organization and protection, interrupt scheduling, file cleanup, user commands and system subroutines available, and console organization.[2] No amount of schedule control and optimization can really make up for poor choice in the supervisory program or hardware. Optimization as discussed here can, however, improve that which is already good.

[1] It might be argued that this is an example of rationality when account is taken of the cost of obtaining the information.

[2] For an interesting solution to these problems, see the report by Corbato, et al., Ref. 2.

2. AN ECONOMIC MODEL

The data processing system, as represented by the equipment and its executive system, is embedded in a larger system which includes programmers, technicians, and administrators. This latter system, which will be called the enterprise, has a broad set of goals and objectives. If we consider only those goals and objectives which are forwarded through the use of the data processing system, it is reasonable to assume that these goals are directly related to the completion of computations (jobs) prepared and requested by the enterprise and executed on the system.

Since it is impossible to achieve all goals simultaneously, it is necessary to compromise some for the advantage of others. The choice of the best possible compromise is the central problem of system control. In order to make the choice, an understanding of the economic relationships is obviously necessary and this chapter is devoted to their description through an economic model.

2.1 Utility of Completed Computation

To begin with, we consider each result of computation (output) to have some utility to the enterprise which can be measured in some standard unit, say dollars. This is not a trivial assumption, albeit quite tempting. Normally, economists refer to the objectives of an enterprise as composed of many separate, noninterchangeable factors. (For example, see Ref. 8.) To use a single unit of utilities implies at least that an ordering of preference of each possible outcome is possible; that if a is preferred to b, and b is preferred to c, then a is necessarily preferred to c. In addition it implies that one can in fact infer relative "distances" between a and b, and b and c.

That this is not unreasonable has been shown by von Neumann and Morgenstern (Ref. 9) using the ideas of probability and indifference curve analysis. In any event, we shall follow our intuition and assume that if sufficient intangible factors are included in the utility, it can be measured in a single unit.

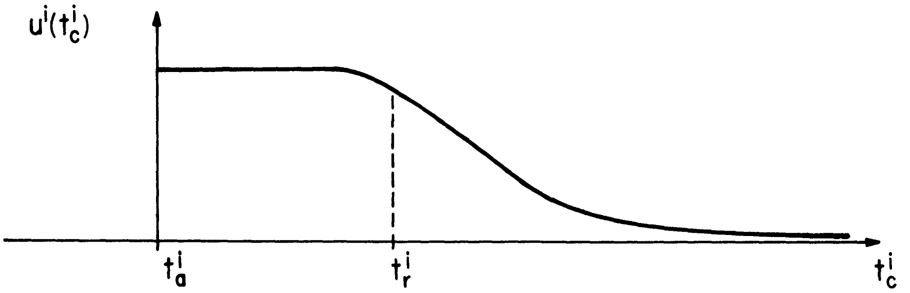

Fig. 1. Utility of a typical problem as a function of completion time.

The utility of each computed result will, in general, depend upon the time at which the result is completed. Thus, if we denote $u = u^i(t_c^i)$ as the utility of the results of job i which is completed at time t_c^i, then a typical utility function u^i would probably be of the form shown in Fig. 1. That is, the utility would generally be decreasing with time, be quite uncritical of time before some relative deadline t_r^i, and then would fall to zero as the delay gets indefinitely large.

The utility might be dependent upon other controllable factors which affect the quality of results, such as the amount of data processing equipment which can be assigned to the job during its execution. However, these factors are ordinarily less significant than the arrival or completion time, and will be ignored in this model except where otherwise stated. [1]

On the other hand, the factors influencing the utility which are not controllable are part of the process which determines the functional form of u^i and hence they also need not be shown explicitly as variables of the function.

2.2 Direct Cost of Program Preparation

Also associated with each job will be some direct costs which represent a loss of utility to the enterprise due to operations which are individually chargeable to the problem. Chief of these are the "loaded salaries" of programmers and analysts during the time they are working on the preparation of the program and data for insertion into the computing system. All costs which would not have occurred had the problem not existed, should be represented in this direct cost. On the other hand, costs such as the (overhead) cost of operating

[1] In particular, the controllable assignment of equipment is likely to have its greatest influence on the service time rather than the utility, and is best represented in that (former) part of the economic model.

the data processing system itself should not be included, since presumably the system would be running even if the particular problem did not require its use.[1] In addition, there are intangible costs to the enterprise which represent such things as inconvenience by the price which the enterprise would be willing to pay to avoid them.[2]

The direct cost can be represented as a function of completion time and the arrival time, t_a^i, of the user to the console.[3] Alternatively, it can be represented as a function of the arrival time and the waiting time t_w^i $(t_w^i = t_c^i - t_a^i)$. Denote $c = c^i(t_w^i, t_a^i)$ as the direct cost of job i. Then this function can take many forms depending on environmental factors.

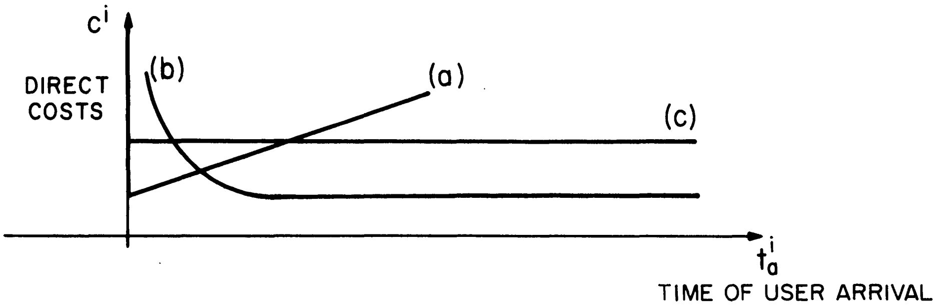

Fig. 2. Some typical direct costs of program preparation.

For example, if a user must wait idly for an appointed arrival time before going to a console with his job, then the longer he must wait the higher will be the direct cost. On the other hand, if the time of his arrival is known in advance and programming is scheduled to meet that time as a deadline, then higher costs will be associated with shorter completion times due to the less efficient "crash" program which is required. The two corresponding cost functions with t_w^i held fixed are represented in Fig. 2(a) and (b), respectively. Also shown is the more neutral situation where the programming is already completed before arrival is considered [Fig. 2(c)] . Of course, the number of possible relations is very great,

[1] The influence of operating costs and overhead will be considered in Section 2. 4.

[2] For example, the cost of inefficiency resulting from "frayed nerves" due to bad scheduling by the computer or the possible loss of an employee because of excessive night-time working hours are legitimate parts of the direct costs of a job.

[3] More precisely, we define the time t_a^i as the time at which the user is first present at the console with a job complete and ready for processing.

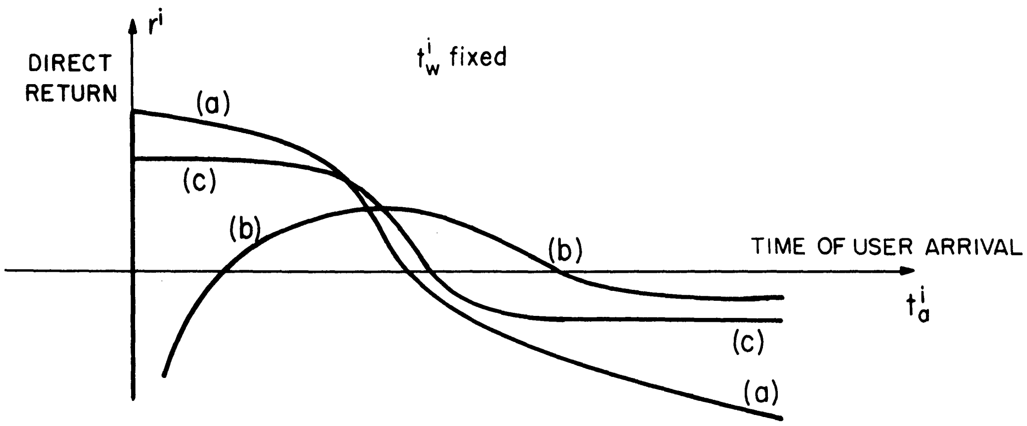

Fig. 3. Some typical direct returns.

and increased still further by a like variety of dependences upon the waiting time as well. Nevertheless, for each job some such relation exists. Moreover, it is assumed as in the previous section that controllable influencing factors other than t_w^i and t_a^i are neglected unless otherwise stated and that noncontrollable factors are implicit in the form of the function c^i.

2.3 Direct Return

The completion of a particular job at a particular time results, therefore, in a net gain in utility which is the consequence of the utility of the computed result $u^i(t_c^i)$, where $t_c^i = (t_w^i + t_a^i)$, less the direct costs $c^i(t_w^i, t_a^i)$ of getting the result. We call this net gain in utility the direct return on the processing, and denote it by the function

$$r = r^i(t_w^i, t_a^i) = u^i(t_w^i + t_a^i) - c^i(t_w^i, t_a^i) . \qquad (1)$$

Typical direct returns which may result from the utility and direct costs of Figs. 1 and 2 when t_w^i is fixed are represented in Fig. 3. It should be noted that negative returns represent a net loss to the enterprise due to the existence of the ith job. Nevertheless, once the enterprise is committed to the job it is often still more costly to postpone the execution indefinitely (i. e., let $t_a^i \to \infty$ in Fig. 3 or let $t_w^i \to \infty$). Thus a job yielding negative returns may not be excluded from consideration in an optimization of the system.

2.4 Total Return

The discussion now turns to the measure of achievement for the enterprise resulting from the entire data processing operation. This measure will be called the total return, R. We let R be the sum of the direct returns of all jobs completed in an arbitrary time interval, say $[t_1,t_2]$, less any fixed or "overhead" costs due to operating the system which are not chargeable to any individual job. Thus

$$R = R(t_1,t_2) = \sum_{\substack{\text{all i:} \\ t_c^i \epsilon [t_1,t_2]}} r^i(t_w^i, t_a^i) - C_o(t_1,t_2) , \tag{2}$$

where $C_o(t_1,t_2)$ is the overhead cost during the time interval $[t_1,t_2]$.

While $R(t_1,t_2)$ may, of course, be negative for some $[t_1,t_2]$, a minimal objective for the system is that for (t_2-t_1) sufficiently large, $R(t_1,t_2)$ should be positive. Any other condition would imply a long term loss from operating the system and would imply that the system should be abandoned (or should never be set up if this determination is made in advance of installation).

The general objective of executive organization is to maximize the total return $R(t_1,t_2)$ over a sufficiently long time interval. If, as will usually be the case, some of the variables of Eq. 2 are stochastic, then the objective will be to maximize the mathematical expectation of $R(t_1,t_2)$, for (t_2-t_1) sufficiently large. This point will be developed later.

There are two types of controls which influence the total return: structural and executive. Structural controls are represented by those decisions which determine the nature of the equipment complement (such as the number of consoles, tapes, and arithmetic units, the quantity of high speed storage, etc.) represented in the machine itself. These decisions are usually made on time scales of many months; sometimes (though rarely) only once for the life of the system.

Executive controls are represented by queueing disciplines, communication links either within the machine or among the users, and administrative rules or procedures. They generally influence short term decisions of the users and are often programmed into the executive system of the machine.

For every structure (i. e., choice of structural controls) there will be an optimum set of executive controls. Though this optimum probably will not be known, some approximation to it must be made and decisions based upon it.

Moreover, structure will be determined by calculating the total return for each structure to be considered, using the approximately optimal executive control corresponding to the structure. The structure resulting in maximum return is the optimal structure. On the other hand, since executive controls are dynamic, their optimization is much more subtle and complex. This report is primarily devoted to describing and approximating the optimum executive control.

2.5 Total Direct Return

It is noted that in Eq. 2 the overhead cost, $C_o(t_1, t_2)$ is dependent only on the structure of the machine. Since the optimal choice of executive control for a given structure is made by maximizing the total return (Eq. 2), it can also be made by maximizing the sum of direct returns. Thus, the maximum of

$$R_d = R_d(t_1, t_2) = \sum_{\substack{\text{all } i: \\ t_c^i \epsilon [t_1, t_2]}} r^i(t_w^i, t_a^i) \tag{3}$$

also results in the optimum executive control. We shall call R_d the <u>total direct return</u> of the enterprise.

Because the objective of the remainder of this report is directed toward describing and approximating executive control only, keeping structure invariant, the overhead cost and Eq. 2 will not be required. Equation 3 represents the quantity to be maximized.

2.6 Constraints and Optimization

The executive control can obviously not be arbitrarily chosen. Two or more jobs cannot be assigned simultaneously to the same equipment module. The nature of this constraint is mathematically described as follows:

We let $\underline{S} = (s_1, \ldots, s_m)$, a vector, represent the <u>structure</u> of the machine, so that its elements s_j are the number of interchangeable modules of type j. For example, the number of interchangeable blocks of storage, tape units, or arithmetic units might be s_1, s_2, and s_3, respectively. Then if we let $\underline{\gamma}^i(t-t_c^i)$ be a vector representing the number of each such module utilized by the ith job at time t if it completes its execution at time t_c^i, the constraint can be represented by

$$\sum_{\text{all } i} \underline{\gamma}^i[t - t_c^i] \leq \underline{S}, \qquad \text{all } t. \tag{4}$$

To summarize, then, the optimization of executive control is accomplished by choosing the set $\{t_w^i\}_i$ of waiting times and the set $\{t_a^i\}_i$ of arrival times for all jobs i, under the constraint (Eq. 4), so that the mathematical expectation of total direct return between two widely separated times, t_1 and t_2,

$$R_d = \sum_{\substack{\text{all } i: \\ t_c^i \in [t_1, t_2]}} r^i(t_w^i, t_a^i) \tag{5}$$

is a maximum.[1]

2.7 The Nature of the Variables

So far, nothing has been said of the properties of the variables and functions found in Eqs. 4 and 5. It has been inferred only that, in fact, counterparts to these variables and functions exist in the real world. As a matter of fact, this omission has been deliberate because under different circumstances very different assumptions concerning their nature apply. Succeeding chapters will make use of these differing assumptions and the useful approximations which result.

It suffices to remark here that (1) the direct returns $r^i(t)$ will normally be either known deterministic functions (e.g., if the ith problem is already in the queue) or a representation of a stochastic (random) process (e.g., if the job is to arrive in the future); (2) the service times t_s^i, the arrival times t_a^i, and the waiting times t_w^i, are similarly treated as either known constants or random variables (which may or may not be correlated to the direct return); and (3) the utilization vectors $\underline{\gamma}^i(t)$ may be a set of known vector integer functions or a vector integer stochastic process.

2.8 Concluding Remarks

This chapter has set forth an economic model for the desired goals of system

[1] By choosing to represent the utility u^i as $-\infty$ for all $t_w^i < 0$, we can avoid the necessity of representing nonexecution before arrival as a constraint. It becomes the natural consequence of the maximization of Eq. 5.

operation and of executive system design. While it is not, by any means, the only model which can be obtained, it appears to give a quite accurate representation of the relationships between utilities and costs, and the quantities normally controlled in such systems.

On the other hand, it is so general that, in practice, little success can be expected from a full scale analysis of it in all its generality. Indeed, similar models such as the classic job-shop problem (Ref. 10), have never been solved generally. Nevertheless, we shall proceed in the next chapter with a simplification of the economic model which retains most of its desirable features and is much more nearly solvable. This work will make use of scheduling theory and the so-called deferral-cost scheduling model.

Finally, the fourth and last chapter will be concerned with several schemes for executive control which approximate, with limited data, the optimum implied by the economic model and Eqs. 4 and 5.

3. THE OPTIMUM STRATEGY

As stated in Chapter 1, the objective of this report is to provide a procedure for making every incentive of users toward improvement of personal service an incentive toward the maximization of value to the entire enterprise by manipulating the charges for service. Chapter 2 has discussed the values to be maximized and the decisions which must be controlled. It is now important to decide how the decisions should be made in order to achieve this maximum.

In this chapter, the structure of the optimization will be developed, and the optimal policies set forth. The optimization will be based upon the apparently unrealistic assumption that the true return functions are known exactly, and known in advance of any decision affected by them. It will be shown in Chapter 4 that this assumption is not unrealistic, by showing that the essential information can be obtained (to the best extent possible) inductively by the prices and service accepted by the user.

On the other hand, it is important to realize that at least one piece of information cannot be precisely known in advance, and that is the actual arrival time to the console (as defined in Footnote 3, p. 6) by each user. This fact is essential to understanding the rationale behind the division of the basic problem into the two interrelated problems solved in this chapter.

In the first place, with a knowledge of the entire return function of each job, it would be formally possible to find the exact arrival times for each job which would result in true optimal return for all jobs, and which would actually involve no waiting time at all. In other words, by arriving at the precise time that the system became available, greater utility would be obtained than by any decision resulting in a wait at the console. However, because the times of arrival cannot be known or specified exactly, such a direct optimization cannot be achieved. Instead, a more complex problem must be solved.

We could go into many philosophical reasons why the exact optimum arrival times for each job cannot be pre-specified by the system. It suffices to say that normal human beings do not desire such precision in their habits, lack the physical capacity for timing within arbitrarily small limits (like the microseconds involved in scheduling computers), and are affected by value systems external to that specified in this model, which result in unexpected small re-evaluations of utilities. The optimization can only expect to specify "intended" arrival times which are within some random "error" of the true arrival.

Moreover, decisions concerning the scheduling of jobs which have already arrived and are awaiting service are necessitated. Thus, there are two decisions which must be made for each job i: (1) the "intended" arrival time, $\hat{t}_a^i$, must be chosen, and (2) the waiting time,[1] t_w^i, must be specified by means of a suitable queue rule. We now recast the problem given in Chapter 2 into these terms.

In the following section, the problem as stated in Chapter 2 will be recast into these terms. This will result in splitting the optimization problem into two related optimization problems; one associated with the choice of suitable queue rules, and the other with the intended arrival times.

While the former subproblem is difficult and cannot be solved exactly, many theoretical results are available which contribute to strong arguments for accepting as optimal the policy of sequencing jobs in decreasing order of the ratio p_i/t_s^i, where p_i is the rate of loss in utility with respect to waiting time, and t_s^i is the service time.

The latter subproblem is solved through development of an expression for each job which should be maximized by the chosen intended arrival time for each user.

3.1 The Structure of the Decision Process

The direct return of each job is a function of its actual arrival time and its waiting time. The most reasonable form to assume for this return is that it is the sum of a function of arrival time and a function of waiting time

$$r^i(t_w^i, t_a^i) = g_i(t_a^i) + f_i(t_w^i) . \tag{6}$$

This assumes that the loss in return due to waiting is the same regardless of the time of arrival, and vice versa.

Waiting time, t_w^i, will range over considerably smaller values (on the order of

[1] Note that decisions concerning priorities, control rules, or anything else related to scheduling of jobs already waiting can be represented by a set of selected waiting times that would result.

seconds or perhaps minutes) than will t_a^i (generally hours). But since g_i can be expected to be a much more slowly varying function than f_i, both factors, f_i and g_i, are significant over the range of t_a, t_w representing optimization. Assuming that the loss due to waiting is a linear function of t_w^i, we can write

$$r_i(t_w^i, t_a^i) = g_i(t_a^i) - p_i t_w^i \,, \tag{7}$$

where we refer to p_i as the "deferral loss rate" of job i, a constant for each job.

It has been said (Eq. 5) that the total direct return is

$$R_d = \sum_{\substack{\text{all } i \\ t_c^i \epsilon [t_1, t_2]}} r^i(t_w^i, t_a^i) \,.$$

However, as a result of the randomness of t_a^i, the direct return r^i for each job is also random. The probability space is the collection, c, of all sets $\left\{t_a^i\right\}_i$ which can result from a particular set of desired arrival times $\left\{\hat{t}_a^i\right\}_i$. As a consequence, the quantity to be maximized is the expectation of R_d over c,

$$R_e = E_c\left\{R_d\right\} = E_c\left\{\sum_{\substack{i \\ t_c^i \epsilon [t_1, t_2]}} [g_i(t_a^i) - p_i t_w^i]\right\} \tag{8}$$

Noting that the sums are finite, and letting

$$g_i^*(\hat{t}_a^i) \triangleq E_c\left\{g_i(t_a^i)\right\} \quad , \tag{9}$$

we can write Equation 8 as

$$R_e = \sum_{\substack{i \\ t_c^i \epsilon [t_1, t_2]}} g_i^*(\hat{t}_a^i) - E_c\left\{\sum_{\substack{i \\ t_c^i \epsilon [t_1, t_2]}} p_i t_w^i\right\} \,. \tag{10}$$

The process of choosing the intended arrival times, $\left\{\hat{t}_a^i\right\}$ and the waiting line control rules governing the waiting times $\left\{t_w^i\right\}$ in order to maximize R_e may be divided into two

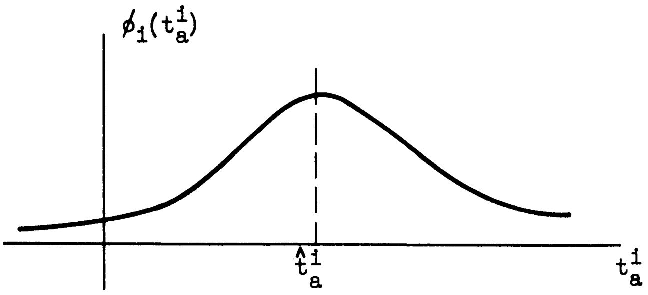

Fig. 4. Probability density of arrival time of job i.

parts. First, we can ask "What waiting line control rules will maximize R_e for each possible set of intended arrival times?" Second, we ask "If the optimum waiting line control rule is assumed to be automatically applied, what set of intended arrival times should be chosen?" The answer to the first question determines the optimum expected waiting times as a function of the intended arrival times. The second question takes the relationship found by substitution of the above waiting time into the return function, notes that it is a function only of the intended arrival times, and asks for the optimum overall possible choices of intended arrival times.

The remainder of this chapter will treat these two questions in turn. The problem represented by first question will be called the Queue Scheduling Problem; that represented by the second is the Arrival Scheduling Problem.

3.2 The Queue Scheduling Problem

The time of actual arrival of the user to the console can be said to have a probability density function $\phi_i(t_a^i)$. Without loss in generality we may assume the mean of this distribution to be the intended arrival time t_a^i. Figure 4 shows a typical probability density as it can be expected to appear. The variance of the distribution is some number, subject to measurement, which may typically be on the order of several minutes. Since the factors enumerated in Section 3.4 as contributing to this randomness are similar for most people, it is not unreasonable to assume that this distribution is the same for all jobs, differing only in mean, that is, there is a single function ϕ such that, for all i

$$\phi_i(t_a^i) = \phi(t_a^i - \hat{t}_a^i) . \tag{11}$$

If this is the case, one can write that the probability that a job arrives in the infinitesimal time interval [t, t + dt], given the set of intended arrivals $\{\hat{t}_a^i\}$, is

$$\nu(t)\,dt = \sum_i \phi(t - \hat{t}_a^i)\,dt\,. \tag{12}$$

There will be considerable smoothing resulting from the summation in Eq. 12 if severe clustering of values of $\hat{t}_a$ does not occur. To illustrate, if the $\hat{t}_a^i$ were uniformly spaced in time, even as far apart as 1.8 times the standard deviation of t_a, and ϕ were the Gaussian distribution, then the sum (Eq. 12) would be within 1 percent of a constant value.[1] If the spacing were equal to the standard deviation the error would be less than .0001 percent. However, since the $\hat{t}_a^i$ are not uniformly spaced but also not strongly clustered,[2] then even though the mean spacing is a small fraction of the standard deviation some variation of ν with time can be expected. Nevertheless, the variation can be assumed to be very small over time intervals of the order of the standard deviation of ϕ, which is, perhaps, of the order of several minutes.

Observing that the probability of an arrival in the interval [t, t + dt] is approximately independent of the occurrence of other arrivals within an interval of the order of the mean waiting time, we must conclude that the arrival process approximates a slowly varying Poisson process. In addition, assuming that the percentage variation of ν over intervals of the order of the waiting time of jobs in the queue (which for a well designed system should be less than the several minutes agreed to above) is small, one can conclude that results based on time invariant queuing theory can be used by replacing the parameter ν by the function of time $\nu(t)$.

A further simplification of the problem can be obtained in the return function. First, since the $\{\hat{t}_a^i\}$ are assumed fixed in this part of the optimization, the sum of the g_i^* terms in Eq. 10 is irrelevant to the optimization. Hence R_e in Eq. 10 can be replaced by the new return, R_q, which is to be maximized in this section,

[1] The author is indebted to T. G. Birdsall, of CEL, for these (unpublished) observations and calculations, which are contained in his personal notes.

[2] The lack of clustering will be verified in the determination of arrival policies in Section 3.4.

$$R_q = -E_c \left\{ \sum_{\substack{i \\ t_c^i \epsilon [t_1, t_2]}} p_i t_w^i \right\} . \tag{13}$$

3.3 The Optimal Queue Scheduling Policy

The queue scheduling problem is now in the form of a problem in scheduling theory, the problem of scheduling with Poisson arrivals and linear loss functions. The return described in Eq. 13 is characteristic of the so-called deferral cost model in scheduling theory, although in the existing theory, random arrivals have not been considered.

In addition, existing theory requires specification of the essential structure of the system in terms of either a single channel facility or a parallel-multiple channel facility. It is very likely that bottlenecks in service, that is, service channels which represent the chief causes for the formation of queues, would fall into these categories so that such a model would represent a reasonable approximation.

We shall assume that this is the case, and proceed with determining the optimum policy for such structures. It will be seen in the following discussion that the structure does not finally affect our choice of strategy in an important way, although extreme deviations from the essentially multiple-channel structure might invalidate the conclusions. The multiple-channel scheduling problem with linear deferral costs and Poisson arrivals is the essential problem to be considered.

In a companion report (Ref. 11), Fife surveys the status of current theory pertinent to this problem, and makes some very desirable extensions to it.

In that report, it is pointed out that while the problem without arrivals has not been solved explicitly there have been a number of results which have strong bearing upon the optimal scheduling strategy. Notably the result of Smith (Ref. 12), restated by McNaughton (Ref. 12), with some extensions, is that in scheduling an existing queue for a single service channel when no further arrivals are permitted, the optimal strategy is to service jobs in the order of descending values of the ratio p_i/t_s^i, where p_i is the loss rate (slope) of the job return, and t_s^i is the service time required by that job.

Fife has further shown in his report that if Poisson distributed arrivals are permitted during service of the queue, and if the cost of pre-emption of the job currently in service is prohibitively high, then the McNaughton-Smith strategy remains the optimum strategy in a single channel system. In other words, new arrivals are to be sequenced along with old problems in descending order of the ratio p_i/t_s^i and jobs once begun are not interrupted. If the cost of pre-emption is zero (and pre-emption requires zero time), he proposes a modified McNaughton-Smith strategy which appears to apply, but which has not been proven.

The generalization to multiple channel facilities, however, is very difficult. In fact, only a few properties of the optimum strategy without arrivals are known. Lawler, in a report prepared for this contract (Ref. 14), has treated the optimization of nonlinear loss functions and multiple channels when new arrivals are not permitted. Several interesting results are obtained, including some algorithms helpful for obtaining the exact optimal schedules in multiple-channel systems. Unfortunately, the need for extensive computation precludes their repeated use in scheduling computer use. The essential conclusion drawn from the work is that an exact optimization of even the nonarrival case is impractical for our purposes.

A procedure for determining an optimum schedule for even the parallel machine is therefore not available. The investigation of these approaches gives reason to believe that if a procedure was known which was successful in the sense of always leading to an optimum schedule, it is likely that it would involve a considerable amount of computation. The drawback of any extensive scheduling algorithm for the multiple-console system is that the required execution time can soon exceed the execution time of some of the problems to be scheduled. When the return loss resulting from this delay is accounted for, it is conceivable that the total return loss of an "optimum" schedule may be greater than that of a nonoptimum schedule which is computed more readily.

This thinking suggests that a more worthwhile approach is to implement a simple, near-optimum scheduling policy. The obvious choice is the McNaughton-Smith rule: schedule jobs for execution on any available processor in order of decreasing value of p_i/t_s^i. Several facts justify this choice. First, it is known to correspond to the optimum parallel machine rule when all jobs have equal execution time. Second, it is easily shown that the set of jobs assigned to a particular channel in an optimum schedule will be in accord with the McNaughton-

Smith rule even when execution times are unequal.

There is further support for the conclusion that the McNaughton-Smith rule is a useful approximation to the optimum from a limiting case of some generality. Suppose that jobs in queue are grouped into types or classes according to the values of p and t_s associated with them. Every job in the same class then has the same associated values of p and t_s. Let there be a sufficiently large number of jobs of each class that the execution time of one job is negligibly small compared to the total time required to execute all jobs in the same class.[1] It follows that time intervals in which the machine is processing jobs from two or more different classes are negligible compared to time intervals in which all jobs in process are of the same class. The parallel machine can therefore be considered as a single "channel" for processing of entire classes, and we can apply the McNaughton-Smith rule to scheduling by classes. The loss rate associated with delay of processing of the ith class is $N_i p_i$, where N_i is the number of jobs of that class. The execution time for the entire class is $N_i t_s^i / M$ in the limit of large N_i, where M is the number of parallel channels. Thus classes are selected for execution in order of decreasing value of $MN_i p_i / N_i t_s^i$, which clearly is equivalent to selecting individual jobs in order of decreasing value of p_i / t_s^i.

Although rigorous analytical justification for applying the McNaughton-Smith rule to scheduling of a parallel machine seems difficult indeed, some check on its accuracy might be obtained by a Monte Carlo sampling of possible queues, to ascertain whether a McNaughton-Smith schedule does lead to nearly maximum return in an overwhelming proportion of cases. The sampling procedure has been previously applied in this way to scheduling problems (Ref. 15). In this connection the transportation formulation (Ref. 14) might prove valuable since the lower bound it gives on the loss of the optimum schedule could be used as the standard of comparison for the McNaughton-Smith schedule. Otherwise the loss of the optimum schedule would have to ve determined by exhaustion.

Nevertheless, without extensive further study, it is necessary to accept this scheduling rule for multiple channel structures as the best available approximation. Of course, if the essential service structure is single channel the rule results in an exact opti-

[1] This limiting case was suggested by F. Waltz of the Cooley Electronics Laboratory

mum schedule (shown by Fife). This latter case would correspond, for example, to a machine in which the primary queueing in busy periods was for the use of the central processing unit. The essential structure would be multi-channel if the chief bottleneck were I/0 transfer time in many identical I/0 channels.

The McNaughton-Smith rule will be taken as the optimal queue scheduling policy.

3.4 The Optimal Arrival Scheduling Policy

The function which was intended to be maximized by the choice of the set of arrival times, $\{\hat{t}_a^i\}$, was given in Section 3.1 as

$$R_e = \sum_{\substack{i \\ t_c^i \epsilon [t_1, t_2]}} g_i^*(t_a^i) - E_C \left\{ \sum_{\substack{i \\ t_c^i \epsilon [t_1, t_2]}} p_i t_w^i \right\} . \tag{10}$$

However, before proceeding with the maximization of this return, we must consider the dependence of the expectation of waiting times, $\{t_w^i\}$, on the intended arrival times $\{\hat{t}_a^i\}$. Since the waiting time is a result of the queueing for service, the Poisson arrival model continues to apply, and jobs are given priority in the queue according to the McNaughton-Smith rule.

The problem of determining the expected waiting time in a priority queueing system with Poisson arrivals has been considered in the literature. Cobham and Holley (Refs. 16 and 17) have determine it for single and multiple channel structures with discrete priority levels, where pre-emption is not permitted. Their result has been extended by Wallace (Ref. 18) to general structures where certain assumptions on statistical properties of the times between inputs to the service facility are satisfied.

A common property of all these solutions is that the expected waiting time is a function of only the total service time of jobs arriving in unit time having priority higher than the one waiting, that is, the waiting time of job i at time t is a single valued function of a quantity which can be written (in the context of this report) as

$$\sum_{\substack{k: \\ m_k > m_i}} t_s^k \phi(t - \hat{t}_a^k) , \tag{14}$$

where $\phi(t - \hat{t}_a^k)dt$ is the probability of job k arriving in the time interval $[t, t+dt]$, $m_k = p_k/t_s^k$ (the McNaughton ratio), and t_s^k is the service time of the kth job (as measured by the time between input to the service facility from the queue and the next time the facility is available to service a job). Representing the function by the symbol w, we may write the expectation of the waiting time of job i as

$$E\{t_w^i\} = w\left[\sum_{\substack{k:\\ m_k > m_i}} t_s^k \phi(t - \hat{t}_a^k)\right] . \tag{15}$$

We must note that although the conditions specified in the above references are quite common, there is a possibility that they do not strictly apply to the service structures which can be encountered in this system. Nevertheless, in the absence of any evidence to the contrary, it is not unreasonable to assume that the form (Eq. 15) applies. We shall assume that it does, at least approximately.

The return R_e of Eq. 10 can now be written

$$R_e = \sum_i g_i^*(\hat{t}_a^i) - \sum_i p_i w\left[\sum_{\substack{k\\ m_k > m_i}} t_s^k \phi(\hat{t}_a^i - t_a^k)\right] . \tag{16}$$

which is now clearly a function of the set of control variables $\{\hat{t}_a^i\}$ only. Since none of these variables is constrained by the values of any of the others, the optimum return is obtained by simply maximizing with respect to each of the variables in turn, keeping the remaining variables as parameters.

If we take $\hat{t}_a^j$ as the typical intended arrival time to be selected, we may write (separating those terms not dependent on $\hat{t}_a^j$ from those which are):

$$R_e = g_j^*(\hat{t}_a^j) - p_j w\left[\sum_{\substack{k\\ m_k > m_j}} t_s^k \phi(\hat{t}_a^j - \hat{t}_a^k)\right]$$

$$- \sum_{\substack{i\\ m_i<m_j}} p_i w \left[\underset{\substack{k\\ m_k>m_i}}{} \; t_s^k \,\phi(\hat{t}_a^i - \hat{t}_a^k) \right] + M \qquad (17)$$

where M is a term representing terms independent of t_a^j. It will be noted that the first two terms on the right hand side of Eq. 17 are the long term return of the job itself, and the loss of the job itself due to its own waiting, respectively. The third term represents the effect on the expected return of all other jobs by the arrival of job j. It is clear that, because of the priority arrangement, only jobs with lower priorities are so affected $(m_i < m_j)$.

The third term may be further reduced by first writing it as:

$$\sum_{\substack{i:\\ m_i<m_j}} p_i w \left[\underset{\substack{k:\\ m_k>m_i\\ k\neq j}}{} \; t_s^k \,\phi(\hat{t}_a^i - \hat{t}_a^k) + t_s^j \,\phi(\hat{t}_a^j - \hat{t}_a^i) \right] .$$

Then, observing that $t_s^j \;\phi(\hat{t}_a^j - \hat{t}_a^j)$ is very small compared to the [ ] bracketed argument, and finally expanding by taking the first two terms of the Taylor expansion of $p_i w$ as an approximation to it, we obtain

$$\sum_{\substack{i:\\ m_i<m_j}} p_i w \left[\sum_{\substack{k:\\ m_k>m_i\\ k\neq j}} t_s^k \phi(\hat{t}_a^i - \hat{t}_a^k) \right] + \sum_{\substack{i:\\ m_i<m_j}} p_i t_s^j \phi(\hat{t}_a^j - \hat{t}_a^i)\, w' \left[\sum_{\substack{k:\\ m_k>m_i\\ k\neq j}} t_s^k \phi(\hat{t}_a^i - \hat{t}_a^k) \right] ,$$

where w' denotes the derivative of w with respect to its argument. Since the first of these two terms is also independent of $\hat{t}_a^j$, we may rewrite Eq. 17.

$$R_e \approx g_j^*(\hat{t}_a^j) - p_i w \left[\sum_{\substack{k:\\ m_k>m_j}} t_s^k \phi(\hat{t}_a^j - \hat{t}_a^k) \right]$$

$$- \sum_{\substack{i: \\ m_i < m_j}} p_i t_s^j \phi(\hat{t}_a^j - \hat{t}_a^i)\, w' \left[\sum_{\substack{k: \\ m_k > m_i \\ k \neq j}} t_s^k \phi(\hat{t}_a^i - \hat{t}_a^k) \right] + M' \qquad (18)$$

where M' is a term representing terms independent of $\hat{t}_a^j$.

If each of the expressions of the form of Eq. 17 for each of the jobs is maximized by the corresponding user's choice of intended arrival time, and if sufficient information transfer is available that all are maximized simultaneously, then an optimal arrival scheduling policy will have been carried out. It should be noted that the first two terms in Eq. 17 represent factors which would normally be considered by a user in his local optimization. It remains for us to show in the next chapter how the third factor in Eq. 17 can be brought into play, and how sufficient motivation for the user to attempt maximization of this return can be developed. The problem of providing close informational coupling between users will also be discussed.

4. CONTROL OF PRICE FOR SYSTEM OPTIMIZATION

One of the principal implications of this economic model for control of computer use is that the control is exerted solely on the basis of return (utilities and direct costs) of the user. Thus, the choice of priorities and schedules is independent of the cost of operating the system.

Since the control of user decisions is to be attained through manipulation of the prices charged to the user for the service to his job, it follows that prices will be dependent on the return to the users, and should not be dependent on system costs at all. This fact should not be surprising since prices in competitive, nonmonopolistic, economic systems are normally quite independent of costs.

It is the purpose of this chapter to show that a particular pricing system to be proposed not only controls the use of the shared computer system, but also controls it in an optimum manner. It was indicated in Chapter 2 that the matters of cost of the operation of the system are resolved only by considerations of the final profitability of the system when the total return, not the total collected from users, is contrasted to operating costs.

Control by the use of prices would not be required at all if there were not an essential difference between the choices or policies which were best for the individual user and the choices or policies which were best for the enterprise itself. Whenever there is no conflict between the objectives of users, the individual user can proceed to solve his decision problems separately. However, when objectives are in conflict, the optimization of one can result in a lower degree of attainment for all the others. This condition is known as the condition of suboptimization. The fact of the competition among users for immediate service on the shared computer results in just such a conflict, so that the choices of priority by individual users must be influenced for the common good. Chapter 2 has presented the structure of what constitutes this "common good," while Chapter 3 has developed the optimal policies for this "common good."

The system for establishing prices will now be presented and it will be shown that, with rational decisions on the part of the users, this system does result in decisions which are truly optimum. Later in this chapter, some of the methods by which rational behavior can be encouraged will be briefly discussed.

4.1 A Pricing System

In order to establish a pricing system we first establish a large number of what we shall call precedence levels, with each precedence level having a certain price per unit service time[1] associated with it. We shall refer to this price per unit service time as the price rate, and designate it by the symbol π. Since price rates are in one-to-one correspondence with precedence levels we shall use this same symbol π to designate the precedence level.

Now, we say that each user has the right to designate for his job any of the possible precedence levels, provided that he is willing to pay the price which corresponds. Moreover, the system will not initiate service on any job of lower precedence until all jobs of higher precedence have entered service.

At any given time, there will be a waiting line of jobs formed. Each job will be tagged by the precedence chosen, or equivalently by the price which the corresponding user is willing to pay for service. As a result of the queue, and the probabilities that other jobs will arrive and take a place in the queue at higher precedence than some waiting jobs, an expected waiting time for each job can be determined.

In particular, on the basis of queueing models and statistical information gathered, the expected waiting time which a newly arriving job would encounter at each precedence level could be calculated by the executive system of the computer and transmitted to the consoles. This information could be presented, perhaps, by a plot of price rate versus expected waiting time on a display screen of the console. The user could then designate his choice of precedence on the basis of this information.

Since the expected waiting times would vary with time, the display would, of course, also vary with time. However, the variation of expected waiting time should not be more rapid than a few per cent per minute, and this variation would not confuse the user.

[1] Service time will continue to be defined as it was in Section 3.4.

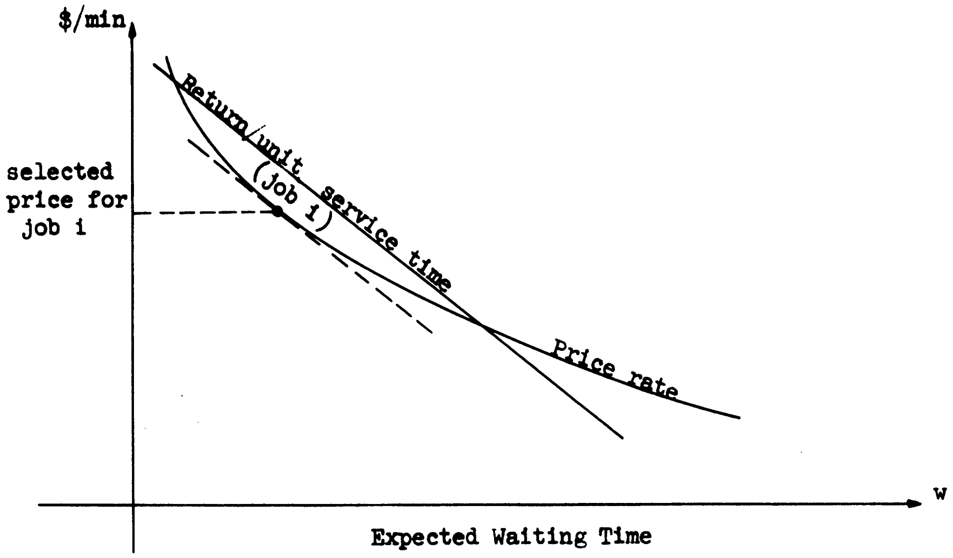

Fig. 5. Illustration of rational price selection by user.

4.2 Satisfaction of the Optimum Queue Scheduling Policy

We shall now show that rational users will select price levels which will automatically schedule jobs according to the McNaughton-Smith Rule, once the jobs have arrived at the console. This corresponds to the optimum queue scheduling rule determined in Chapter 3.

Figure 5 illustrates what would constitute a rational selection of price by a user. Since the return for job i has been approximated in Chapter 3 by Eq. 7

$$r_i(t_w^i, \hat{t}_a^i) = g_i(\hat{t}_a^i) - p_i t_w^i \ , \tag{7}$$

the expected return per unit service time of job i can be represented by a linear function of the expected waiting time w_i

$$E\left\{\frac{r_i(t_w^i, \hat{t}_a^i)}{t_s^i}\right\} = \frac{g_i(\hat{t}_a^i)}{t_s^i} - m_i E\left\{t_w^i\right\} = K - m_i w_i \ , \tag{19}$$

as illustrated in the figure. Since the user should select his price to maximize the difference between return and price, his selected price rate should be the one for which the slope of the price curve is equal to the slope of his return curve.

We say that such a selection will be made, although we recognize that the return function may have an unfocused image in the mind of the user. Nevertheless, the conscious

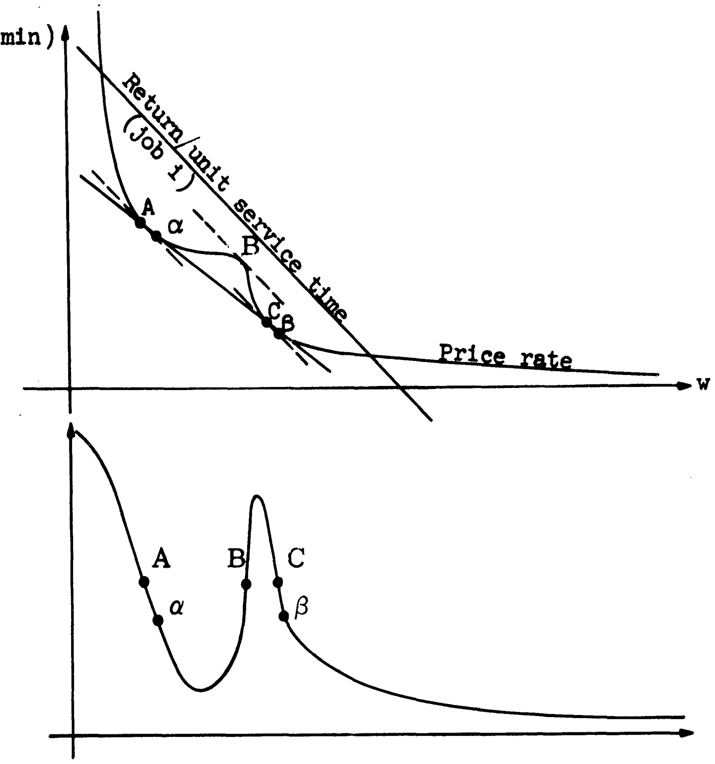

Fig. 6. Illustration of an unstable price condition.

or unconscious balance of urgency and price paid must represent some attempt at carrying out the above procedure, and the above is a model for it.

It is easy to show that the price rate so derived must be a monotonic decreasing function of expected waiting time. However, it can also be shown that the derivative of the price rate will be negative, and a monotonically increasing function of expected waiting time. This can be shown by showing that a price curve which does not have the monotonic derivative property is unstable under this pricing system, and tends toward the monotonic condition.

The situation that would occur if the derivative of the price rate were not monotone is illustrated in Fig. 6. Suppose job i has the return rate shown. In that case, the precedence level corresponding to point A is the one that would be chosen, since it obviously results in a maximum of the net return to the user. However, there are three points, A, B, and C on the price rate curve where the slope is the same as that of the return rate curve, and points B and C would obviously not be chosen for this or any other job. In fact, it can be argued in this manner that none of the points between α and β, which are points of common tangency on the price curve, would be chosen by any user. The points α and β represent points of indifference for a job having a slope equal to that of the common tangent.

If, then, no jobs were to arrive between α and β, then the queue for jobs in the precedence levels corresponding to this interval would gradually be depleted by completions of service and no new arrivals would replace them. Hence, the waiting time for a job at point α and one at point β would approach equality and the perturbation in the curve would have shrunk to nothing. In this manner, we can argue that the occurrence of even small nonmonotonic slope regions on the price curve would be quickly eliminated by the avoidance of such regions by the rational users.[1] As a result, we shall assume that the slope is monotonically increasing at all points and at all times.

It is now evident that the jobs will be matched with precedence levels corresponding to slopes in the price rate curve equal to their McNaughton ratio. Because of the monotonic property of the slopes, this will always give a job of higher McNaughton ratio a higher precedence level as well as a higher price rate. Thus, this pricing system results in scheduling jobs according to the optimum queue scheduling policy determined in Chapter 3.

4.3 Satisfaction of the Optimum Arrival Scheduling Policy

Figures 5 and 6 represent only an approximation because, since there are only a finite number of jobs, there must only be a finite number of precedence levels. Therefore, what appears to be smooth price curves are actually made up of a large number of tiny steps. Nevertheless, the slopes of lines drawn connecting the points of the steps must still have the desired relationships discussed above. For convenience, we are allowing all prices to be possible, and associating each job of the finite set with a distinct price.

If one takes the price of the job with the lowest McNaughton ratio as zero, then the expected price rate of any job can be represented as the sum of the McNaughton ratio (the slope of price rate) multiplied by the increment of expected waiting time for all jobs of lower priority. Thus

[1] We assume at this point that the reaction of users to the nonmonotonic slope is not so violent as to cause a new non-monotonic region on the curve. Such a reaction might result in a completely unstable response of the system, but could be eliminated (at some loss in return) by either not allowing the price curve to reflect very fast variations in arrivals, or restricting the user's freedom to change his mind once he has designated his priority. However, because of the expected slowness and diversity of user reactions, this difficulty is not anticipated.

$$E\{\pi_j\} = \sum_{\substack{i: \\ m_i < m_j}} m_i \left\{ w \left[\sum_{\substack{k: \\ m_k > m_i}} t_s^k \phi(\hat{t}_a^j - \hat{t}_a^k) + t_s^i \phi(\hat{t}_a^j - \hat{t}_a^i) \right] - w \left[\sum_{\substack{k: \\ m_k > m_i}} t_s^k \phi(\hat{t}_a^j - \hat{t}_a^k) \right] \right\} \tag{20}$$

is the expression for the expected price rate of job j. If we multiply this price rate by the service time of job j, we get the expected price of job j. In addition, we note that $t_s^i \phi(\hat{t}_a^j - \hat{t}_a^i)$ is very small compared to the summation in the argument of w in Eq. 20. Hence, the difference of expected waiting times in Eq. 20 cam be represented by the first nonzero term of a Taylor series expansion. Then the expected price paid for the service of job j can be written approximately

$$E\{t_s^j \pi_j\} \cong \sum_{\substack{i: \\ m_i < m_j}} m_i t_s^j t_s^i \phi(\hat{t}_a^j - \hat{t}_a^i) \, w' \left[\sum_{\substack{k \\ m_k > m_i}} t_s^k \phi(\hat{t}_a^j - \hat{t}_a^k) \right] \tag{21}$$

where w' denotes the derivative of w.

On the other hand, the expected return as expressed in Eq. 18 was

$$R_e \cong g_j^*(\hat{t}_a^j) - p_j \, w \left[\sum_{\substack{k \\ m_k > m_j}} t_s^k \phi(\hat{t}_a^j - \hat{t}_a^k) \right]$$

$$- \sum_{\substack{i: \\ m_i < m_j}} m_i t_s^i t_s^j \phi(\hat{t}_a^j - \hat{t}_a^i) \, w' \left[\sum_{\substack{k \\ m_k > m_i \\ k \neq j}} t_s^k \phi(\hat{t}_a^i - \hat{t}_a^k) \right] + M' \tag{22}$$

where the equivalent expression $m_i t_s^i$ has been used in place of p_i. But the expected net return for job j as viewed by the user is the first two terms of Eq. 22 (representing his general return less his loss due to waiting) minus the price he must pay for service. This net personal return can be written

$$R_p^j \cong R_e + \sum_{\substack{i: \\ m_i < m_j}} m_i t_s^i t_s^j \phi(\hat{t}_a^j - \hat{t}_a^i)\, w' \left[\sum_{\substack{k: \\ m_k > m_i \\ k \neq j}} t_s^k \phi(\hat{t}_a^i - \hat{t}_a^k) \right]$$

$$- \sum_{\substack{i: \\ m_i < m_j}} m_i t_s^j t_s^i \phi(\hat{t}_a^j - \hat{t}_a^i)\, w' \left[\sum_{\substack{k: \\ m_k > m_i}} t_s^k \phi(\hat{t}_a^j - \hat{t}_a^k) \right] - M' \tag{23}$$

The expression

$$\sum_{\substack{k \\ m_k > m_i}} t_s^k \phi(\hat{t}_a^j - \hat{t}_a^k)$$

represents an arrival rate of jobs for service at higher priority than job i. By an argument essentially the same as that used is Section 3. 2, we can argue that this expression is a slowly varying function of $\hat{t}_a^j$. In particular, there will be very little change in the expression as $\hat{t}_a^j$ varies over intervals of the order of several standard deviations of the distribution ϕ. Thus, the bracketed [ ] quantities in the second and third terms of Eq. 23 are approximately equal provided that $\hat{t}_a^i$ is within the order of several standard deviations of $\hat{t}_a^j$. However, if these two intended arrival times differ by more than several standard deviations then the quantities $\phi(\hat{t}_a^j - \hat{t}_a^i)$ will be very small.

We then conclude that the two summations in Eq. 23 are essentially equal to one another, and hence

$$R_p^j \cong R_e - M' . \tag{24}$$

Since M' is not a function of the intended arrival time of job j,then a choice of this intended arrival time to maximize the personal return as viewed by the user for job j actually results in a maximization of the global return R_e. Hence, the proposed pricing system will satisfy the optimum arrival scheduling policy described in Chapter 3.

4.4 Information for Users

In order that the maximization of Eq. 24 represent a global optimum selection of the set of intended arrival times, all individual decisions must simultaneously reflect the individual decisions of the other users. For this reason, it is vitally important that the users be given up-to-date and accurate estimates of the expected waiting times and corresponding prices for all hours of the day. These estimates can be based upon past performance, existing queues, existence of holidays and other such factors, and projections using queueing models. The precise methods by which these estimates would be determined are critically dependent upon system specifics and specific job mix characteristics, and hence require special study for each system. In any event, it is beyond the scope of this report to do more than simply observe that there are many statistical tools available for the purpose.

Besides providing up-to-date estimates, they should be in a clear graphical form which will be easy for the user to evaluate. They should not cause appreciable delay to the user, since that would represent a degradation of service.

One way to make the assimilation of predictions more convenient to the user would be to allow the user to designate a priority which suits him at present, and the system would then present prices and waiting times as a function of time which correspond to the same McNaughton ratio as the one chosen. In this way, he could obtain a plot on the screen of his console showing price and waiting time versus time of the day for his self-specified priority.

Finally, it should be noted that provision for only a fairly small number of priority levels should not significantly degrade the efficiency of the chosen pricing system. For example, one could divide the range of all likely McNaughton ratios into a fixed number of intervals (say ten), and then automatically project from an arbitrarily low price rate for the lowest interval a slope equal to the McNaughton ratio at the upper boundary of the interval, allow this slope to stand until the waiting time of the next higher interval is encountered, and proceed similarly as in Fig. 7.

It is seen that users will choose a priority level corresponding to a point which is between the two line segments whose slopes bracket the McNaughton ratio of the job. The matching, of course, automatically prepares and displays the price rate curve, as before.

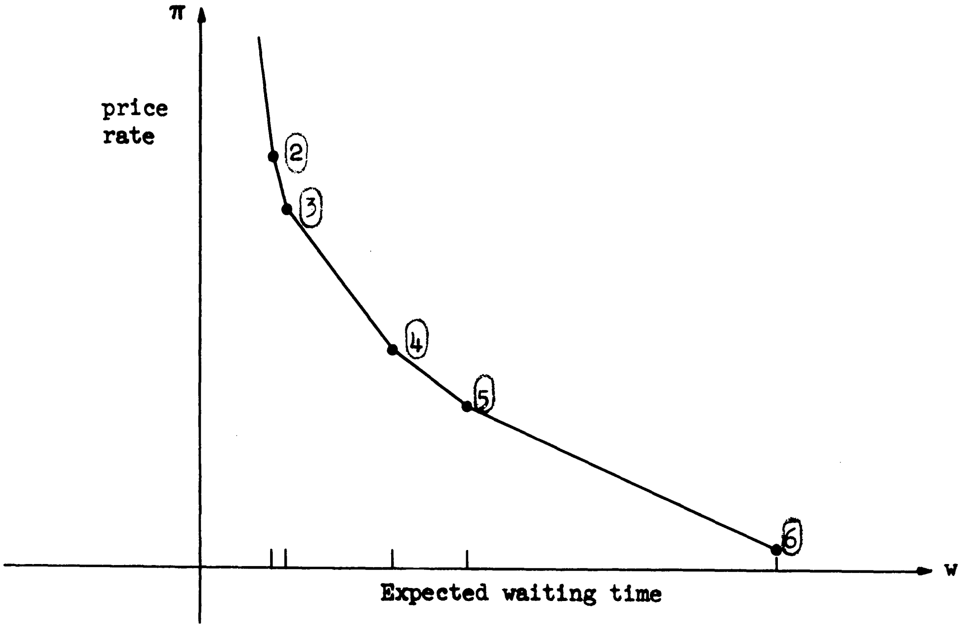

Fig. 7. A discrete price rate curve.

While the mathematical analysis for a discrete system such as this is less straightforward than that presented in the previous section, it will nevertheless represent an approximation to the optimization of the shared multiple console data processing system if the number of priority levels is chosen reasonably.

4.5 Correcting for Subjective Utilities of Users

The last remaining problem is that of providing proper motivation of the user toward assuming the correct utility for his job. It is obvious that different people will evaluate the utilities of their work in different manners, some more conservatively than others. The utilities and returns which have been used in the development of the conclusions of this report are what would be termed the "objective" utilities. That is, they are utilities and returns as they would be evaluated by an objective second party or authority. On the other hand, unless something is done the users will be making their choices and decisions on the basis of what might be termed the "subjective" utilities and returns.

In addition to the subjectivity, there may be a problem of sufficient motivation of the user toward maximizing the return of his work. Although the purpose of this report is to reduce the number of decisions to be made by supervisory personnel to tolerable limits, it is in this broad realm of the motivation of the user where administrative capability must come into play. Motivation is, as always, principally an administrative problem. It can be influenced primarily by such things as personal encouragement, incentive plans, and perhaps even by mild threats!

Similar techniques can be used to enhance objectivity, but it appears that the principal and most useful technique might here be one of compensating for it elsewhere. For example, different prices might be presented to different operators (which can be quite feasible in a computer system if the user identifies himself to the computer upon arrival). Similarly a fixing of the budget or allotment to an individual, to a project, or both, can have the same effect. Either procedure gives control of major decisions concerning allocation and value to supervisory personnel, but will not require frequent decisions concerning the detailed division of labor and resource for the completion of the projects.

Such a realm of decision and control is distinctly possible as well as desirable in a computer based enterprise.

4.6 Conclusions

It has been shown by an intricate but necessary argument, that by displaying expected waiting times to the console user as a function of price and by allowing operators to select their price within only broad influences by administration, a good approximation to optimum utility of the multiple-console shared computer system can be obtained.

It should be noted that the optimality is essentially independent of the structure and equipment complement of the system, provided the system is big enough for the engineering approximations utilized throughout the report to apply. The principal one of these approximations requires that a large number of jobs be serviced in the time interval of the uncertainty of user arrivals. Note that the optimum total return is not independent of the structure or equipment complement, but that the manner of obtaining the optimum return is.

In implementing this system, some of the philosophical attitudes held by the administration toward budgeting may require some change. We have noted that budgets are one means of exercising broad control over users' subjective evaluation of the value to apply to their work. To the extent that this is done, systematic pressures by users may be introduced which detract from uses of the budget by the administration for such purposes as projection of expenditure. This disadvantage is minor, since experience will indicate how to compensate for it.

Secondly, since the receipts of the computer facility from user payments has little relation to the return of the computer to the enterprise, a budget for the computer must be allowed to account for the differences between costs and receipts. Of course, this budget should not exceed the total return of the computer to the enterprise minus the receipts.

Unfortunately, if irrational rules of procedure, or rules forced by purely administration considerations preclude adjustment of price in the manner described, then the optimum utility for the shared system cannot be attained. For example, this might occur if auditors require charges for service to be based on operating costs. In our view the "free market" pricing system proposed is, in fact, also the fairest one which can be used, and such arbitrary requirements are unreasonable.

Finally, the economic considerations in the sharing of a multiple-console computer system by several enterprises whose utilities for service cannot be fairly and directly compared, may or may not be optimized by this procedure. This service bureau type application, however, is probably very nearly optimized <u>in some sense</u> by this "free market" pricing system.

REFERENCES

1. Hamming, R. W., "Frontiers in Computer Technology," Proc. of the Annual Computer Applications Symposium, Chicago, Ill., 1958.

2. Corbato, F. J., et al., "An Experimental Time-Sharing System," Proc. of Spring Joint Computer Conference, Vol. 21, 1962, pp. 335-344.

3. Culler, G. J., and B. D. Fried, "An On-Line Computing Center for Scientific Problems," Proc. Pacific Computer Conference, IEEE, 1963, pp. 221-242.

4. McCarthy, J., et al., "A Time-Sharing Debugging System for a Small Computer," Proc. of Spring Joint Computer Conference, Vol. 23, 1963, pp. 51-57.

5. Corbato, F. J., et al., The Compatible Time Sharing System - A Programmer's Guide, MIT Press, 1963 (see e.g., "Scheduling," pp. 10-11).

6. Rothman, S., "RW-40 Data Processing System," Int. Conf. on Information Processing, June 1959.

7. Simon, H. A., Models of Man, Wiley, 1957.

8. Drucker, Peter, The Practice of Management, Harper, 1954, pp. 62-65.

9. von Neumann and Morganstern, Theory of Games and Economic Behavior, Princeton University Press, 1953, Chapter 3.

10. Sisson, R. L., "Method of Sequencing in Job Shops," Operations Research, Vol. 7, No. 1, pp. 10-29, Jan.-Feb. 1959. (Review article including bibliography.)

11. Fife, D. W., "Scheduling with Random Arrivals and Linear Loss Functions," Technical Report No. 144, Cooley Electronics Laboratory, in progress.

12. McNaughton, R., "Scheduling with Deadlines and Loss Functions," Management Science, 6, pp. 1-12, 1959.

13. Smith, W. E., "Various Optimizers for Single Stage Production," Naval Research Logistics Quarterly, 3, pp. 59-66, 1956.

14. Lawler, E. L., "Computational Techniques for Scheduling Problems with Deferral Costs," Technical Report No. 140, Cooley Electronics Laboratory, July 1963.

15. Heller, D., "A Monte Carlo Approach to the Approximate Solution to Sequencing Problems," ACM National Conference, 11. 2, September 1962.

16. Cobham, A., "Priority Assignment in Waiting-Line Problems," Operations Research, 2, 1, February 1954, pp. 70-76.

17. Holley, J. L., "Waiting Line Subject to Priorities," Operations Research, 2, 3, August 1954, pp. 341-343.

18. Wallace, V. L., "Mean Waiting-Time for An Arbitrary Service Facility," Memo to 04967 File, Cooley Electronics Laboratory, September 30, 1963.

DISTRIBUTION LIST

1. John McLean, EMIIF (1)
RADC
Griffiss AFB
Rome, New York

2. Lt. L. W. Odell, EMIICA (2)
RADC
Griffiss AFB
Rome, New York

3. Pat Langendorf, EMIIT (1)
RADC
Griffiss AFB
Rome, New York

4. Fred Dion, EMIIT (1)
RADC
Griffiss AFB
Rome, New York

5. Rome Air Development Center
Griffiss AFB
Rome, New York

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)

6. GEEIA
Griffiss AFB
Rome, New York
Attn: ROZMA (1)
ROZMC (1)
ROZME (1)
ROZMN (1)
ROZMCAT (1)

7. RAOL (Maj. Shields) (1)
Griffiss AFB
Rome, New York

8. RAOL (S/L Tanner) (2)
Griffiss AFB
Rome, New York

9. AFSC
Andrews AFB
Washington 25, D. C.
Attn: SCSE (1)
SCLDE (1)
SCFRE (1)

10. Redstone Scientific Information Center (2)
U. S. Army Missile Command
Redstone Arsenal, Ala.
Attn: Chief, Document Section

11. Bureau of Naval Weapons (2)
Main Navy Building
Washington 25, D. C.
Attn: Tech. Lib. DL1-3

12. Chief, Bureau of Ships (1)
Code 454 F
Main Navy Building
Washington 25, D. C.

13. Central Intelligence Agency (1)
2430 E Street NW
Washington 25, D. C.
Attn: OCR Mail Room

14. U. S. Army Material Command (1)
Harry Diamond Laboratories
Washington 25, D. C.
Attn: AMXDO-TIB

15. Scientific and Technical Information Facility (2)
P. O. Box 5700
Bethesda, Maryland
Attn: NASA Representative (S-AK/DL)

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. USAFSS (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 RandD Lab
Attn: SELRA/ ADT
Ft. Monmouth, New Jersey

DISTRIBUTION LIST (Continued)

28. Institute of Technology Library (1)
MCLI-LIB, Bldg 125 Area "B"
Wright-Patterson AFB
Ohio

29. NAFEC Library (1)
Bldg. 3
Atlantic City, New Jersey

30. Commandant (1)
US Army War College (Lib)
Carlisle Barracks, Pa.

31. Commanding Officer (1)
US Army Electronic Research Unit
P O Box 205
Mountain View, California

32. Electronic Defense Labs (1)
Attn: Library
P O Box 205
Mountain View, California

33. U S Naval Avionics Facility (Library) (1)
Indianapolis 18
Indiana

34. Commander
US Naval Air Dev Cen (NADC Lib) (1)
Johnsville, Pa.

35. Director (2)
US Naval Research Lab (Code 2027)
Washington 25, D. C.

36. Hq USAF
Attn: AFRSTE (1)
AFRDPC (2)
AFGOA (1)
Washington 25, D. C.

37. National Aeronautics and Space Admin (3)
Langley Research Center
Langley Station
Hampton, Virginia
Attn: Librarian

38. RTD (RTH) (1)
Bolling AFB 25 D. C.

39. Federal Aviation Agency (1)
Information Retrieval Br Hq-630
Washington 25, D. C.

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